Machine Learning: Science and Technology

PAPER • **OPEN ACCESS**

Chemformer: a pre-trained transformer for computational chemistry

To cite this article: Ross Irwin *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 015022

IOP Publishing Mach. Learn.: Sci. Technol. **3** (2022) 015022 https://doi.org/10.1088/2632-2153/ac3ffb

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
6 September 2021

REVISED
15 November 2021

ACCEPTED FOR PUBLICATION
3 December 2021

PUBLISHED
31 January 2022

PAPER

Chemformer: a pre-trained transformer for computational chemistry

Ross Irwin[1], **Spyridon Dimitriadis**[1,2], **Jiazhen He**[1] **and Esben Jannik Bjerrum**[1,*]

[1] Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden
[2] Department of Computer and Information Science, Linköping University, Linköping, Sweden
[*] Author to whom any correspondence should be addressed.

E-mail: esben.bjerrum@astrazeneca.com

Keywords: transformer, self-supervision, chemistry, reaction prediction, molecular optimization, QSAR

Supplementary material for this article is available online

Abstract

Transformer models coupled with a simplified molecular line entry system (SMILES) have recently proven to be a powerful combination for solving challenges in cheminformatics. These models, however, are often developed specifically for a single application and can be very resource-intensive to train. In this work we present the Chemformer model—a Transformer-based model which can be quickly applied to both sequence-to-sequence and discriminative cheminformatics tasks. Additionally, we show that self-supervised pre-training can improve performance and significantly speed up convergence on downstream tasks. On direct synthesis and retrosynthesis prediction benchmark datasets we publish state-of-the-art results for top-1 accuracy. We also improve on existing approaches for a molecular optimisation task and show that Chemformer can optimise on multiple discriminative tasks simultaneously. Models, datasets and code will be made available after publication.

1. Introduction

Recent years have witnessed an explosion in research applying neural network models to cheminformatics tasks. Sequence-to-sequence models, such as the Transformer [1] and models based on the recurrent neural network architecture [2, 3], are well suited to tasks such as direct reaction prediction [4, 5], retrosynthesis prediction [5] and molecular optimisation [6, 7]. Applying molecules encoded using the simplified molecular line entry system (SMILES) [8] to the Transformer model has produced state-of-the-art results in benchmark datasets for these tasks [5–7]. Transformers have also been successfully applied to discriminative tasks, such as biological activity prediction (virtual screening) [9] and molecular property prediction (QSAR modelling) [9–15]. Training Transformer models on SMILES strings, however, can be computationally expensive. For example, a recently proposed model for direct synthesis prediction requires two days of training [4] for a single set of hyperparameters. Depending on the availability of computational resources, hyperparameter tuning can then lead to weeks or months of work for research teams. Additionally, separate models must be built, trained and tuned for each task, increasing the amount of effort required by researchers. Moreover, long training times may also limit the exploration of the performance of larger models and larger datasets.

Self-supervised learning using the Transformer has revolutionised natural language processing (NLP) in recent years; large language models such as BERT [16], BART [17], GPT [18, 19], UniLM [20] and T5 [21] have provided significant improvements to key benchmark NLP tasks. Pre-training these models—training on a large unlabelled dataset of text before fine-tuning on the dataset of interest—has been shown to improve results in downstream tasks, especially when the amount of data for fine-tuning is limited. Furthermore, pre-training can also significantly reduce the amount of time required for fine-tuning [16], thereby reducing computational costs. This can then enable more extensive downstream hyperparameter tuning or make state-of-the-art models more accessible to those with limited computational resources.

Transfer learning has also recently been shown to improve performance in reaction informatics tasks [22–27] and, separately, in discriminative tasks [9–11, 15, 28]. However, many of these approaches pre-train on a task-specific dataset, such as reaction informatics data. It is unclear how well these models would be able to transfer their knowledge to other domains. Other approaches make use of the encoder stack of the Transformer only, along with a fully-visible attention mask [9, 10, 12]. This makes it difficult to apply these models to sequence-to-sequence tasks. In one study, embeddings from a self-supervised *Augmented Transformer* were used to build QSAR models [29], but the pre-trained weights were not subsequently fine-tuned.

One model, X-MOL [11], uses a Transformer encoder with a combined fully-visible and autoregressive attention mask. This allows the model to be applied to both discriminative and sequence-to-sequence tasks. However, this is very resource intensive for the latter since X-MOL does not process the input and output sequences separately: rather both are processed together as one long sequence. For vanilla Transformer neural networks, the amount of memory and computation required grows quadratically with the length of the sequence [1]. Additionally, X-MOL does not approach pre-training from a language-modelling perspective and it explores only a single pre-training task.

Taking inspiration from NLP, we aim to address the resource challenges within computational chemistry by exploiting transfer learning to provide a model which can be quickly applied to diverse tasks. The purpose of this work is therefore to use SMILES as a 'language for Chemistry' [8] to provide a common data format to which we then apply Transformer-based language models. We investigate the ability of self-supervised pre-training on a large dataset of unlabelled molecules to decrease convergence time for a number of sequence-to-sequence tasks, thereby improving the results in these tasks when training time is limited. We explore a number of self-supervised pre-training tasks and model architectures, and quantitatively compare their performance in both sequence-to-sequence and discriminative downstream tasks. We show that, with the help of transfer learning, our models can achieve state-of-the-art results on four downstream datasets. Additionally, we examine the ability of these models to fine-tune on multiple discriminative tasks simultaneously, further improving cheminformatics research efficiency.

2. Methods

Chemformer is based on the BART language model, which uses both the encoder and decoder stacks of the Transformer. This makes it very suitable for sequence-to-sequence tasks such as reaction prediction and molecular optimisation since the input sequences to the encoder and decoder are processed separately, reducing the amount of computation required in comparison to a model which processes both sequences together. The BART model can also easily be applied to discriminative tasks by using only the encoder stack.

The Chemformer models were firstly trained in a self-supervised manner and the learnable weights were saved. These weights were then loaded separately for each downstream task of interest and the task-specific fine-tuning procedure took place. Figure 1 provides an overview of how the pre-training and downstream fine-tuning tasks are applied to the Chemformer model.

To investigate the importance of the number of learnable model parameters, we pre-trained both a base model, Chemformer, and a larger model, Chemformer-Large. The Chemformer model uses the same hyperparameters as the original Transformer and contains approximately 45 million learnable weights, whereas the Chemformer-Large model expands this to 230 million weights. Full details of the models can be found in section 2.4.

2.1. Pre-training

2.1.1. Dataset

An unlabelled dataset of approximately 100 million SMILES strings was used to pre-train the models. These molecules were randomly selected from roughly 1.5 billion molecules available from the publicly accessible ZINC-15 dataset [30] with the following constraints: reactivity set to reactive, purchasability set to annotated (the most permissive option), molecular weight $\leqslant$500 Daltons and LogP (the logarithm of the n-octanol:water partition coefficient) $\leqslant$5. Train, validation and test splits were then randomly assigned, with training data taking 99% and validation and testing each assigned 0.5% of the 100 million molecules. We use only 100 million molecules for this work due to computational resource constraints.

2.1.2. Procedure

The pre-training procedure begins by converting each molecule in the batch to a non-canonical SMILES form, which corresponds to the given molecule. SMILES strings are then randomly modified, tokenised and embedded into a sequence of vectors. Sinusoidal positional embeddings [1] are added before the sequence is passed into the Transformer layers of the model. The modified sequence is passed to the bidirectional

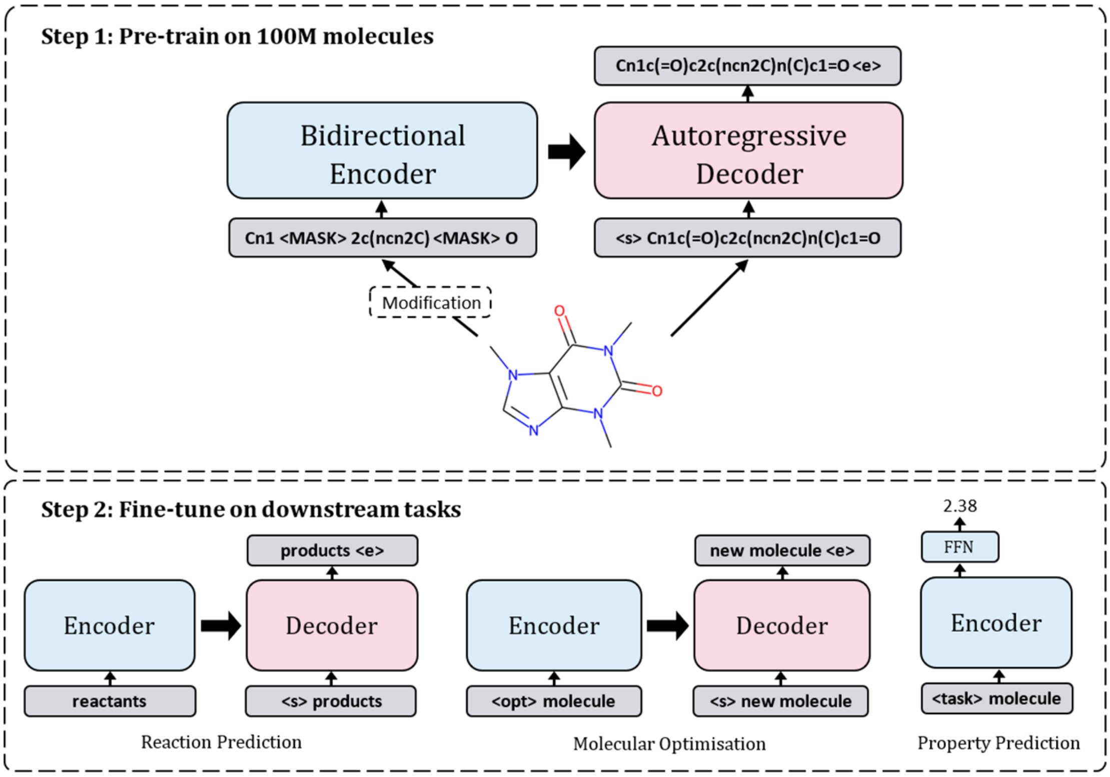


Figure 1. An illustration of the pre-training and fine-tuning procedures for downstream tasks. Here, we use the term pre-training to refer to training on a large dataset in an unsupervised manner before fine-tuning the weights of the model on the specific cheminformatics task of interest. FFN in the figure refers to a feed-forward neural network—a sequence of fully-connected layers of artificial neurons with non-linear activations.

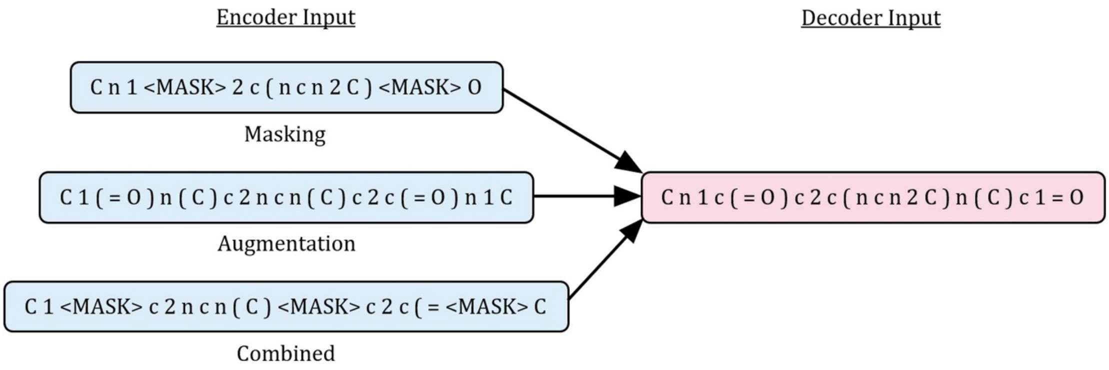

Encoder Input
Decoder Input
C n 1 <MASK> 2 c (n c n 2 C) <MASK> O
Masking
C 1 (= O) n (C) c 2 n c n (C) c 2 c (= O) n 1 C
Augmentation
C 1 <MASK> c 2 n c n (C) <MASK> c 2 c (= <MASK> C
Combined
C n 1 c (= O) c 2 c (n c n 2 C) n (C) c 1 = O

Figure 2. Comparison of the three pre-training tasks for a caffeine molecule. On the left is the modified SMILES given to the encoder, and on the right is the original SMILES, which we train the decoder to predict.

encoder, while the autoregressive decoder is asked to predict the original SMILES sequence, given the same sequence right-shifted. A fully-connected layer with linear activation is applied to the output of the decoder to produce a distribution over the model's vocabulary and a cross-entropy loss function is used to train the model.

For the base Chemformer model, we investigated three SMILES modification techniques in this work: masking, augmentation and a combination of masking and augmentation. Due to resource constraints, however, the Chemformer-Large model was pre-trained only on the combined task. Figure 2 illustrates example SMILES strings for all three pre-training tasks. Each of the tasks are implemented as follows:

- **Masking.** Masking is conducted with the span masking algorithm used by the BART [17] model—short sequences of tokens within a SMILES string are randomly replaced by a single ⟨*MASK*⟩ token.
- **Augmentation.** The augmentation task is conducted similarly to the approach of the heteroencoder model [31]; the input to the model is modified by randomly generating another SMILES string, which

corresponds to the same molecule as the output. This is carried out by following the SMILES enumeration technique [32]—permuting the atom order before generating a non-canonical SMILES form. Unlike many corruption tasks used for pre-training NLP models [16, 17], this task is specific to the SMILES language.

- **Combined.** Data for the combined task are created by first augmenting and then masking each SMILES string. This task can be seen as a method of combining pre-training techniques for both natural language and chemistry.

2.2. Sequence-to-sequence fine-tuning

After pre-training, the models were fine-tuned on downstream datasets. For this work we investigated three downstream sequence-to-sequence tasks: direct synthesis prediction, retrosynthesis prediction and molecular optimisation.

2.2.1. Datasets

For the direct synthesis prediction task we made use of the benchmark USPTO-MIT dataset [33], which contains approximately 470 000 reactions originally extracted from patents [34]. We evaluated performance on both USPTO Mixed, where reactants and reagents are assorted arbitrarily within the input string, and USPTO Separated, where reactants and reagents are split by a separator token. The USPTO-50K [35] dataset, which contains approximately 50 000 reactions, was used to benchmark Chemformer on the retrosynthesis prediction task.

Molecular optimisation aims to improve the property profile of a starting molecule towards desirable molecular properties [6, 7]. The dataset [6] for the molecular optimisation task consists of a set of matched molecular pairs (MMPs) extracted from ChEMBL [36], together with the property changes of the MMPs. Three molecular properties: LogD (the logarithm of the n-octanol:water partition coefficient at pH 7.4), solubility and clearance, are optimised simultaneously. Property values for each molecule were predicted from models built using internal experimental data. The property prediction models were used for both the construction of training data and for the evaluation of the generated molecules during testing. The dataset includes 160 831 train, 17 871 validation and 19 856 test MMPs. Full details of the dataset and the models used to generate molecular property predictions can be found in [6].

2.2.2. Procedure

Sequence-to-sequence fine-tuning is analogous to pre-training; inputs are passed to the encoder, right-shifted outputs along with the memory embeddings from the encoder are applied to the decoder and the decoder output embeddings are passed through a fully-connected layer to produce a distribution over the model's vocabulary. A cross-entropy loss function is used to train the model.

For the direct reaction prediction task, the model is given the reactants and asked to predict the products, with the reverse being true for the retrosynthesis prediction task. Fine-tuning for the molecular optimisation task is performed by prefixing the molecule to be optimised with optimisation tokens. For example, if we wish the solubility to be increased, the clearance to be decreased and the LogD to be left unchanged, we encode this into an optimisation using tokens in the model's vocabulary. There is a separate token for each property optimisation. So, in this example, we would prefix the molecule to be optimised with three tokens that respectively represent the following: *increase solubility*, *decrease clearance* and *LogD unchanged*. The model is then trained to predict the MMP output molecule given in the dataset. When evaluating Chemformer on the molecular optimisation task we pass the molecules generated by the model to the in-house property prediction model, mentioned above, to assess whether they meet the property requirements. Full details of the tokens used, the molecular optimisation task and the in-house model used can be found in [6].

In addition to our novel pre-training tasks, we introduce a novel SMILES augmentation scheme for downstream tasks, which uses a tunable augmentation probability. Given a canonical input–output pair of SMILES from the training set, $(s_{\mathrm{in}}, s_{\mathrm{out}}) \in \mathcal{D}_{\mathrm{train}}$, we randomly augment s_{in} and s_{out} independently with probability p_{aug}. For sequence-to-sequence tasks we use $p_{\mathrm{aug}} = 0.5$ throughout, unless stated otherwise. Since the augmentations do not need to be precomputed, we can augment on-the-fly, similarly to a previous study [37]. Thus, this approach has three key advantages. Firstly, the augmentation probability can be tuned. Secondly, only the canonical data needs to be stored, rather than every augmented version of the dataset. And, thirdly, the model sees a different form of the same data every epoch, regardless of the number of epochs; we conjecture that this could improve the model's ability to generalise to unseen data and improve performance, as has been observed in other studies [5, 32, 38].

2.3. Discriminative fine-tuning

In addition to sequence-to-sequence fine-tuning, we examined Chemformer's application to discriminative tasks. In particular, we fine-tuned on molecular property prediction and biological activity tasks. Since we aim to improve efficiency in cheminformatics research, and since there should be significant synergy between tasks, we trained the models to optimise for multiple tasks simultaneously—an approach known as 'multi-task learning' [21, 39]. Specifically, we trained molecular property models to solve three property prediction tasks simultaneously, and trained biological activity models to predict activity for 133 genes simultaneously, rather than having separate models for each task.

2.3.1. Datasets

The Chemformer model was applied to three molecular property datasets from MoleculeNet [40]: ESOL, Free Solvation and Lipophilicity, containing 1128, 642 and 4200 molecules, respectively. Since we are interested in optimising the model for all three tasks simultaneously, we ensure that all molecules which appear in more than one dataset appear only in the train set. From ESOL 211/1128, from Lipophilicity 16/4200 and from FreeSolvation 196/642 are simultaneously at least in two of the three datasets. After splitting the remaining molecules we end up with train, validation and test splits corresponding to 75%, 10% and 15% of the dataset, respectively. We generated 20 different random splits in these proportions. The data was preprocessed by scaling the values in the training set to be between 0 and 1 (using a min-max scaler from the SciKit-Learn library [41]). Each of the three datasets was scaled independently and the same scaling functions are used for validation and testing. Due to the size imbalance between the three datasets, the ESOL and Free Solvation datasets were upsampled by factors of 2 and 3, respectively, during training. Without upsampling, the model would train mostly on samples from the lipophilicity dataset each epoch, leading to potential imbalances between the training of the datasets.

The biological activity data were downloaded from the Exascale Compound Activity Prediction Engine (ExCAPE) database [42]. The data consists of the standardised, log-transformed activity values (pXC50 values) for chemical compounds against an array of protein targets. We selected the subset of genes from the dataset which had biological activity readings for more than 1200 compounds to have a reasonable sized train and test split for more stable performance evaluation. Additionally, we selected only genes which obtained a regression coefficient over 0.4 when a ridge regression model was applied to the compounds' Morgan fingerprints with radius 2. The full list of the 133 included genes can be found in the supplementary information (available online at stacks.iop.org/MLST/3/015022/mmedia). The final dataset contains 312 202 molecules with biological activity readings. Molecules for each gene were randomly split into train, validation and test splits of 70%, 5% and 25%, respectively.

2.3.2. Procedure

Unlike sequence-to-sequence tasks, discriminative tasks only make use of the encoder stack of the model. Firstly, the tokenised SMILES string of a molecule is prefixed with one or more task tokens—gene symbols for biological activity prediction, or molecular properties for QSAR modelling. For example, if we want to ask the model to predict the ESOL for a molecule, M, we would construct the following sequence: $\langle \mathrm{ESOL} \rangle\ M_{\mathrm{SMI}}$, where M_{SMI} is the SMILES representation of M. This sequence of tokens is passed through the model's embedding layer, followed by the model's encoder. The output vector that is aligned to the task token in the input is then passed through a small multi-layer perceptron (MLP) head to produce either a class distribution vector or a single output number for classification and regression tasks, respectively. Since we only investigated regression tasks in this work, a mean squared error loss function is applied to the MLP output for each task token. We augmented the input SMILES string for discriminative tasks with $p_{\mathrm{aug}} = 1.0$ and, as with sequence-to-sequence tasks, this augmentation was performed on-the-fly during training.

For each high-level task—biological activity prediction and molecular property prediction—Chemformer models were trained simultaneously on all subtasks. The models were then evaluated separately on each subtask, to facilitate easy comparison. Property prediction models were trained on all 20 dataset splits and an average of the evaluation results was taken. The hyperparameters which were not fixed during the pre-training phase (for example, the fine-tuning learning rate, size of the MLP head and dropout, among others) were tuned separately for each Chemformer model. Additionally, to combat overfitting, the size (number of layers, attention heads, model dimension and feed-forward dimension) of the randomly initialised model was also tuned. The full details of the tuned hyperparameters for each Chemformer model can be found in the supplementary information.

In addition to the four different base Chemformer models, we trained support vector regression (SVR) models as comparison baselines. Here, 2048-bit Morgan fingerprints with radius 2 were calculated for each molecule, and an SVR with a Tanimoto kernel was then applied. The SVR models were tuned, trained and evaluated on each subtask separately.

Table 1. A comparison of the differences in hyperparameters and number of learnable weights between the two Chemformer model sizes we investigated.

	Chemformer	Chemformer-Large
Model dimension	512	1024
Feed-forward	2048	4096
Layers	6	8
Attention heads	8	16
Parameters	45M	230M

2.4. Implementation details

The Chemformer model was implemented using the PyTorch [43] and PyTorch Lightning [44] frameworks. We used the Transformer in the *pre-norm* layout—layer normalisation [45] is applied before the attention and feed-forward blocks—and the GELU activation function [46] throughout. A comparison of the size of the Chemformer and Chemformer-Large models is shown in table 1.

Each model was pre-trained for 1000 000 steps using 4 NVIDIA V100 GPUs with a batch size of 128 molecules per GPU. The original Transformer learning rate schedule was used, along with 8000 linear warm-up steps. Pre-training took approximately 2.5 d for Chemformer and 6 d for Chemformer-Large. The one-cycle learning rate schedule [47] was used for fine-tuning, for both sequence-to-sequence and discriminative tasks. Additionally, we used the Adam optimiser [48] with the parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for both pre-training and fine-tuning on all tasks.

Chemformer's vocabulary is constructed by applying regular expression matching (we use the same regex as the Molecular Transformer [4]) to the canonical SMILES of the molecules in the ChEMBL 27 [36]. There are 523 tokens in the vocabulary in total, including 250 chemical tokens from the regex matching. There are also 200 tokens which are unused during the pre-training stage; at the fine-tuning stage some of these are replaced with task-specific tokens, such as those tokens used for biological activity prediction and molecular property prediction. The remaining tokens are either meta-tokens, such as ⟨MASK⟩ or ⟨PAD⟩, or tokens for the molecular optimisation task. Tokenisation and augmentation of SMILES was performed by extending the PySMILESUtils framework [49].

3. Results

We evaluate the performance of the Chemformer and Chemformer-Large models on three downstream sequence-to-sequence tasks: direct synthesis prediction, retrosynthesis prediction and molecular optimisation. Additionally, we investigate Chemformer's ability to train simultaneously on multiple downstream discriminative tasks. Specifically, we look at three molecular property prediction tasks and biological activity prediction for 133 genes.

3.1. Effects of transfer learning

3.1.1. Improvement in performance

Table 2 compares the downstream results for the three different pre-trained models, as well as a model with randomly initialised weights (no pre-training) on a selection of the tasks. The training time is limited to no more than 12 h for each task. In particular, this corresponds to: 40 epochs for direct reaction prediction on the USPTO Separated dataset; 500 epochs for retrosynthesis prediction on the USPTO-50K dataset; 100 epochs for the molecular optimisation task; 150 epochs of simultaneous fine-tuning on the property prediction tasks; and the same for the biological activity tasks. For sequence-to-sequence tasks, output SMILES are generated using the beam search algorithm with a beam width of 10, and the top-1 prediction is used for evaluation.

From table 2 we can see that transfer learning provides a marked improvement; pre-trained models beat the randomly initialised baseline for all datasets. We can also see that the Chemformer model pre-trained on the combined task is the strongest performer. Other than molecular optimisation, the combined model performs best on all tasks. For molecular optimisation, the model pre-trained using only masking is the best performer, while the combined model is unable to beat the model with no pre-training. We discuss possible explanations for this in more detail in section 4.

By examining the molecular property prediction tasks in more detail, we continue to see that transfer learning provides a performance boost. Table 3 outlines the results of Chemformer models for these tasks. The most significant increase in performance from transfer learning is witnessed in the lipophilicity task; the performance boost on the ESOL and free solvation datasets is more modest. The table also compares the Chemformer models against literature baselines and an SVR baseline, trained as described in section 2. The

Table 2. Results on downstream tasks for a selection of pre-training approaches when fine-tuning is limited to no more than 12 h. The Random model uses randomly initialised weights rather than weights learned during pre-training. For the molecular optimisation task we measure the percentage of generated molecules which fulfil the desirable properties. For discriminative datasets we report the mean R^2 over all of the subtasks.

	Sequence-to-sequence (%)			Discriminative (Mean R^2)	
Model	Direct	Retro	Mol opt	Mol prop	Bioactivity
Random	91.1	50.8	73.1	0.680	0.480
Mask	91.2	52.1	**75.0**	0.843	0.603
Augment	91.1	51.8	74.3	0.848	0.606
Combined	**91.8**	**53.6**	72.2	**0.857**	**0.631**

Table 3. R^2 (higher is better) and root mean square error (RMSE, lower is better) downstream molecular property prediction single-model results for baseline models, as well as Chemformer models pre-trained on different tasks. The Random model uses randomly initialised weights rather than weights learned during pre-training. Each of the Chemformer models was fine-tuned on all three molecular property subtasks simultaneously, whereas the baseline models were trained on each subtask separately.

	Lipophilicity		ESOL		Free solvation	
Model	R^2	RMSE	R^2	RMSE	R^2	RMSE
SVR	0.617	0.746	0.766	1.031	0.754	2.107
MPNN [40]	—	0.719	—	0.580	—	1.150
X-MOL [11]	—	0.596	—	0.578	—	1.108
D-MPNN [50]	—	**0.555**	—	0.555	—	1.075
MolBERT [9]	—	0.561	—	**0.531**	—	**0.948**
Random	0.398	0.946	0.855	0.803	0.786	1.887
Mask	0.736	0.621	0.903	0.657	0.889	1.366
Augment	0.738	0.618	0.904	0.652	0.901	1.287
Combined	0.754	0.598	0.910	0.633	0.908	1.230

literature baselines do not however use the same dataset splits as the Chemformer model and, as with the SVR, are trained on each subtask separately. The SVR is able to beat the randomly initialised Chemformer model on the lipophilicity task, but is otherwise outperformed by all other models across all tasks. The combined model is the best performing of all the Chemformer models in all three tasks. However, when we examine the results from previous works, we can see that Chemformer is outperformed by models from the literature. The directed message-passing neural network (D-MPNN) [50] and MolBERT [9] models outperform the combined Chemformer model in all three subtasks, and all of the literature models we have presented outperform the combined model in the ESOL and Free Solvation tasks.

Figure 3 provides a more detailed view of the results of the biological activity prediction tasks. The performance of each Chemformer model is compared to that of the SVR for each of the 133 tasks. While there is a lot of variation in the results for each gene—some tasks are challenging irrespective of the model—the improvement provided by transfer learning is clear. All three pre-trained models perform significantly better than the random initialised model and, again, the model pre-trained on the combined task is the strongest performer. However, despite the improvement provided by pre-training, none of the Chemformer models are able to beat the SVR baseline on average across all tasks. The full set of results of the 133 tasks can be found in the supplementary information.

3.1.2. Decreased convergence time

In addition to stronger performance in downstream tasks, transfer learning can significantly speed up training convergence. Figure 4 illustrates the considerable effect pre-training can have on performance and convergence speed for the retrosynthesis task. Firstly, the Chemformer model pre-trained on the combined task is able to outperform (on top-1 comparison) the existing SMILES-based state-of-the-art, the Augmented Transformer, with 20 epochs of fine-tuning. This corresponds to fewer than 30 min of training on one GPU. In addition to this, fine-tuning for 50 epochs provides a better top-1 result than 500 epochs of training from randomly initialised weights—an order of magnitude difference in training time.

3.2. Comparison with existing approaches

Allowing the model to fine-tune for longer than 12 h improves the results further for most tasks; in table 4 we compare existing direct reaction prediction implementations against Chemformer and Chemformer-Large, fine-tuned for 150 and 100 epochs, respectively. Additionally, table 5 compares the Chemformer model, fine-tuned for 500 epochs, and the Chemformer-Large model, fine-tuned for 200

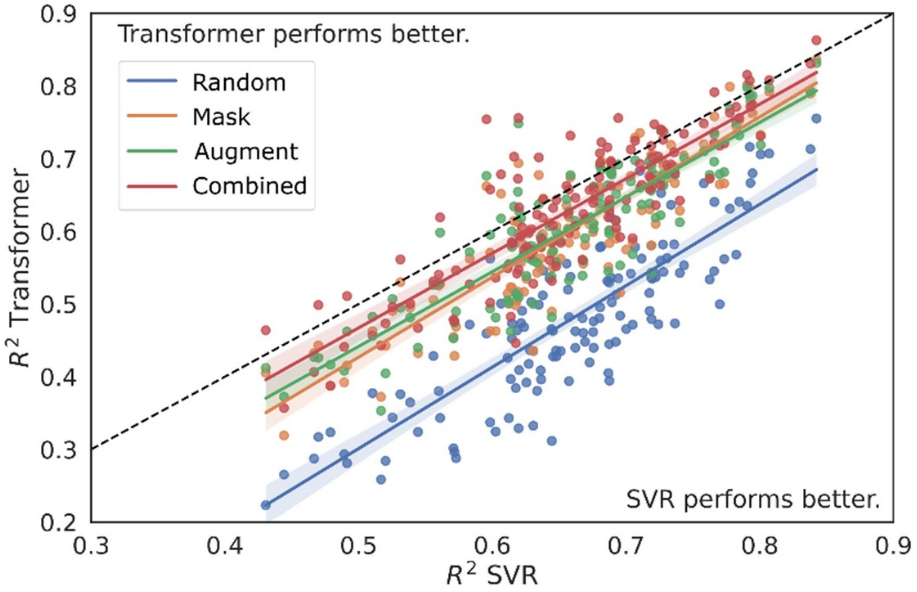


Figure 3. Comparison of the performance of Chemformer models with that of an SVR baseline across 133 bioactivity prediction tasks. Each dot corresponds to the bioactivity prediction result for a single gene. If the dot is above the dashed line the Chemformer model is a better predictor for that gene.

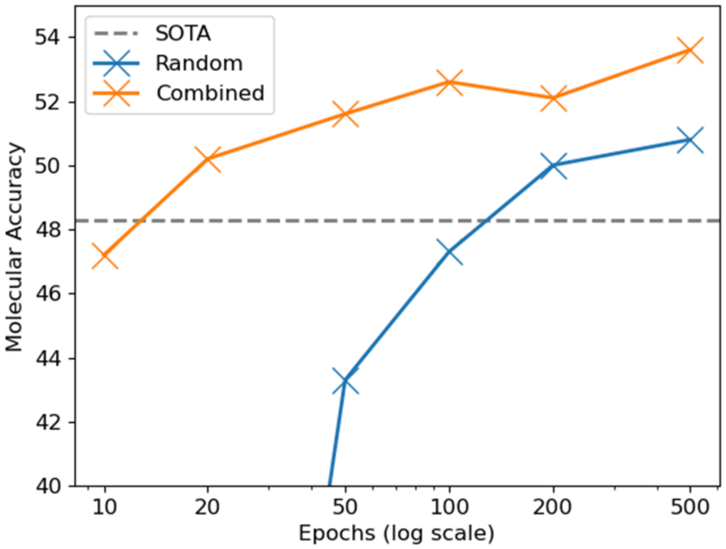


Figure 4. Comparison of the convergence on the USPTO-50K dataset of the randomly initialised Chemformer model with that of the model pre-trained on the combined task. Each point shows the result on the test dataset after a full training cycle for the specified number of epochs. The state-of-the-art (SOTA) model refers to the performance of the Augmented Transformer [5].

Table 4. The percentage of reactions predicted correctly in the forward direction from the USPTO MIT dataset. In the Mixed dataset reactants and reagents are assorted arbitrarily while, in the Separated dataset, they are separated by an otherwise unused token.

	Mixed			Separated		
Model	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10
Mol Transformer [4]	88.6	94.2	—	90.4	95.3	—
Aug Transformer [5]	90.0	**95.8**	**96.2**	91.1	**96.3**	**96.7**
Chemformer	90.9	93.8	94.1	92.5	94.9	95.1
Chemformer-Large	**91.3**	93.7	94.0	**92.8**	94.9	95.0

Table 5. The percentage of retrosynthesis reactions predicted correctly on the USPTO-50K dataset on a selection of SMILES- and graph-based approaches.

Model	Top-1	Top-5	Top-10
SMILES-based			
SCROP [51]	43.7	65.2	68.7
Two-way transformer [52]	47.1	73.1	76.3
Aug transformer [5]	48.3	**73.4**	**77.4**
Chemformer	53.6	61.1	61.7
Chemformer-Large	**54.3**	62.3	63.0
Graph-based			
MEGAN [53]	48.1	**78.4**	**86.1**
GLN [54]	52.5	75.6	83.7
GraphRetro [55]	**53.7**	72.2	75.5

Table 6. The percentage of top-1 generated molecules which fulfil the desirable properties, are matched molecular pairs and are valid, for a selection of Chemformer models and existing implementations. The pre-training tasks for the Chemformer models are shown in brackets.

Model	Desirable	MMP-33	Valid
Transformer [6]	65.2	96.0	97.3
Transformer-R [7]	70.2	**99.0**	98.4
Chemformer (*Mask*)	**75.0**	97.0	**99.9**
Chemformer (*Augment*)	74.3	97.8	**99.9**
Chemformer (*Combined*)	72.2	96.0	**99.9**
Chemformer-Large (*Combined*)	70.1	94.6	**99.9**

epochs, against existing SMILES- and graph-based approaches on the USPTO-50K retrosynthesis dataset. All Chemformer models were pre-trained on the combined pre-training task.

From the results on the forward prediction datasets and the retrosynthesis prediction dataset we can see that both Chemformer model sizes are able to outperform the existing SMILES-based state-of-the-art model on top-1 results. Chemformer-Large is also able to outperform the best graph-based models on top-1 predictions. However, the tables also show that the existing methods predict significantly more reactions correctly for top-5 and top-10 evaluation. We examine this effect in more detail in section 4.

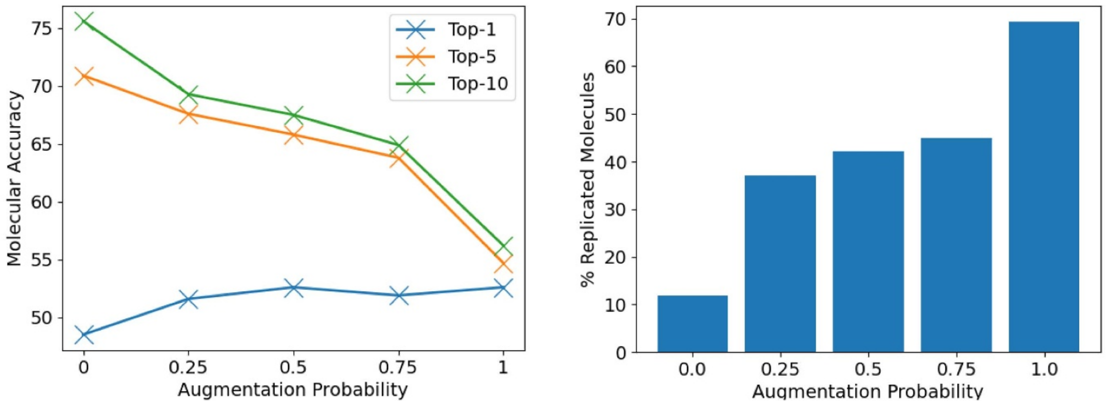
In table 6 we compare the downstream molecular optimisation performance of a number of pre-trained Chemformer models with existing implementations. In particular, we examine the performance of all three pre-training tasks with base Chemformer models, after fine-tuning for 100 epochs, along with a Chemformer-Large model (pre-trained on the combined task) fine-tuned for 80 epochs. For the Transformer [6] and Transformer-R [7] benchmarks we use the published models, but examine only top-1 performance. From the table we can see that, while all Chemformer models perform strongly in comparison to existing benchmarks, the smaller Chemformer models outperform the larger on the percentage of desirable molecules generated. The Transformer-R model, however, generates more molecules which meet the MMP-33 requirement. This metric measures the percentage of generated molecules for which, firstly, a single transformation has been applied to the starting molecule and, secondly, the ratio between the number of heavy atoms (non-hydrogen atoms) in the transformation and the number of heavy atoms in the entire molecule is not greater than 0.33. All the models we examined generated a very high proportion of valid molecules, but the Chemformer models generated slightly more than the existing approaches.

4. Discussion

The downstream results presented in section 3 show that the Chemformer model can be successfully applied to both sequence-to-sequence and discriminative tasks. The results also show that transfer learning can provide a significant boost to downstream performance and convergence speed. With the exception of the molecular optimisation task, the model pre-trained on the combined task outperforms all other Chemformer models and, in some cases, outperforms the existing state-of-the-art model. This result suggests that valuable chemical information is contained in the weights of the pre-trained Chemformer model.

4.1. Sequence-to-sequence tasks

Downstream results on sequence-to-sequence datasets show that our pre-trained Chemformer models outperform not only their randomly initialised (no transfer learning) counterparts, but also the current



(a) Top-1, 5 and 10 molecular accuracy

(b) Percentage of replicated molecules

Figure 5. Analysis of the impact of p_{aug}, the augmentation probability, on performance when fine-tuning Chemformer on USPTO-50K. The percentage of replicated molecules is determined by calculating the proportion of molecules in the top-10 which are replicated, and taking an average over the whole dataset.

state-of-the-art models for a number of tasks. Specifically, Chemformer is able to beat the existing state-of-the-art model on top-1 prediction for direct synthesis and retrosynthesis prediction, and is able to produce more desirable molecules than existing approaches in the molecular optimisation task. However, for molecular optimisation, our Chemformer models used beam search (with a beam width of 10) to generate output molecules, while the Transformer and Transformer-R benchmarks used greedy search. Furthermore, these models contain fewer learnable parameters than Chemformer and use a different augmentation strategy. The randomly initialised Chemformer model is able to generate more desirable molecules than the baselines, suggesting that the performance of these existing models would improve with different sampling or augmentation techniques, or with more parameters.

While the Chemformer model pre-trained on the combined task performed strongest in the reaction informatics tasks, the model pre-trained with only the masking task performs best on the molecular optimisation dataset. This is a surprising result since, for the combined task, the model is required to solve both the masking and the augmentation pre-training tasks. One possible explanation for this is that the combined task model overfits quickly in the molecular optimisation task. This is supported by the lower performance of the Chemformer-Large model in the same task. We therefore conjecture that with further hyperparameter tuning the performance of the combined task model could be improved. In particular, more work is needed to determine the optimal number of epochs required for fine-tuning.

When comparing the performance in the forward synthesis and retrosynthesis prediction tasks, we noted that the augmentation approach we employed resulted in stronger top-1 performance, but that the top-5 and top-10 performances were weaker than the existing methods. By analysing the output of the beam search we found that the proportion of augmented forms of the same molecule in the beam outputs was significantly larger in the models trained with augmentation than the model trained without augmentation. Figure 5(a) shows how the augmentation probability affects the top-1, top-5 and top-10 molecular accuracy for the USPTO-50K retrosynthesis prediction task. Fine-tuning with no augmentation provides the lowest top-1 performance but the highest top-5 and top-10 performances. Fine-tuning with $p_{\text{aug}} \in \{0.25, 0.5, 0.75, 1.0\}$ all lead to comparable top-1 performance, but top-5 and top-10 performances steadily decrease as p_{aug} is increased. Figure 5(b) provides an explanation for this effect by examining the percentage of the ten beam outputs that contain an augmented SMILES form of the same reactants. Filling the beam outputs with augmented forms results in a lower diversity when SMILES are converted back to molecules; this causes the top-5 and top-10 results to converge towards the top-1. Our augmentation strategy therefore creates a trade-off between an improvement in top-1 performance with a decrease in top-5 and top-10.

To combat the detrimental effect of augmentation on top-5 and top-10 results the beam width could be increased significantly. This would essentially counter the reduction in molecular diversity by sampling more molecules. However, the amount of computation required scales linearly with the beam width; increasing the beam width from 10 to 50 would require five times as much computational resource. Alternatively, Levenshtein augmentation [38] could be used to ensure the input and generated SMILES sequences are similar. This would reduce the likelihood of many SMILES forms being generated—specifically those which are dissimilar to the input—therefore improving molecular diversity.

4.2. Discriminative tasks

In addition to fine-tuning on sequence-to-sequence tasks, we have shown that it is possible to train Chemformer simultaneously on multiple discriminative tasks. For both the molecular property prediction tasks and the biological activity prediction tasks, the Chemformer model pre-trained on the combined task shows strong performance in comparison to randomly initialised and SVR baselines, but is consistently outperformed by existing implementations.

While the three pre-trained Chemformer models perform comparably in both sets of discriminative tasks, the randomly initialised model performs significantly worse. A possible explanation for this is that, without transfer learning, Chemformer overfits quickly on small datasets—the number of molecules per biological activity prediction task varies from 1241 to 5830, and the number of molecules per molecular property is no more than 4200. This explanation is supported by the observation that, for biological activity prediction, the optimal architecture found for the randomly initialised model used only 6 million learnable parameters, in comparison to almost 20 million for the pre-trained models. The randomly initialised model for property prediction uses even fewer parameters. Larger randomly initialised models were found to perform worse. These results suggest that, in small-data regimes, pre-training is crucial for strong performance with Transformer models in discriminative tasks.

In the molecular property prediction tasks the results show that pre-trained Chemformer models are able to outperform the SVR baseline. In biological activity prediction, however, even the best performing Chemformer model, the combined task model, shows marginally lower performance than the SVR. The SVR models are, however, trained on each activity prediction task separately; meaning 133 models need to be maintained and productionised, in comparison with a single Chemformer encoder. The performance of the SVR models may benefit from this separation, but more work is needed to determine the extent of this performance improvement. For both sets of discriminative tasks, more experimentation is also required to compare our models with existing baselines, including the use of additional molecular fingerprinting algorithms.

Previous works [9–14] have also attempted to use Transformers for molecular property prediction, and others have used graph neural networks [40, 50]. As shown in section 3, many of these works publish stronger results than Chemformer on the three MoleculeNet tasks we investigated. It is, however, difficult to make a direct comparison since Chemformer is trained simultaneously on multiple subtasks, whereas the literature models are trained and tuned separately. The use of multi-task learning also leads to different dataset splits for the molecular property prediction tasks. Nonetheless, it is clear that existing implementations perform better than the best Chemformer models in discriminative tasks, and we propose a number of possible explanations for this. Firstly, there is a significant difference in the number of learnable parameters between the models; the MolBERT model contains approximately 85 millions parameters compared to fewer than 20 million for the Chemformer encoder. Secondly, pre-training the BART architecture may be better suited to sequence-to-sequence tasks than discriminative tasks. Additionally, using only the pre-trained encoder may not be the best way of making use of the BART model for these tasks. Finally, unlike MolBERT, Chemformer's pre-training does not consider molecular properties. Extending the pre-training objective with molecular property prediction tasks may help to improve downstream discriminative performance. Scaling up the size of the Chemformer model and experimenting with additional pre-training objectives is something we intend to investigate in future work.

5. Conclusion

In this work we introduced the Chemformer model, which makes use of the SMILES language for application to diverse computational chemistry tasks. We investigated three different self-supervised pre-training techniques and applied these to a large dataset of unlabelled SMILES. Finally, we fine-tuned the pre-trained Chemformer models on a selection of downstream tasks and compared their performance to randomly initialised models and existing benchmarks.

From the fine-tuning results we presented, three key conclusions can be drawn. Firstly, the Chemformer model can be applied to a wide variety of downstream tasks, including both sequence-to-sequence and discriminative tasks, fairly easily. Secondly, self-supervised pre-training can improve convergence of the Chemformer model on downstream Cheminformatics tasks, and can therefore significantly improve results of these tasks when training time is limited. Finally, a combination of transfer learning and our novel augmentation strategy is able to produce state-of-the-art top-1 results in all the downstream sequence-to-sequence tasks we examined.

Given its ability to quickly fine-tune on both sequence-to-sequence and discriminative cheminformatics tasks, the proposed Chemformer model is a significant step towards a generally applicable deep learning model for computational chemistry.

Data availability statement

The data that support the findings of this study will be openly available following an embargo at the following URL/DOI: https://github.com/MolecularAI/Chemformer. Data will be available from 16 November 2021.

ORCID iD

Esben Jannik Bjerrum https://orcid.org/0000-0003-1614-7376

References

[1] Vaswani A *et al* 2017 Attention is all you need *31st Annual Conf. on Neural Information Processing Systems (NIPS)*

[2] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80

[3] Cho K *et al* 2014 Learning phrase representations using RNN encoder–decoder for statistical machine translation *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)* pp 1724–34

[4] Schwaller P *et al* 2019 Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction *ACS Cent. Sci.* **5** 1572–83

[5] Tetko I V, Karpov P, Van Deursen R and Godin G 2020 State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis *Nat. Commun.* **11** 1–11

[6] He J *et al* 2021 Molecular optimization by capturing chemist's intuition using deep neural networks *J. Cheminform.* **13** 1–17

[7] He J *et al* 2021 Transformer neural network for structure constrained molecular optimization *ChemRxiv Preprint* (available at: https://doi.org/10.26434/chemrxiv.14416133.v1)

[8] Weininger D 1988 Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules *J. Chem. Inf. Comput. Sci.* **28** 31–36

[9] Fabian B *et al* 2020 Molecular representation learning with language models and domain-relevant auxiliary tasks (arXiv:2011.13230)

[10] Chithrananda S, Grand G and Ramsundar B 2020 ChemBERTa: large-scale self-supervised pretraining for molecular property prediction (arXiv:2010.09885)

[11] Xue D *et al* 2020 X-MOL: large-scale pre-training for molecular understanding and diverse molecular analysis *bioRxiv Preprint* (available at: https://doi.org/10.1101/2020.12.23.424259)

[12] Wang S, Guo Y, Wang Y, Sun H and Huang J 2019 SMILES-BERT: large scale unsupervised pre-training for molecular property prediction *Proc. 10th ACM Int. Conf. on Bioinformatics, Computational Biology and Health Informatics* pp 429–36

[13] Zhang X-C *et al* 2021 MG-BERT: leveraging unsupervised atomic representation learning for molecular property prediction *Brief. Bioinform.* **22** bbab152

[14] Maziarka Ł *et al* 2020 Molecule attention transformer (arXiv:2002.08264)

[15] Ross J *et al* 2021 Do large scale molecular language representations capture important structural information? (arXiv:2106.09553)

[16] Devlin J, Chang M-W, Lee K and Toutanova K 2018 BERT: pre-training of deep bidirectional transformers for language understanding (arXiv:1810.04805)

[17] Lewis M *et al* 2020 BART: denoising sequence-to-sequence pre-training for natural language generation, translation and comprehension *Proc. 58th Annual Meeting of the Association for Computational Linguistics* pp 7871–80

[18] Radford A, Narasimhan K, Salimans T and Sutskever I 2018 Improving language understanding by generative pre-training (available at: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf) (Accessed 24 January 2021)

[19] Radford A, Wu J, Child R, Luan D, Amodei D and Sutskever I 2019 Language models are unsupervised multitask learners (available at: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf) (Accessed 24 January 2022)

[20] Dong L *et al* 2019 Unified language model pre-training for natural language understanding and generation (arXiv:1905.03197)

[21] Raffel C *et al* 2020 Exploring the limits of transfer learning with a unified text-to-text transformer *J. Mach. Learn. Res.* **21** 1–67

[22] Bai R *et al* 2020 Transfer learning: making retrosynthetic predictions based on a small chemical reaction dataset scale to a new level *Molecules* **25** 2357

[23] Ishiguro K, Ujihara K, Sawada R, Akita H and Kotera M 2020 Data transfer approaches to improve seq-to-seq retrosynthesis (arXiv:2010.00792)

[24] Wang L, Zhang C, Bai R, Li J and Duan H 2020 Heck reaction prediction using a transformer model based on a transfer learning strategy *Chem. Commun.* **56** 9368–71

[25] Kreutter D, Schwaller P and Reymond J-L 2021 Predicting enzymatic reactions with a molecular transformer *Chem. Sci.* **12** 8648–59

[26] Zhang Y *et al* 2021 Data augmentation and transfer learning strategies for reaction prediction in low chemical data regimes *Org. Chem. Front.* **8** 1415–23

[27] Pesciullesi G, Schwaller P, Laino T and Reymond J-L 2020 Transfer learning enables the molecular transformer to predict regio- and stereoselective reactions on carbohydrates *Nat. Commun.* **11** 1–8

[28] Li X and Fourches D 2020 Inductive transfer learning for molecular activity prediction: *next-gen* QSAR models with MolPMoFiT *J. Cheminform.* **12** 1–15

[29] Karpov P, Godin G and Tetko I V 2020 Transformer-CNN: Swiss knife for QSAR modeling and interpretation *J. Cheminform.* **12** 1–12

[30] Sterling T and Irwin J J 2015 Zinc 15–ligand discovery for everyone *J. Chem. Inf. Model.* **55** 2324–37

[31] Bjerrum E J and Sattarov B 2018 Improving chemical autoencoder latent space and molecular *de novo* generation diversity with heteroencoders *Biomolecules* **8** 131

[32] Bjerrum E J 2017 SMILES enumeration as data augmentation for neural network modeling of molecules (arXiv:1703.07076)

[33] Jin W, Coley C W, Barzilay R and Jaakkola T 2017 Predicting organic reaction outcomes with Weisfeiler–Lehman network *Proc. 31st Int. Conf. on Neural Information Processing Systems* pp 2604–13

[34] Schneider N, Lowe D M, Sayle R A, Tarselli M A and Landrum G A 2016 Big data from pharmaceutical patents: a computational analysis of medicinal chemists' bread and butter *J. Med. Chem.* **59** 4385–402

[35] Schneider N, Stiefl N and Landrum G A 2016 What's what: the (nearly) definitive guide to reaction role assignment *J. Chem. Inf. Model.* **56** 2336–46

[36] Mendez D *et al* 2019 ChEMBL: towards direct deposition of bioassay data *Nucleic Acids Res.* **47** D930–40

[37] Kotsias P-C *et al* 2020 Direct steering of *de novo* molecular generation with descriptor conditional recurrent neural networks *Nat. Mach. Intell.* **2** 254–65

[38] Sumner D, He J, Thakkar A, Engkvist O and Bjerrum E J 2020 Levenshtein augmentation improves performance of SMILES based deep-learning synthesis prediction *ChemRxiv* (available at: https://doi.org/10.26434/chemrxiv.12562121.v2)

[39] Ruder S 2017 An overview of multi-task learning in deep neural networks (arXiv:1706.05098)

[40] Wu Z *et al* 2018 MoleculeNet: a benchmark for molecular machine learning *Chem. Sci.* **9** 513–30

[41] Pedregosa F *et al* 2011 Scikit-learn: machine learning in python *J. Mach. Learn. Res.* **12** 2825–30

[42] Sturm N *et al* 2020 Industry-scale application and evaluation of deep learning for drug target prediction *J. Cheminform.* **12** 1–13

[43] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* vol 32, ed H Wallach *et al* (Curran Associates, Inc.) pp 8024–35

[44] Falcon W A 2019 PyTorch lightning *GitHub* vol 3 (available at: https://github.com/PyTorchLightning/pytorch-lightning)

[45] Ba J L, Kiros J R and Hinton G E 2016 Layer normalization (arXiv:1607.06450)

[46] Hendrycks D and Gimpel K 2016 Gaussian error linear units (GELUs) (arXiv:1606.08415)

[47] Smith L N and Topin N 2019 Super-convergence: very fast training of neural networks using large learning rates *Proc. SPIE* **11006** 1100612

[48] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)

[49] Bjerrum E, Rastemo T, Irwin R, Kannas C and Genheden S 2021 PySMILESUtils–enabling deep learning with the SMILES chemical language *ChemRxiv Preprint* (available at: https://doi.org/10.33774/chemrxiv-2021-kzhbs)

[50] Yang K *et al* 2019 Analyzing learned molecular representations for property prediction *J. Chem. Inf. Model.* **59** 3370–88

[51] Zheng S, Rao J, Zhang Z, Xu J and Yang Y 2019 Predicting retrosynthetic reactions using self-corrected transformer neural networks *J. Chem. Inf. Model.* **60** 47–55

[52] Kim E, Lee D, Kwon Y, Park M S and Choi Y-S 2021 Valid, plausible and diverse retrosynthesis using tied two-way transformers with latent variables *J. Chem. Inf. Model.* **61** 123–33

[53] Sacha M, Błaż M, Byrski P, Włodarczyk-Pruszyński P and Jastrzebski S 2020 Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits (arXiv:2006.15426)

[54] Dai H, Li C, Coley C W, Dai B and Song L 2020 Retrosynthesis prediction with conditional graph logic network (arXiv:2001.01408)

[55] Somnath V R, Bunne C, Coley C W, Krause A and Barzilay R 2020 Learning graph models for template-free retrosynthesis (arXiv:2006.07038)