Chemical transformer compression for accelerating both training and inference of molecular modeling

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 045009 https://doi.org/10.1088/2632-2153/ac99ba

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
13 May 2022

REVISED
23 September 2022

ACCEPTED FOR PUBLICATION
12 October 2022

PUBLISHED
31 October 2022

PAPER

Chemical transformer compression for accelerating both training and inference of molecular modeling

Yi Yu and Karl Börjesson*

Department of Chemistry and Molecular Biology, University of Gothenburg, Kemivägen 10, 412 96 Gothenburg, Sweden
* Author to whom any correspondence should be addressed.

E-mail: karl.borjesson@gu.se

Keywords: model compression, transformer, molecular modeling

Supplementary material for this article is available online

Abstract

Transformer models have been developed in molecular science with excellent performance in applications including quantitative structure-activity relationship (QSAR) and virtual screening (VS). Compared with other types of models, however, they are large and need voluminous data for training, which results in a high hardware requirement to abridge time for both training and inference processes. In this work, cross-layer parameter sharing (CLPS), and knowledge distillation (KD) are used to reduce the sizes of transformers in molecular science. Both methods not only have competitive QSAR predictive performance as compared to the original BERT model, but also are more parameter efficient. Furthermore, by integrating CLPS and KD into a two-state chemical network, we introduce a new deep lite chemical transformer model, DeLiCaTe. DeLiCaTe accomplishes 4× faster rate for training and inference, due to a 10- and 3-times reduction of the number of parameters and layers, respectively. Meanwhile, the integrated model achieves comparable performance in QSAR and VS, because of capturing general-domain (basic structure) and task-specific knowledge (specific property prediction). Moreover, we anticipate that the model compression strategy provides a pathway to the creation of effective generative transformer models for organic drugs and material design.

1. Introduction

By silico modeling and analysis of chemical structures, molecular computational approaches have facilitated the development in various fields, such as drug discovery and material design [1–3]. Nowadays, deep learning methods have made significant breakthroughs in these fields [4–6]. To go from a chemical structure to a computational descriptor, molecules are encoded by different representations (figure 1(A)) [7], such as the simplified molecular input line entry specification (SMILES) [8, 9] or 2D undirected cyclic graphs [10]. Then, a suitable neural network is designed and trained to connect the molecular representation to the output related to the intended tasks, such as predictive and generative ones. It is essential to introduce a suitable model that are optimize for one's purposes. On the one hand, discriminative models could be utilized for predicting physicochemical properties, reaction performance and bioactivity [11]. One the other hand, generative models could design novel molecules efficiently and automatically with respect to specific objectives [12–14].

In the last few years, transformer models have shown to be an efficient deep learning method within chemical science [15]. They have formed a new paradigm and works by self-supervised learning as the pre-training step. This self-supervised learning is based on large unlabeled chemical sequence or graph datasets. Then the models are fine-tuned in downstream tasks [16]. More concretely, models could learn basic molecular structure knowledge (like chemical formula, bonds and charges) in the pre-training process. Then this knowledge is transferred in the fine-tuning to overcome data scarcity for specific tasks, such as quantitative structure–activity relationship (QSAR) [17–21], virtual screening (VS) [22], de novo design [14], reaction prediction [16], molecular optimization [23] and drug-drug interaction predictions [24].

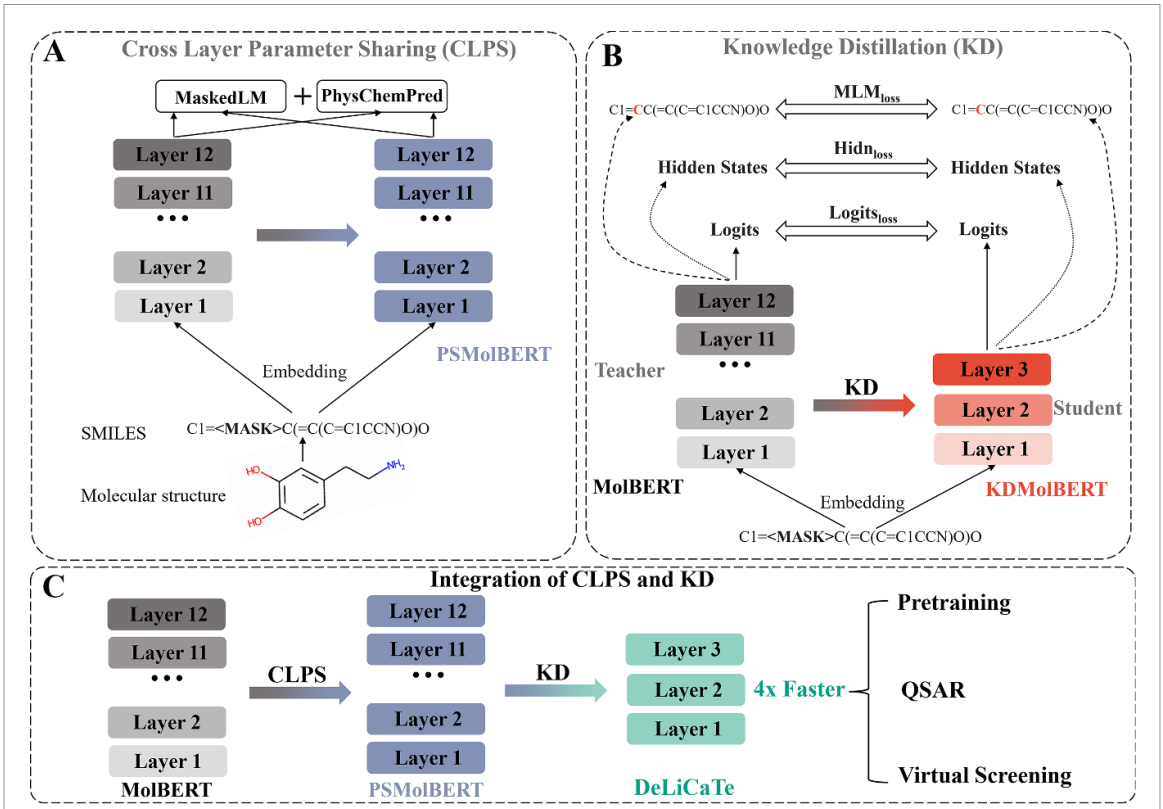

Figure 1. An illustration of the processes of the CLPS, KD methods, as well as their implementation with MolBERT forming DeLiCaTe. (A) Molecule are encoded by SMILES representation. Next, the PSMolBERT is pretrained based on SMILES which passes into embedding and 12 parameter-sharing attention layers in turn. Then, the loss is calculated by MaskedLM and PhysChemPred as semi-supervised tasks [22]. The same color depth of layers in PSMolBERT represents the parameter sharing feature in-between layers of CLPS, while the different color depth in MolBERT symbolizes that the parameters are different among layers. (B) The KDMolBERT is obtained by general distillation in which the pretrained MolBERT is acted as a teacher model. The student model, KDMolBERT, is distilled to reproduce the behavior of the teacher model, the pretrained MolBERT. The behavior is evaluated by triple losses, including logits, hidden states and MaskedLM. (C) The DeLiCaTe model integrates the advantages of the two mentioned method. It is distilled from the pretrained PSMolBERT, and a 4× speedup is realized in both training and inference compared with MolBERT.

Despite being high performance, given the size of at least several ten million parameters, transformer models are computationally expensive in both training and inference processes [25]. For instance, recently proposed models require several days of pre-training with at least four high performance graphics processing units (GPUs) [16, 24]. Furthermore, the inference time in the fine-tuning process is much longer than in other models [26]. Besides being time consuming, the environmental cost of the hardware running should be considered as another disadvantage. Additionally, data shortage is common in the fields of drug and material science, which obstacles the use of large models such as transformers. Hence, the growing computational and hardware requirements, as well as data shortage, are likely to hinder the wide practical use of transformer models [27].

In order to improve the parameter-efficiency, model compression has recently been an active research area in the fields of computer vision (CV) and natural language processing (NLP). By compression, large models, like transformers, could be scaled down and the training or inference processes are then accelerated. Cross-layer parameter sharing (CLPS) is one strategy to decrease the computational cost of the training process, while retaining the high performance of downstream tasks [27, 28]. In CLPS, the default decision is to share all parameters across layers. In other words, parameters are tied across positions and time steps. Additionally, knowledge distillation (KD) is reported to result in lighter models with faster inference [26, 29]. In KD, a student model with small size is trained to reproduce the behavior of the teacher model. Nonetheless, neither of them has been utilized in transformer models within chemical science.

Inspired by model compression within CV and NLP applications [30–35], we aim to explore more parameter-efficient transformer models for molecular modeling based on SMILES-based molecular representation. The concept is illustrated in figure 1. We begin by using CLPS or KD strategies independently to compress a high-performance chemical transformer, MolBERT [22]. Then the training and inference time is compared among the models, as well as the model sizes. Next, we show that QSAR performances of the two compressed models are competitive on various tasks, compared with MolBERT. Finally, given the positive effects above, the CLPS and KD are integrated with each other to establish a new deep lite

chemical transformer model, DeLiCaTe (shown in figure 1(C)). DeLiCaTe achieves more than 96% of the performance of MolBERT in QSAR as well as VS applications, while being 4× faster in the rates of training and inference due to a ∼10- and 3-times reduction of the number of parameters and layers, respectively.

2. Methods

2.1. Dataset for model compression

We used the GuacaMol benchmark dataset [12] from ChEMBL [36] containing ∼1.6 M compounds for pre-training or distillation. The setting was used for the pre-training dataset in MolBERT as well [22]. For the dataset, the ratio of training to validation was 16:1.

2.2. CLPS

MolBERT is a bidirectional chemical model derived from the BERT architecture [22]. It is a well-recognized molecular transformer with state-of-the-art QSAR and VS performance. The backbone of the CLPS model (PSMolBERT) uses a transformer encoder with gelu activation function, similar to MolBERT. The default decision is to share all the parameters across the layers. In other words, the weights among different layers are the same. The method for pre-training PSMolBERT is shown in figure 1(A). The pre-training tasks include masked language modeling (MaskedLM) and calculated molecular descriptor prediction (PhysChemPred), which is consistent with the original MolBERT [22]. The first one is the similar self-supervised learning method to NLP, while the latter one predicts the normalized set of chemical descriptors for each molecule. The general-domain chemical knowledge is well-learned by these two tasks. Following this method, the parameter-efficiency of CLPS was studied by directly comparing the modeling performance with MolBERT and other baseline methods.

2.3. Knowledge distillation

Regarding the mechanism of KD, as shown in figure 1(B), a student model is trained to reproduce the behavior of the teacher model. The behavior is evaluated by the loss function which is a linear combination of three types of losses:

$$L = L_{\text{mlm}} + L_{\text{hidn}} + L_{\text{logits}} \tag{1}$$

where L is the final training loss, L_{mlm} is the MaskedLM loss in the self-supervised learning process, and L_{hidn} and L_{logits} are the losses of hidden state and logits between the student and teacher model calculated by the mean square error. Logits is the raw (non-normalized) scores for classification or regression model before softmax. While hidden states represent the hidden representation of each token in each input sequence of the batch. When taking into account these two losses, the student model will learn the distribution of the teacher one. Then, the generalization of the student model can be improved. As for the loss of logits, following Hinton *et al* [35], the softmax-temperature method was used and shown below,

$$p_i = \frac{\exp\left(z_i/T\right)}{\sum_j \exp\left(z_j/T\right)} \tag{2}$$

where T controls the smoothness of the output distribution and z_i or z_j is the model score for the class i or j, respectively. The same temperature T is used to the student and the teacher when training. However, T is set to 1 to recover a standard softmax at inference. Here, we find that $T = 8$ performs well through trial-and-errors, as well as referring previous work [35]. Meanwhile, the ablation study for KD is conducted by removing any one of the triple losses.

The teacher model here was the pre-trained MolBERT with 12 transformer layers, while the student only had three layers of which the weights were randomly initialized. The other structures between these two models remained the same and are shown in supplementary table 1.

3. The DeLiCaTe model

In order to obtain DeLiCaTe, firstly, PSMolBERT was first pre-trained by CLPS. Then, the PSMolBERT model was distilled to obtain DeLiCaTe (figure 1(C)). The detail parameter and architectures are shown in section 3.1.

3.1. Experiments, baseline and evaluation

The performances of the aforementioned models were evaluated in QSAR and VS experiments. In addition, three baseline methods were used: (a) RDKit descriptors [37], (b) Extended Connectivity Fingerprints with a

Table 1. Overview of the ten different QSAR datasets.

Dataset	Acronym	Number of molecules	Reference	Type of tasks
Human β-secretase 1 inhibitors	BACE	1483	[39]	Classification
Blood–brain barrier penetration	BBBP	1879	[39]	Classification
Inhibition of HIV replication	HIV	41 101	[39]	Classification
Mutagenicity	AMES	6130	[40]	Classification
Endocrine disruptors	Eds	817	[41]	Classification
Aqueous solubility	ESOL	1128	[39]	Regression
Free solvation energy	FreeSolv	642	[39]	Regression
Lipophilicity	Lipo	4200	[39]	Regression
Epidermal growth factor inhibition	EGFR	4113	[42]	Regression
Fibroblast growth factor receptor	FGFR1	4177	[43]	Regression

diameter of 4 (ECFP4), and (iii) continuous and data-driven descriptors (CDDD) [38], which is a recurrent neural network model that today achieve state-of-art performance in QSAR and VS experiments.

Ten QSAR tasks were selected to compare the performance of the different models. The QSAR datasets were taken from MoleculeNet [39] and other sources in which one half of them are classification tasks and the other half are regression tasks. Table 1 summarizes the description of ten datasets. As for the classification tasks, they are datasets containing active and inactive labeled molecules for specific targets. Regarding the regression tasks, the first three tasks represent basic physicochemical properties for molecules, which are essential to successful drug design considering drug metabolism. For example, the lipophilicity for a molecule should be generally less than 5 to ensure drug absorption by cells. In addition, the last two datasets are kinase inhibitors for specific anti-cancer targets with corresponding molecular bioactivities. The data splitting method followed the strategy provided from ChemBench in which the splitting ratio was 80% on training, 10% on validation, and 10% on testing. Then, a 3-fold cross validation was conducted. Finally, the area under the receiver operating characteristic (ROC-AUC) and precision-recall AUC-ROC (PR-AUC) values were used as the metrics for classification tasks. Additionally, coefficients of determination (r^2) and root-mean-square deviation (RMSE) were used in regression tasks.

The VS was conducted on 69 datasets of which each one represents an individual protein target and contains a small number of active molecules amongst a much larger number of inactive ones. The benchmarking protocol by Riniker *et al* [44] was followed. Then, the ROC-AUC was used as the metrics to report the result.

3.2. Implementation and hardware

The models were implemented by PyTorch [45] and Hugging Face Transformers [46], and TextBrewer [47] was adapted for chemical model distillation. Additionally, an Adam optimizer was used for both the pre-training and fine-tuning processes. The vocabulary size for all the chemical transformer models was 42. One NVIDIA RTX 3080 was used for all the training.

4. Results and discussion

4.1. Configurations and speeds of models

The main differences among the chemical transformer models in this study are the number of parameters and attention layers. Table 2 outlines the used configurations and relative rates of models. As for the CLPS method, the number of parameters of the new model (parameter sharing MolBERT (PSMolBERT)) was compressed into ∼12% of the original one, which accelerates pre-training with 1.3× per epoch. Meanwhile, according to the validation loss curve (shown in supplementary figure 1), the loss function of PSMolBERT converges much faster than MolBERT. This phenomenon has previously been observed within the NLP field, where parameter-sharing has an effect on stabilizing network parameters [28]. Hence, it is hypothesized that the pre-training could be significantly accelerated, not only in the rate per epoch but also by a fewer number of epochs for convergence. However, even though the number of parameters is tremendously decreased, the inference is not speeded up, which is consistent with previous work in NLP [27, 28]. As for the KD method, the 3-layer KDMolBERT was distilled from the original 12-layer MolBERT, while other settings were left unchanged. Given the significantly reduced number of transformer layers, a 3.8× speedup on inference was achieved. DeLiCaTe was distilled from PSMolBERT. It is ∼4× faster on both training and inference compared with the original model. The reduced number of parameters obtained by CLPS and layers obtained by KD significantly speeds up the pre-training and inference processes, respectively. In other words,

Table 2. The configurations and training/inference time of the chemical transformer models.

Model	Parameters (M)	Training or KD time[a] (h)	Time/epoch (h)	Layers	Inference speedup
MolBERT	86	192 (100)[b]	1.92	12	1.0
PSMolBERT	8	44 (30)[b]	1.47	12	1.0×
KDMolBERT	21	12.5 (13)[b]	0.96	3	3.8×
DeLiCaTe	8	4 (6)[b]	0.67	3	3.8×

[a] The value given for MolBERT and PSMolBERT relates to the pre-training time, whereas the value given for KDMolBERT and DeLiCaTe represent the KD time.

[b] The values in parentheses represent the number of epochs for pre-training or KD. The epoch training time was calculated as the ratio between the training time and number of epochs.

Table 3. The area under the receiver characteristic curve (ROC-AUC) for classification datasets (the higher is better, and best scores are highlighted with bold font).

Method[a]	BACE	BBBP	HIV	AMES	EDC	Avg
RDKit	0.844	0.757	0.776	0.801	0.853	0.807
ECFP4	0.855	0.749	0.768	0.783	0.836	0.798
CDDD	0.832	0.823	0.771	0.807	0.872	0.821
MolBERT	**0.907**	**0.910**	**0.830**	**0.894**	**0.941**	**0.896**
PSMolBERT	0.891	0.904	0.823	0.879	0.917	0.883

[a] The standard mean errors are shown in supplementary table 2.

Table 4. Coefficient of determination (r^2) for regression datasets (the higher is better, and best scores are highlighted with bold font).

Method[a]	ESOL	FreeSov	Lipop	EGFR	FGFR1	Avg
RDKit	0.870	0.795	0.729	0.659	0.666	0.744
ECFP4	0.843	0.738	0.738	0.625	0.641	0.726
CDDD	**0.920**	**0.834**	**0.797**	0.671	0.715	0.787
MolBERT	0.905	0.816	0.780	**0.720**	**0.730**	**0.790**
PSMolBERT	0.892	0.832	0.758	0.710	0.721	0.784

[a] The standard mean errors are shown in supplementary table 3.

it combines the advantages of CLPS and KD. In the next sections we will discuss the performance of achieved models, in relation to QSAR and VS tasks.

4.2. Effect of CLPS on model performance

4.2.1. Retained QSAR performance with CLPS

Having settled the model size and rate of training using the CLPS method, we now turn our attention to the performance. This was done by comparing the predictive capacity in QSAR of PSMolBERT and MolBERT as well as with other baseline methods, including RDKit descriptors, ECFP4 and CDDD [38]. Tables 3 and 4 outline the results for classification (AUC-ROC as the metric) and regression (r^2 as the metric) tasks, respectively. As for classification tasks, MolBERT pre-trained with 100-epoch performs the best among all models, which is in agreement with the results from previous work [22]. Meanwhile, PSMolBERT pre-training with 30-epoch is not far behind MolBERT and outperforms traditional methods. Quantitatively, it retains 98.5% of the performance of MolBERT. In addition, PR-AUC has been utilized as the other metric of the performance as well and shown in supplementary table 4. It is a better indicator for imbalanced dataset (here the HIV dataset is imbalanced). The ratio of the PR-AUCs between PSMolBERT and MolBERT is 98.9%, which further indicates the retained performance of the compressed model.

According to the result in table 4 for regression tasks, the trends are very similar to the ones for classification. PSMolBERT is able to compete with MolBERT for regression modeling, achieving 99.2% of the performance of MolBERT on average. In addition, the only difference compared to classification tasks is that the performance of CDDD now is comparable to two chemical transformer models. Among the individual tasks, CDDD performs the best in aqueous solubility (ESOL), FreeSov and Lipop, while the two transformer models do better in epidermal growth factor inhibition (EGFR) and fibroblast growth factor receptor (FGFR1). It indicates that CDDD could extract physicochemical correlations from SMILES more efficiently, while transformer-based methods could perform better on bioactivity tasks, which is also found in previous literature [48]. Due to three regression tasks for physicochemical prediction, PSMolBERT is slightly worse than the CDDD. However, the result in table 4 still indicates that CLPS methods not only accelerate the modeling rate, but also retain the QSAR performance compared with original MolBERT. Additionally, RMSE

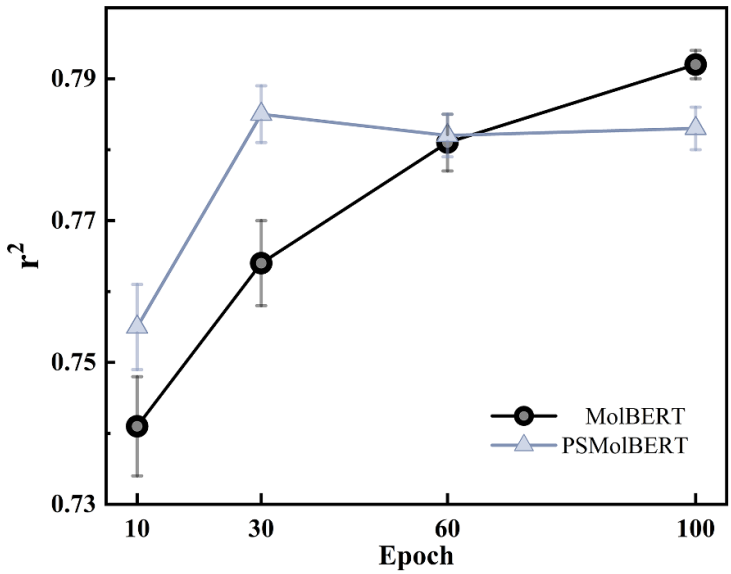

Figure 2. The QSAR performance of MolBERT and PSMolBERT as a function of pre-training epochs (each model were pretrained 3 times with different random seeds and error bars represents the standard deviation of the performance).

has been used as another indicator of the performance as well and is shown in supplementary table 5. MolBERT achieves the smallest average RMSE (0.467). Meanwhile, the RMSE for PSMolBERT (0.475) is slightly larger than the one for MolBERT, which indicates efficient model compression as well.

4.2.2. The effect of number of pre-training epochs on QSAR performance

According to the results of QSAR modeling, PSMolBERT achieves competitive performance and is 4.4× faster (table 2, 192/44 = 4.4) on pre-training compared with MolBERT. Besides less training time per epoch, we assume that a quicker convergence contributes to the pre-training acceleration. We will therefore examine the QSAR performance as a function of pre-training epochs in more detail. PSMolBERT and MolBERT pre-trained with 10, 30, 60 and 100 epochs were compared with each other. Figure 2 displays the results of regression task performance. It demonstrates that the performance variation trends differ between PSMolBERT and MolBERT. For MolBERT, the performance improves as the pre-training epoch increases. However, in the case of PSMolBERT, no significant improvement is achieved after 30-epoch training. In other words, the best performance with respect to computational cost is achieved when the model is trained with 30 epochs. The result strengthens our hypothesis in section 3.1 that PSMolBERT converges much faster than MolBERT, which leads to less pre-training time.

Summing up, by CLPS, a 4.4× faster training speed is achieved while retaining about 99% of the performance of the original model. Therefore, CLPS can be concluded to be a parameter-efficient method for molecular modeling, reducing the high hardware requirements for pre-training transformer models. In the next section, we will discuss the effect of KD on inference speed and modeling performance.

4.3. Effect of KD on model performance

4.3.1. Retained QSAR performance by KD

We concluded in section 3.1 that distillation resulted in a faster rate of inference due to less attention layers. We will now explore the effect of KD on the QSAR performance as well. Model distillation contains two lines of strategies, general- and task specific distillation [26]. General distillation is conducted by self-supervised learning to get general-domain knowledge, while task specific distillation compress models for specific tasks. In this work, only general distillation is studied on the chemical transformer models. By general distillation, the chemical knowledge is distilled from a teacher model to the student one using unlabeled data. The teacher model here was MolBERT, while the student model, KDMolBERT, was obtained after KD for 12.5 h. Additionally, one random initialized 3-layer MolBERT model was pre-trained using the same time (12.5 h), named MolBERT-3. It was used as a baseline for comparison. Table 5 shows the results of the QSAR performance of classification and regression tasks on average. The results demonstrate that: (a) 99.3% and 96.7% of the performances of MolBERT were achieved by KDMolBERT in classification and regression tasks, respectively; (b) Except for CDDD on regression tasks, KDMolBERT outperforms other traditional methods in QSAR prediction (the performance by the CDDD method is shown in tables 4 and 5); (c) Even though the same time was used, the performance of KDMolBERT is much better than for MolBERT-3. These results

Table 5. The QSAR performance (AUC-ROC and r^2 for classification and regression, respectively) of models by KD (KDMolBERT) and learning from scratch (MolBERT-3).

Method	KDMolBERT	Retained performance[a] (%)	MolBERT-3	Retained performance[b] (%)
Classification	0.890	99.3	0.821	91.6
Regression	0.764	96.7	0.714	90.4

[a] The performance ratio between the student (KDMolBERT) and teacher model (MolBERT, model with best performance compared with others)

[b] the performance ratio between 3-layer model learning from scratch and MolBERT.

Table 6. Ablation study and variations to the model trained with triple loss.

	Variation on QSAR	
Without	Classification	Regression
L_{logits}	−3.6	−4.1
L_{hidn}	−1.5	−1.7
L_{mlm}	−0.5	−0.4

suggest that the general-domain chemical knowledge can be effectively transferred from a teacher to student model in the KD process. Then, the compressed student model could be used for downstream tasks with 3.8× faster inference (shown in table 2).

4.3.2. Ablation study on distillation objectives

The loss function of the KD, also called distillation objective, includes the losses of maskedlm, hidden states and logits (see section 2). The influence of each component in the triple loss was investigated by an ablation study. Table 6 presents the performance of removing each learning procedure. Firstly, the performances without logits significantly decrease by 3.6% and 4.1% units for classification and regression, respectively. The reason for the significant decrease lies in the mechanism of the QSAR modeling, in which pooled output from chemical transformers is used for classification or regression [22]. Both logits and pooled output is derived from the sequence output. When considering logits loss, the distribution of sequence output from the student model is matched with the one from teacher model. In order to receive high performance in fine-tuning processes, the logit loss need to be taken into account. Then, the impact of hidden states loss on performance was tested. The decreases of performance are 1.5% and 1.7% units for classification and regression, respectively. Therefore, the effect was moderate, which is consistent with cases in NLP, and therefore not examined further. Lastly, the effect of MaskedLM loss was examined, showing only minor changes in QSAR performance. Previous work on MolBERT indicated that additive gain from the MaskedLM is relatively minor, compared with the one in NLP. Our results indicate that MaskedLM has the least effect on chemical KD as well. Hence, the result above implies that the influence of each part in the triple loss is in agreement with the one in NLP [25].

According to the abovementioned results, it can be concluded that KD is an empirically effective way to scale down the model size to facilitate inference and achieve competitive performance with its teacher chemical transformer. In the next section, the two compression methods, CLPS and KD, will be integrated with each other and the performance of the final model will be studied.

4.4. Effect of integration on QSAR performance

The discussion so far has revolved around the effect of compression by CLPS and KD independently. These two compression methods work at separate stages in model constructions. CLPS reduces the time for training and KD for inference. They can therefore be applied within the same model. By applying CLPS on the original model MolBERT, we received PSMolBERT. Then by applying KD on PSMolBERT, we integrate CLPS and KD into one single model. We call this deep light chemical transformer, DeLiCaTe, and will assess its performance in QSAR.

The analysis of the QSAR performance of DeLiCaTe was done in an analogue's manner as in section 3.3.1. As a baseline, a 3-layer PSMolBERT model (named PSMolBERT-3) was pre-trained from scratch to compare with DeLiCaTe. The pre-training time of the 3-layer PSMolBERT was the same as the distillation time of DeLiCaTe (6 h). Their QSAR performances are shown in table 7. DeLiCaTe achieves comparable performance to the teacher model, the 12-layer MolBERT. Specifically, it retains 97.2% and 94.7% of the performance of MolBERT for classification and regression tasks, respectively. It indicates that the combination of CLPS and KD not only compresses the model effectively, but also retains the chemical modeling ability. In comparison, the 3-layer PSMolBERT only achieve ∼91% of the performance of

Table 7. QSAR performance (AUC-ROC and r^2 for classification and regression, respectively) of DeLiCaTe and 3-layer PSMolBERT in comparison with MolBERT.

Method	DeLiCaTe	Retained performance (%)[a]	PSMolBERT-3	Retained performance (%)[b]
Classification	0.870	97.2	0.815	91.0
Regression	0.748	94.7	0.721	91.2

[a] The performance ratio between DeLiCaTe (obtained by KD from PSMolBERT) and MolBERT (MolBERT, model with best performance compared with others).
[b] the performance ratio between 3-layer PSMolBERT learning from scratch and MolBERT.

Table 8. AUC-ROC and standard deviation (SD) for virtual screening.

	RDKit	ECFP4	CDDD	MolBERT	PSMolBERT	KDMolBERT	DeLiCaTe
AUC-ROC	0.633	0.603	0.725	0.743	0.737	0.730	0.719
SD	0.027	0.056	0.057	0.062	0.059	0.066	0.064

MolBERT. It implies that a limited pre-training time and a small parameter scale (fewer number of layers) reduces the efficiency of molecular modeling by the learning from scratch method. Besides retaining performance, DeLiCaTe is ∼4× faster on both training and inference compared to MolBERT (table 2). Therefore, the integration of the two compression methods is an efficient strategy to counteract the time consuming and high hardware requirement of chemical transformers.

4.5. VS performance of compressed models

To further assess the efficiency of compression methods, the VS performance was studied with the aforementioned models. Given the compressed structure, the speed of VS was about 4× faster for DeLiCaTe than for MolBERT. Table 8 illustrated the average result of VS performance, and the performances of individual dataset are shown in supplementary figure 2. The results demonstrate that all the compressed models achieve comparable performance. For instance, DeLiCaTe retain 96.8% of the performance of the original MolBERT. Considering the mechanism of VS, the results suggest that DeLiCaTe can discriminate the structural similarity among different molecules to facilitate the VS process. To strengthen this hypothesis, we further calculated the average pairwise cosine similarity of the molecules in the ChEMBL dataset using DeLiCaTe, MolBERT and ECFP4. As shown in supplementary figure 3, the results of DeLiCaTe and MolBERT are very closed to each other and much lower than the one of ECFP4, which implies the excellent ability of molecular similarity discrimination by DeLiCaTe. Summing up, DeLiCaTe exhibits competitive performance with the original transformer and outperforms some of the baseline methods for VS tasks.

5. Conclusion

In this work, we demonstrate the effect of implementing the CLPS and KD methods individually, as well as the integration of these two methods to compress chemical transformers. Both a 4× speedup for training and inference were achieved by applying CLPS and KD, respectively. Furthermore, a deep light chemical transformer model, DeLiCaTe, was introduced to integrate the accelerating abilities of both compression strategies. According to the result of QSAR and VS performance, all the compressed transformers retain the molecular modeling capability of the original model and outperform or compete with state-of-the-art baseline methods. Consequently, time consuming and high hardware requirement are mitigated by the compressed parameter-efficient method. These results can facilitate the application of molecular discrimination based on chemical transformer encoders to a broader scientific community. Furthermore, due to the similarity between transformer encoders and decoders, this strategy is anticipated to promote the use of generative transformer models for organic drug and material design in the future.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/YiYuDL/DeLiCaTe.

Acknowledgments

We gratefully acknowledge financial support from the European Research council (ERC2017-StG-757733). Chao Fang is acknowledged for discussion on theory.

Conflict of interest

The authors declare no competing interests.

ORCID iDs

Yi Yu https://orcid.org/0000-0001-8360-005X
Karl Börjesson https://orcid.org/0000-0001-8533-201X

References

[1] Muratov E N *et al* 2021 A critical overview of computational approaches employed for COVID-19 drug discovery *Chem. Soc. Rev.* **50** 9121–51
[2] Singh N, Chaput L and Villoutreix B O 2020 Virtual screening web servers: designing chemical probes and drug candidates in the cyberspace *Brief. Bioinform.* **22** 1790–818
[3] Vamathevan J *et al* 2019 Applications of machine learning in drug discovery and development *Nat. Rev. Drug Discovery* **18** 463–77
[4] Pandey M, Fernandez M, Gentile F, Isayev O, Tropsha A, Stern A C and Cherkasov A 2022 The transformational role of GPU computing and deep learning in drug discovery *Nat. Mach. Intell.* **4** 211–21
[5] Sanchez-Lengeling B and Aspuru-Guzik A 2018 Inverse molecular design using machine learning: generative models for matter engineering *Science* **361** 360–5
[6] Schneider P *et al* 2020 Rethinking drug design in the artificial intelligence era *Nat. Rev. Drug Discovery* **19** 353–64
[7] Krenn M, Häse F, Nigam A, Friederich P and Aspuru-Guzik A 2020 Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation *Mach. Learn.: Sci. Technol.* **1** 045024
[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] James C A 2010 *OpenSMILES specification* (available at: http://opensmiles.org/opensmiles.html)
[10] Kearnes S, McCloskey K, Berndl M, Pande V and Riley P 2016 Molecular graph convolutions: moving beyond fingerprints *J. Comput. Aided Mol. Des.* **30** 595–608
[11] Sandfort F, Strieth-Kalthoff F, Kühnemund M, Beecks C and Glorius F 2020 A structure-based platform for predicting chemical reactivity *Chem* **6** 1379–90
[12] Brown N, Fiscato M, Segler M H S and Vaucher A C 2019 GuacaMol: benchmarking models for de novo molecular design *J. Chem. Inf. Model.* **59** 1096–108
[13] Segler M H S, Kogej T, Tyrchan C and Waller M P 2018 Generating focused molecule libraries for drug discovery with recurrent neural networks *ACS Cent. Sci.* **4** 120–31
[14] Wang M, Sun H, Wang J, Pang J, Chai X, Xu L, Li H, Cao D and Hou T 2021 Comprehensive assessment of deep generative architectures for de novo drug design *Brief. Bioinform.* **23** bbab544
[15] Vaswani A, Shazeer N and Parmar N 2017 Attention is all you need (arXiv:1706.03762)
[16] Irwin R, Dimitriadis S, He J and Bjerrum E J 2022 Chemformer: a pre-trained transformer for computational chemistry *Mach. Learn.: Sci. Technol.* **3** 015022
[17] Zhang X-C, Wu C-K, Yang Z-J, Wu Z-X, Yi J-C, Hsieh C-Y, Hou T-J and Cao D-S 2021 MG-BERT: leveraging unsupervised atomic representation learning for molecular property prediction *Brief. Bioinform.* **22** bbab152
[18] 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
[19] Ross J, Belgodere B, Chenthamarakshan V, Padhi I, Mroueh Y and Das P 2021 Do large scale molecular language representations capture important structural information? (arXiv:2106.09553)
[20] Maziarka Ł, Pocha A, Kaczmarczyk J, Rataj K, Danel T and Warchoł M *2020* Molecule attention transformer (arXiv:2002.08264)
[21] Chithrananda S, Grand G and Ramsundar B 2020 Chemberta: large-scale self-supervised pretraining for molecular property prediction (arXiv:2010.09885)
[22] Fabian B, Edlich T, Gaspar H, Segler M, Meyers J, Fiscato M and Ahmed M 2020 Molecular representation learning with language models and domain-relevant auxiliary tasks (arXiv:2011.13230)
[23] He J, You H, Sandström E, Nittinger E, Bjerrum E J, Tyrchan C, Czechtizky W and Engkvist O 2021 Molecular optimization by capturing chemist's intuition using deep neural networks *J. Cheminformatics* **13** 26
[24] Xue D *et al* 2022 X-MOL: large-scale pre-training for molecular understanding and diverse molecular analysis *Sci. Bull.* **67** 899–902
[25] Sanh V, Debut L, Chaumond J and Wolf T 2019 DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter (arXiv:1910.01108)
[26] Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, Wang F and Liu Q 2019 Tinybert: distilling bert for natural language understanding (arXiv:1909.10351)
[27] Dehghani M, Gouws S, Vinyals O, Uszkoreit J and Kaiser Ł 2018 Universal transformers (arXiv:1807.03819)
[28] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P and Soricut R 2019 Albert: a lite bert for self-supervised learning of language representations (arXiv:1909.11942)
[29] Jia D, Han K, Wang Y, Tang Y, Guo J, Zhang C and Tao D 2021 Efficient vision transformers via fine-grained manifold distillation (arXiv:2107.01378)
[30] Goel A, Tung C, Lu Y H and Thiruvathukal G K 2020 A survey of methods for low-power deep learning and computer vision (arXiv:2003.11066)
[31] Tang R, Lu Y, Liu L, Mou L, Vechtomova O and Lin J 2019 Distilling task-specific knowledge from bert into simple neural networks (arXiv:1903.12136)
[32] Sun S, Cheng Y, Gan Z and Liu J 2019 Patient knowledge distillation for bert model compression (arXiv:1908.09355)
[33] Wang W, Wei F, Dong L, Bao H, Yang N and Zhou M 2020 Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers (arXiv:2002.10957)
[34] Sun Z, Yu H, Song X, Liu R, Yang Y and Zhou D 2020 Mobilebert: a compact task-agnostic bert for resource-limited devices (arXiv:2004.02984)

[35] Hinton G, Vinyals O and Dean J 2015 Distilling the knowledge in a neural network (arXiv:1503.02531)
[36] Gaulton A *et al* 2016 The ChEMBL database in 2017 *Nucleic Acids Res.* **45** D945–D954
[37] Landrum G A 2020 *RDKit: Open-Source Cheminformatics Software* (available at: www.rdkit.org)
[38] Winter R, Montanari F, Noé F and Clevert D-A 2019 Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations *Chem. Sci.* **10** 1692–701
[39] Wu Z, Ramsundar B, Feinberg E, Gomes J, Geniesse C, Pappu A S, Leswing K and Pande V 2018 MoleculeNet: a benchmark for molecular machine learning *Chem. Sci.* **9** 513–30
[40] Hansen K, Mika S, Schroeter T, Sutter A, ter Laak A, Steger-Hartmann T, Heinrich N and Müller K-R 2009 Benchmark Data Set for in Silico Prediction of Ames Mutagenicity *J. Chem. Inf. Model.* **49** 2077–81
[41] Rybacka A, Rudén C, Tetko I V and Andersson P L 2015 Identifying potential endocrine disruptors among industrial chemicals and their metabolites—development and evaluation of in silico tools *Chemosphere* **139** 372–8
[42] Bento A P *et al* 2013 The ChEMBL bioactivity database: an update *Nucleic Acids Res.* **42** D1083–90
[43] Li Y, Xu Y and Yu Y 2021 CRNNTL: convolutional recurrent neural network and transfer learning for QSAR modeling in organic drug and material discovery *Molecules* **26** 7257
[44] Riniker S, Fechner N and Landrum G A 2013 Heterogeneous classifier fusion for ligand-based virtual screening: or, how decision making by committee can be a good thing *J. Chem. Inf. Model.* **53** 2829–36
[45] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library (arXiv:1912.01703)
[46] Wolf T *et al* 2019 Huggingface's transformers: state-of-the-art natural language processing (arXiv:1910.03771)
[47] Yang Z, Cui Y, Chen Z, Che W, Liu T, Wang S and Hu G 2020 Textbrewer: an open-source knowledge distillation toolkit for natural language processing (arXiv:2002.12620)
[48] Karpov P, Godin G and Tetko I V 2020 Transformer-CNN: swiss knife for QSAR modeling and interpretation *J. Cheminformatics* **12** 17