JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

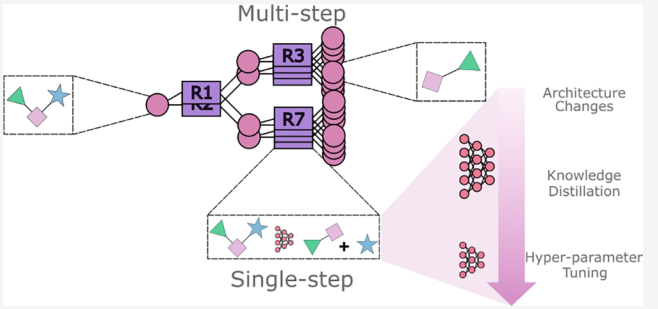
pubs.acs.org/jcim Article

Investigations into the Efficiency of Computer-Aided Synthesis Planning

Peter B.R. Hartog,* Annie M. Westerlund, Igor V. Tetko, and Samuel Genheden*

Cite This: J. Chem. Inf. Model. 2025, 65, 1771−1781 Read Online

ABSTRACT: The efficiency of machine learning (ML) models is crucial to minimize inference times and reduce the carbon footprints of models deployed in production environments. Current models employed in retrosynthesis to generate a synthesis route from a target molecule to purchasable compounds are prohibitively slow. The model operates in a single-step fashion in a tree search algorithm by predicting reactant molecules given a product molecule as input. In this study, we investigate the ability of alternative transformer architectures, knowledge distillation (KD), and simple hyper-parameter optimization to decrease inference times of the Chemformer model. Initially, we assess the ability of closely related transformer architectures and conclude that these models under-performed when using KD. Additionally, we investigate the effects of feature-based and response-based KD together with hyper-parameters optimized based on inference sample time and model accuracy. We find that although reducing model size and improving single-step speed are important, our results indicate that multi-step search efficiency is more significantly influenced by the diversity and confidence of single-step models. Based on this work, further research should use KD in combination with other techniques, as multi-step speed continues to prevent proper integration of synthesis planning. However, in Monte Carlo-based (MC) multi-step retrosynthesis, other factors play a crucial role in balancing exploration and exploitation during the search process, often outweighing the direct impact of single-step model speed and carbon footprints.



INTRODUCTION

The efficiency of machine learning (ML) models is crucial to minimize inference times and reduce the carbon footprints of models.[1] This is especially pronounced for ML deployed in production environments, where model queries are more frequent. ML models have become standard in most areas of drug design, including computer-aided synthesis planning tools. Retrosynthesis analysis predicts the synthesis route from a target molecule to a purchasable material. This route is generated using a single-step ML model that predicts reactant molecules for a product molecule. The single-step model is then employed in a multi-step tree search to generate full synthesis routes.[2] Such algorithms typically only produce the reactants needed for the synthesis and need to be combined with other algorithms to predict reagents such as solvents and catalysts, as well as other reaction conditions.[3] Optimizing single-step retrosynthesis prediction becomes especially relevant as retrosynthesis tools are routinely used by chemists and the computational complexity of the multi-step search is heavily dependent on single-step performance.[4]

Single-step retrosynthesis models are either template-based methods, template-free methods, or a combination of both.[5] Transformer encoder-decoder architectures[6] have been used in retrosynthesis research[7−10] for their ability to scale to large data set and the promise of extrapolation to novel chemistry. However, no specific architecture has been recognized to outperform, as depending on training data and architectures cover different aspects of the chemical field.[11] Additionally, although single-step accuracy and multi-step success rates are high with template-free methods, inference and search times are too slow to be readily used in production.[12] Furthermore, recent work by Torren-Peraire et al.,[13] Maziarz et al.[14] highlights the difficulty of translating single-step performance to multi-step retrosynthesis success.

Knowledge distillation (KD)[15] exploits previously trained teacher models to train smaller and more efficient student models. The information from a teacher model can be transferred to a student model in three ways: (1) response-based KD,[15] (2) feature-based KD and (3) relation-based KD.[16] These approaches distill knowledge from the output, internal representations, or the angles/distances in between batches of internal representations, respectively (Figure 1).

Received: October 4, 2024
Revised: December 13, 2024
Accepted: December 18, 2024
Published: January 31, 2025

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

Additionally, more efficient Transformer architectures have been pursued to increase the inference speed of encoder-decoder architectures. Efficient alternative Transformer architectures are a large area of research[17] and include models with induced sparsity for faster compute times,[18−20] and memory-efficient transformers with lower complexity.[21−24] Previous work specific to retrosynthesis also included increasing the efficiency of the sampling algorithms using speculative decoding.[25]

Here, we build on previous work on the AiZynthFinder tool[12,26] that has been used successfully in drug discovery projects.[4] We address the efficiency issue of a retrosynthesis Transformer model by employing alternative architectures, KD, and hyper-parameter optimization. We then analyze the impact these optimizations have on the fine-tuned models in terms of accuracy, speed, and carbon footprint. Importantly, we investigate whether single-step speed-up translates into multi-step retrosynthesis search times. This work thus demonstrates the promise of scaling up a retrosynthesis transformer model to reduce carbon footprints while retaining prediction accuracy and multi-step success rates.

METHODS

Data Collection and Processing. We adapted the Chemformer model[10] into different KD variants. The baseline Chemformer was originally pretrained on a masked auto-translation task on approximately 100 million randomly selected SMILES strings of molecules from the around 1.5 billion molecules in the ZINC-15 data set.[27] It was then fine-tuned on the USPTO-50k for the retrosynthesis task.[10] We collected the corresponding data set, original model weights, and parameters from Irwin et al.[10] The USPTO-50k data set contains around 50,000 reactions and was here used to benchmark the model variants. Each Chemformer variant was trained on UPSTO-full, gathered from Genheden et al.[28] Both USPTO-50k and UPSTO-full originated from the USPTO data set from Lowe.[29] To evaluate the multi-step search, we used 5000 ChEMBL structures from Genheden et al.[28] The stock was taken as the building block set of 24.7 million molecules from eMolecules.[30]

The original Chemformer vocabulary was constructed by applying regular expression matching, similar to Schwaller et al.,[9] to the canonical SMILES[31] of the molecules in the ChEMBL 27.[32] The total 523 tokens include 250 chemical tokens, 200 tokens that are unused during the pretraining stage but filled during fine-tuning, and 73 token meta-tokens, such as masking, padding, beginning-of-sentence, and end-of-sentence tokens, or tokens for the originally implemented molecular optimization task. Tokenization and augmentation of SMILES was performed by the ChemformerTokenizer which extends the routines in PySMILESUtils.[33]

Model Architectures. We experimented with three different model architectures in Chemformer, so-called model variants, including the original Chemformer architecture,[10] the Perceiver architecture,[23] and the Switch Transoformer.[20] The original Chemformer architecture is a BART (Bidirectional and Auto-Regressive Transformer) model with encoder-decoder architecture,[6,34,35] using sinusoidal positional embedding, prenorm layer-normalization, and GELU activation functions. A Perceiver variation was used,[23] keeping the original Chemformer architecture with the exception of an initial cross-attention layer in the Transformer encoder. This cross attention is combined with a learnable projection tensor, which was used for the query, and the embedded source sequence was used as the key-value pair, projecting the internal representation to a smaller dimension. Finally, to experiment with induced sparsity, we used the Switch Transformer[20] layer implementation of Varuna Jayasiri.[36] In the Switch transformer, the traditional feed-forward was replaced with eight experts with feed-forward layers of one-eight times the size of the original model.

Model Training and Hyper-Parameter Optimization. Model training was performed using PyTorch (version 2.2.2)[37] and Lightning (version 2.2.1)[38] on a single GPU. Most of the original training parameters from the Chemformer paper[10] were used to allow for direct comparisons (Table 1). Using Optuna,[39] hyper-parameter optimization was carried out using the Bayesian Tree-structured Parzen estimator (TPE).[40,41] Inspired by earlier work on hyper-parameter optimization for multi-step retrosynthesis,[42] we set the objectives for the optimization to the validation greedy search accuracy, and an adapted version of the rate-correct score (RCS).[43] It was obtained by dividing the correct samples by the cumulative time ($\frac{\text{Correct Samples}}{\text{Total Time}}$). The choices of the model parameters are given in Table 1.

Table 1. Training Hyper-parameters for the Baseline Implementations[a]

	Training		
Parameter	UPSTO-50k	UPSTO-full	Optimization
Batch size	128	128	128
Epochs	500	50	100
Scheduler	cycle	cycle	cycle
Learning rate	10^{-3}	10^{-3}	$(10^{-2},10^{-3},10^{-4})$
Model Dim	512	512	(64,128,256,512)
Feed-forward	2048	2048	(64,128,256,1024,2048)
Encoder Layer	6	6	(1,2,3,6)
Decoder Layer	6	6	(1,2,3,6)
Attention Head	8	8	8
Switch Experts	8	8	N/A
Perceiver Dim	32	32	N/A
Max seq length	512	512	512
Vocab Size	523	523	523
Dropout	0.1	0.1	0.1
Optimizer	Adam	Adam	Adam

[a]Multiple values represent choices for hyper-parameter optimization. Feed forward dimension was divided by the number of experts in Switch Transformer architectures.

Distillation was performed using implementations from Textbrewer (version 0.2.1 - post1),[44] which was adapted to Chemformer. Cross entropy losses were computed on the output values (response-based KD) or internal representations using direct comparison (feature-based representations) with or without standard cross-entropy loss, also named hard-label loss. All losses were combined using a weighted average based on weight hyper-parameters, normally set to 1.0, which were subsequently normalized to sum to one. Feature-based representations were compared between the output of the embedding, encoder, or decoder modules.

Carbon footprints were approximated using eco2AI (version 0.3.9).[45] CO_2 emissions were approximated by multiplying the energy consumption by the average emissions coefficient of a country, specifically Sweden.

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

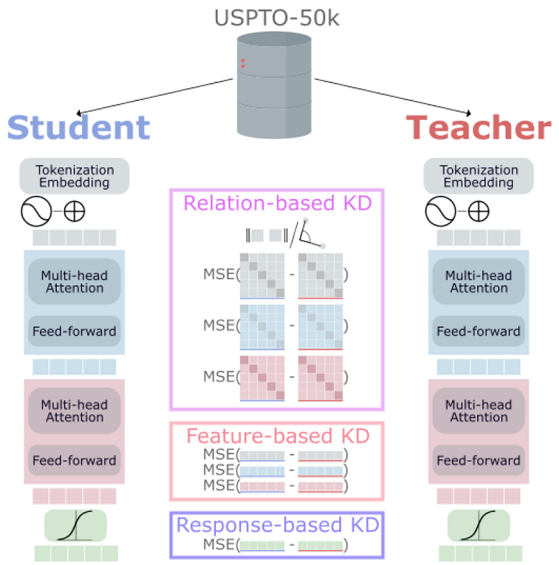


Figure 1. Illustrative example of different types of knowledge distillation. The information from a teacher model can be transferred to a student model in three ways. Response-based KD learns from the output of the teacher model. Feature-based KD, which learns from the internal representations. And relationship-based KD, which learns from the angles/distances between samples of the same models and correlates them between models.

Multi-step Search. The multi-step search was performed using AiZynthfinder (version 3.6.0).[2,26] To compare the performance of Chemformer to the current in-production model, we used the template model from Genheden et al.[28] The multi-step expansion policy was set to either template-based, Chemformer-based (including all models developed here) or multiexpansion. In the multiexpansion policy, the template-based was combined with the original Chemformer model trained on USPTO-50k. The maximum search time per search was set to 500 s, maximum iterations to 100, and the maximum depth to 6.

We evaluate the performance of the multi-step search in terms of search time (elapsed wall-clock time) and success rate. The success rate is defined as the percentage of the ChEMBL targets for which the search finds at least one synthesis route leading to starting material in the e-Molecules stock (also referred to as solvability). Although not a measure of route quality, it is a necessary condition to find the solved synthesis routes. The measure of route quality is a debated topic,[14,46] and herein we have chosen not to evaluate it because our primary objective is to compare the effect of different single-step models on multi-step search. To compare the routes produced by the different models, we compute the route similarity between the highest-ranked routes for each target.[47] The routes were ranked by the AiZynthFinder reward function, which takes into account the fraction of starting material in stock and the length of the route.[48]

RESULTS

Investigating Alternative Model Architectures for Retrosynthesis Prediction. In order to assess the general effects of using alternative architectures, we investigated three different architectures, the baseline BART transformer model, the mixture-of-experts Switch Transformer, and the linear scaling Perceiver model. Additionally, we studied the effects of halving the number of encoder and decoder layers and using KD on the various architectures. To isolate the effects of architecture and KD, all other hyper-parameters were kept identical to the baseline.

First, we look at the changes originating from the architecture changes alone. The Switch transformer architecture initially yielded similar effects as training from scratch with the baseline Transformer (Table 2), generating similar top-1 and top-10 accuracy. However, the Switch Transformer is more than twice as slow, 39.42 s over 17.20 s per batch, and has a significant increase in the carbon footprint of training similarly sized Transformers from scratch. The Perceiver architecture similarly yields a reduction in accuracy, 47.2% over 51.1% top-1 accuracy, but has the fastest inference times at 14.10 s per batch.

Next, we investigated the effects of halving the number of model weights from around 45 to around 23 M through the reduction of encoder and decoder layers. This architectural

Table 2. Effects of Alternative Model Architectures On Single-step Retrosynthesis Prediction[a]

model	size	sample time (s)	top 1 accuracy	top 10 accuracy	percentage invalid	percentage unique	CO_2 emissions (g)
			Full Size				
Chemformer	44.7 M	17.72	53.5	61.6	0.68	18.06	0.62
Transformer	44.7 M	17.20	51.1	64.2	0.89	22.03	0.48
Switch	47.9 M	39.42	50.9	64.3	1.04	24.54	0.78
Perceiver	45.7 M	14.10	47.2	57.5	1.02	22.92	0.66
			Half Size				
Transformer	22.6 M	9.25	50.8	70.8	1.34	31.44	0.28
Switch	24.2 M	16.00	49.6	70.3	1.64	35.29	0.38
Perceiver	23.7 M	7.84	42.8	57.2	1.91	31.92	0.25
			Knowledge Distillation				
Transformer	22.6 M	11.57	50.6	68.3	4.40	78.34	0.51
Switch	24.2 M	18.76	32.6	62.9	5.54	81.04	0.99
Perceiver	23.7 M	8.49	29.0	58.4	6.67	85.24	0.26

[a]All models are trained from scratch with different architectures, including an encoder-decoder transformer, an implementation of the Switch transformer with mixture-of-expert layers, and an implementation of the Perceiver model. CO_2 emissions are estimates based on kWh consumption, GPU type, and location of the compute cluster.

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

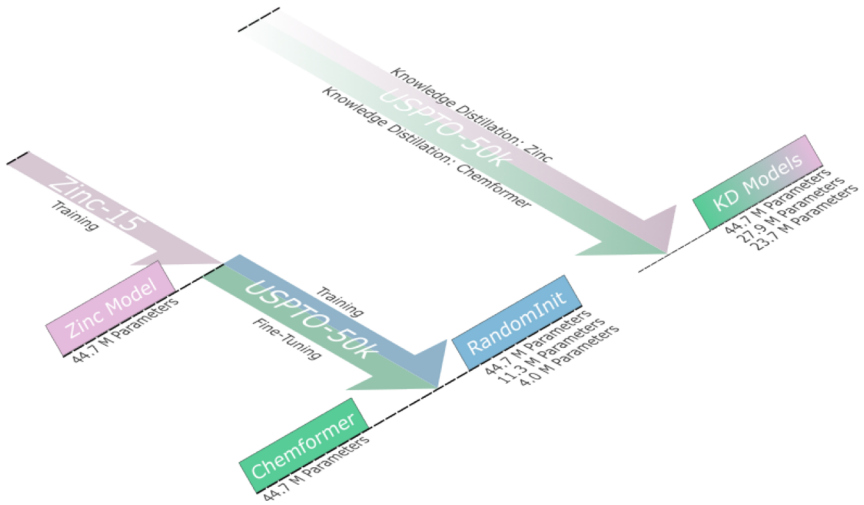


Figure 2. Illustrative example of model training types. Visualization of different types of model training employed, including standard training, fine-tuning, and knowledge distillation.

change yielded a small decrease in top-1 paired with an increase in top-10 accuracy for both the baseline Chemformer as well as the Switch Transformer (Table 2). Halving the amount of encoder and decoder layers resulted in slightly halved inference times. However, the Perceiver model experienced a marked decrease in top-1 accuracy when halving the model size compared to the BART and Switch models. Top-10 accuracy in the Perceiver also remained similar to the large-sized variant, while top-10 accuracy actually increased substantially in the other two architectures upon decreasing the model size. Carbon footprints also decreased in all half-size model variants, although the effect was less than the reductions in sample times.

Finally, we examined whether training the architecture variants with KD could reduce the inference time while preserving high accuracy. Response-based KD was used for all three variants, whereas feature-based KD on the Encoder and Decoder was used on the baseline and Switch transformer but was not used in the Perceiver model due to the changes of internal sequence length which are inherent to the model. The results of KD on the original Chemformer indicated that most of the reduction in top-1 accuracy from 53.5 to 50.6% could be explained by the significant increase of the invalid and unique molecules during inference (Table 2). When comparing the original Chemformer to the KD Transformer, we observed a reduction in sample time and carbon footprint, paired with slight reductions in model accuracy (50.6% vs 53.5%). The Switch transformer, however, displayed detrimental results when trained with KD, reaching at most 32.6% top-1 accuracy. The Perceiver KD had the worst results overall, with a top-1 accuracy of 29%, while the speed-up was not substantial compared to the BART Chemformer (8.49 s vs 11.57 s). Notably, this difference is likely due to the lack of a feature-based KD.

Overall, reducing the number of encoder and decoder layers decreases batch inference times and carbon footprints but impacts model accuracy when evaluated on similar training times. Additionally, changes in architecture are not directly compatible with the KD of the original architecture, as indicated by the low top-1 and top-10 accuracy scores when using KD on alternative architectures. In conclusion, for single-step prediction, decreasing the size of the BART Chemformer variant appears to be more efficient (in accuracy, sample time, and fraction of invalid) than employing knowledge distillation on any architecture variant.

Hyper-parameter Optimization Using Knowledge Distillation To Increase Inference Speed. To assess the impact that hyper-parameters have on single-step retrosynthesis for speed, Bayesian optimization was used to optimize the types and proportions of KD during training, model dimension, feed-forward dimension, number of attention heads, and number of encoder and decoder layers (Figure 3). We used the optimization objectives of validation accuracy and the validation rate-correct score (RCS) on one of three training variations: training from scratch (randomly initializing the weights with Xavier uniform distribution), response-based, and feature-based KD using the original Chemformer (KD: Chemformer) or the pretrained model trained on Zinc (KD: Zinc). The Pareto front indicates the highest validation accuracy in combination with the fastest average inference time, shown in red.

First, we observe the overall model quality generated by the hyper-parameter optimization. By optimization on either high validation accuracy or RCS, models were obtained that were both fast and accurate. However, the Pareto front shows that no model achieves a top-1 accuracy above or near the original Chemformer model of 53.5%. The highest validation accuracy was achieved using the optimal parameters from randomly initialized models, 46.7% (Table A1, RandomInit). Overall, most configurations were dominated by KD using the pretrained Zinc model (Figure 3), with the highest accuracy models on the Pareto front coming from the randomly initialized and KD Chemformer models.

Finally, specific hyper-parameters that contributed to the highest validation accuracy and RCS models are identified (Table A2). All models had the maximum number of six encoder layers and fewer than six decoder layers, with the exception of the slowest, best-performing randomly initialized model. Furthermore, all chosen models, with the exception of the randomly initialized models, selected the largest possible feed-forward and embedding dimensions of 2048 and 512, respectively. With respect to KD, models distilling from the Zinc models selected high values for all possible KD values, whereas models distilling from Chemformer prioritized

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

embedding and decoder feature-based KD over encoder feature-based KD and response-KD, with all preferring training knowledge distillation in combination with standard training. In conclusion, hyper-parameter optimization, in general, showed that on average KD based on the pretrained zinc model is closest to the Pareto front; however, all training methods have models on the Pareto front.

Effects of Hyper-Parameters on Fully Trained Models. Here, we train a variety of models based on the Transformer architecture to analyze the effects of hyper-parameter tuning on both the original data set and transferring these to the larger data set (Figure 2). Table 3 shows a full breakdown of beam search accuracies including and excluding valid and unique SMILES.

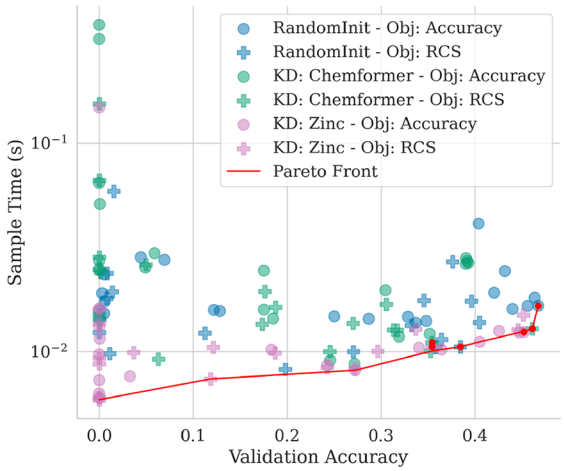


Figure 3. Results of hyper-parameter optimization with two optimization objectives and three training variations. Greedy search validation accuracy of hyper-parameter optimization with Pareto front of most accurate and fastest models. Three training variations include training from scratch, KD using the original Chemformer, and KD using the pretrained model trained on Zinc. Red points represent models chosen for further analysis for each variation of the highest rate correct score and highest accuracy.

When analyzing the randomly initialized transformer models trained on USPTO-50k, we compared each model with baseline parameters to the model with optimal hyper-parameters in terms of accuracy and RCS (Figure 4). When optimizing for accuracy, the resulting model had around 25% of the original size with only a 1.4% decrease in top-1 accuracy, coupled with a 5% increase in top-10 accuracy (Table 3). Optimizing on the RCS instead resulted in a model 10% of the original size with a 5% drop in the top-1 accuracy and a 16% increase in the top-10 accuracy. Notable, even though the sampling time dropped from 9.67 to 6.52 s between optimizing on accuracy instead of RCS, the carbon footprint was slightly higher for the RCS-optimized model (0.40 g instead of 0.34 g). However, both were substantially lower than the original implementation, indicating that carbon footprints decreased in these optimized models.

Subsequently, we analyze the difference between the original model and the newly retrained version to verify that changes in implementation did not result in significant changes in baselines. The results are shown in Table 3. First, we observed no significant difference between accuracy scores and only slight deviations in invalid and unique values. However, a significant difference was observed in the carbon footprints, which can be mostly attributed to the loading of model weights of the original model over inference after training. Additionally, models were also trained using the lowest validation loss, which showed lower top-1 accuracy but increased top-10 accuracy in all randomly initialized and fine-tuned models (Table A5). The highest change was the Chemformer (FT-Zinc) which went from 53.6 top-1 and 61.7 top-10 accuracy to 48.3 top-1 and 79.7 top-10 accuracy, for the last and optimal validation epoch, respectively. In this setting, the optimized models showed minimal differences or even improvements with respect to the randomly initialized smaller models, all reaching above 79% top-10 accuracy.

Furthermore, we compared the effects of KD on the full model predictions using the original Chemformer and the Zinc model with the original. Using KD, all models are significantly faster with respect to the randomly initialized models and Zinc fine-tuned models. However, this is paired with a decrease in the top-1 and top-10 accuracy scores. Furthermore, significant increases in the percentage of invalid smiles generated are observed in the KD models. Specifically, the best-performing Chemformer-based KD with optimal hyper-parameters for accuracy has a top-1 accuracy of 48.5% compared to 53.5% of the original Chemformer. A major explanation for this is the high percentage of invalid predictions, which range from 2.78% up to 23.32% for the KD models. These models are on average half the size of the original implementation but are as fast or faster than the randomly initialized model optimized for RCS, which is only 10% of the original model.

When looking at the ability of transferring optimal hyper-parameters across related training data, we observe similar effects found in the USPTO-50k models but more pronounced. The randomly initialized baseline model has a 6% higher top-1 and 4% higher top-10 accuracy over the smaller randomly initialized model with hyper-parameters optimized on accuracy from the UPSTO-50k data set (Table 3). The decrease in sample time and carbon footprint was also less pronounced between the accuracy model and the baseline model. The decrease in accuracy was larger in the randomly initialized model with parameters optimized for RCS, where accuracy dropped to 29.8 and 57.8%, for top-1 and top-10 respectively. This constitutes a 16.7 and 14.9% drop in accuracy from the baseline model for top-1 and top-10 accuracy, respectively.

Overall, all optimized models are faster but less accurate than the baseline and original implementations. Furthermore, the original implementations outperform all other models in top-1 accuracy but have lower top-10 accuracy and percentage of unique predictions. Additionally, the original implementations have carbon footprints higher than those of the newly trained models. Finally, we additionally investigated the influence of the invalid percentage and the unique percentage on top-n accuracies and observed minimal differences between top-n accuracy with or without valid and uniquely valid predictions (Table A3).

Translating Single-Step into Multistep. In order to assess whether single-step speed-ups translate to a multistep retrosynthesis search, a full multistep retrosynthesis search on 5000 ChEMBL molecules was analyzed for each of the single-step models. We compared model performances with a small template-based model which is currently used as the default in the AiZynthFinder framework. We focus our analysis on the speed, carbon footprint, and percentage of targets for which a route to starting material in stock is found (percentage solved).

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

Table 3. Single-step Model Full Training Model Statistics Based on the Transformer[a]

training	size	sample time (s)	top 1 accuracy	top 10 accuracy	percentage invalid	percentage unique	CO_2 emissions (g)
USPTO-50k				Original			
RandomInit	44.7 M	17.97	50.6	63.1	0.93	21.90	0.70
FT-Zinc	44.7 M	17.45	53.5	61.6	0.68	18.06	2.01
				Baselines			
RandomInit	44.7 M	16.20	51.3	63.7	0.87	21.75	0.76
FT-Zinc	44.7 M	16.73	53.6	61.7	0.82	18.61	0.61
				Optimized: Randomly Initiated			
Opt: Acc	11.3 M	9.67	50.9	70.2	1.14	30.96	0.34
Opt: RCS	4.0 M	6.52	45.8	79.5	2.52	62.04	0.40
				Optimized: KD Chemformer			
Opt: Acc	27.9 M	6.94	48.5	60.4	2.78	29.71	0.33
Opt: RCS	23.7 M	4.58	43.3	55.7	23.32	57.27	0.29
				Optimized: KD Zinc			
Opt: Acc	27.9 M	6.64	19.0	53.5	7.92	60.56	0.33
Opt: RCS	23.7 M	4.30	19.5	49.1	21.03	71.73	0.24
USPTO-full				Baselines			
RandomInit	44.7 M	29.69	46.5	72.7	1.10	66.46	12.30
FT-Zinc	44.7 M	24.22	47.9	72.6	0.98	60.53	6.42
				Optimized: Randomly Initiated			
Opt: Acc	11.3 M	21.85	40.2	68.6	1.23	73.21	9.40
Opt: RCS	4.0 M	7.55	29.8	57.8	1.64	75.59	2.25

[a]Models are trained either from scratch (RandomInit), fine-tuned from training on Zinc (FT-Zinc), using KD on the Chemformer model (KD-Chem) or KD on the zinc model (KD-Zinc). Hyper-parameters are either from the original publication (original), retrained models (baseline), or using hyper-parameters gathered from an Optuna search (opt: Acc/RCS) where specific parameters depend on the training set. Models are then trained and evaluated on USPTO-50k and USPTO-full. Reported statistics are based on beam-search results on random split test set values.

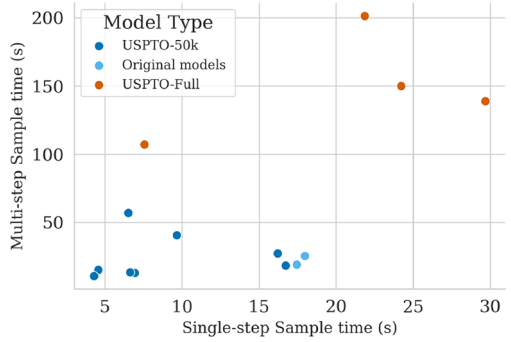


Figure 4. Median multi-step sample times set against single-step sample time. Median sample times of 5000 ChEMBL molecules set out against single-step average beam search sample times. Models include those trained on USPTO-50k, including the original models highlighted in light blue and models trained for USPTO-full.

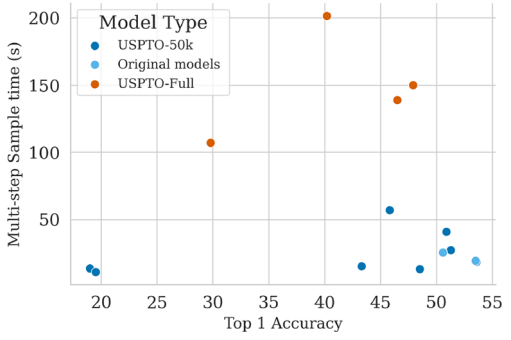


Figure 5. Median multi-step sample times set against top-1 single-step accuracy. Median search times of 5000 ChEMBL molecules set out against the accuracy of single-step models. Models include those trained on USPTO-50k, including the original models highlighted in light blue, and models trained for USPTO-full.

Carbon footprints are noted in grams and are mostly in agreement with search times.

First, we analyzed the differences between the original implementations and newly trained baselines (Table 4). When comparing the original implementations of the randomly initialized model and the Chemformer model and fine-tuning the Zinc model (FT-Zinc), we observed only small differences (Table 4). Most notable was the difference in the percentage solved between the original and the retrained baseline of the Chemformer model which decreased from 54.6 to 56.2% and the small decreases in carbon footprints from the original implementations, 196.34 and 123.46 g, to the baseline implementations, 168.15 and 110.17 g, for the randomly initialized and Chemformer models, respectively.

When investigating model behaviors on a single- and multistep basis, we could draw five key conclusions. First, an increased training data size, as exemplified by going from USPTO-50k to USPTO-full, showed significant increases in percentage solved but also markedly increased both search time and carbon footprints, with similar model calls and policy probabilities. Second, single-step top-1 is not predictive of the number of model calls (Figure 5). In contrast, multistep search times are more correlated with the average number of model calls per search. The number of model calls reflects how often the single-step model was queried over the retrosynthesis search. Third, the top-10 accuracy and policy probability are the major predictors of the average number of model calls, with lower policy probability correlating with a higher number of

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

Table 4. Multi-Step Retrosynthesis Search[a]

	median	average over search			average per molecule		
	search	search	CO_2	percentage	# solved	policy	# model
training	time (s)	time (h)	emissions (g)	solved (%)	routes	probability	calls
USPTO-50k			Including template-based model				
template-based	26.01	40	0.25	67.3	20.74	0.17	446
multiexpansion	382.08	495	2482.29	89.0	159.17	0.19	1832
			Original				
RandomInit	25.48	62	196.34	61.5	132.92	0.48	1384
FT-Zinc	19.16	47	123.46	56.2	125.21	0.53	1259
			Baselines				
RandomInit	27.29	69	168.15	60.4	134.19	0.48	1391
FT-Zinc	18.54	46	110.17	54.6	120.21	0.54	1277
			Optimized: Randomly Initiated				
Opt: Acc	40.96	94	237.94	69.7	136.59	0.39	1641
Opt: RCS	57.71	93	258.59	84.8	134.16	0.22	1733
			Optimized: KD Chemformer				
Opt: Acc	13.19	35	106.63	60.3	110.48	0.40	1053
Opt: RCS	15.37	34	97.60	62.4	65.99	0.28	773
			Optimized: KD Zinc				
Opt: Acc	13.59	29	101.10	63.9	37.77	0.15	442
Opt: RCS	10.82	25	61.89	63.5	35.50	0.16	365
USPTO-Full			Baselines				
RandomInit	138.87	259	967.68	87.5	119.53	0.22	1493
FT-Zinc	149.99	271	1340.05	86.8	116.00	0.22	1472
			Optimized: Randomly Initiated				
opt: acc	201.37	325	975.75	88.4	116.86	0.20	1500
Opt: RCS	107.15	180	552.17	91.1	122.95	0.17	1701

[a]Translating effects of single-step models to multi-step search. The models used here are the standard template-based model, a multi-expansion model combining template-based, and the original Chemformer. Furthermore, it includes the original implementations of the Chemformer paper, randomly initialized and fine-tuned from the pretrained Zinc model (RandomInit, FT-Zinc) and the retrained baselines. Additionally, optimized versions used either accuracy or rate correct score (OptAcc, OptRCS) of the randomly initialized (RandomInit), KD using Chemformer (KD-Chemformer), and KD using the pretrained Zinc model (KD-Zinc). Finally, it also includes models trained on USPTO-full.

model calls. The exceptions are the template-based and KD models, where low policy probability is more associated with fewer model calls, as well as the KD models with a high percentage of invalid molecule predictions where lower policy probability is also associated with fewer model calls. Fourth, the percentage of searches where a molecule has a solved route is also inversely correlated with single-step top-1 accuracy. Additionally, the percentage solved is associated with the number of model calls, where fewer calls correlate to a higher percentage solved. Finally, choosing the final epoch as done in previous research results in faster but fewer solved routes (Table A6). We observe that using the lowest validation loss checkpoints, with lower top-1 accuracies, but higher top-10 accuracies, results in significant increases in model calls. This is paired with increases in percentage solved, where the baseline FT-Zinc increases from 54.6 to 84.1% when the lowest validation weights are used.

Taken together, an uncertain diverse model may lead to more model calls, as it promotes exploration in the tree search. The multi-step efficiency of a single-step model thus depends on the overall feasibility and quality of the predicted top-10 reactions. Moreover, the ability to properly rank predictions becomes important, something that is difficult to assess in a single-step setting. While all predictions produced by the single-step model affect the multi-step performance, the top 10 only considers the best prediction.

Route Similarity. To investigate the similarity between the predicted routes from different models, the average route similarity was analyzed using a recently published metric.[47] In this analysis, we only compared the top-ranked routes of each target and only made comparisons where both methods produced a route to commercial starting material (Figure 6). Most methods show an average route similarity of approximately 0.7 or higher. This indicates that the methods on average produce routes that share a majority of the synthetic strategy. Thus, these methods would produce routes of similar quality. However, the two KD-Zinc models stand out with their low route similarities to those of the other model routes. This can be explained by these two models producing routes with fewer starting materials relative to the route length, which is not considerably different compared to routes generated by the other models (Table A4). In Figure A1, we show routes for an example target, CHEMBL1668049, that can be synthesized in only one step. The template-based and KD-Chemformer-OptRCS models suggest simple esterification. This is likely the suggestion of KD-Zinc-OptRCS as well, but only one of the reactants is produced. This causes the similarity metric to return zero, as it cannot determine that the bond-forming similarity is in fact identical in all of these three routes. This feature is consistent for the KD-Zinc models; 44.1% of the routes produced by the KD-Zinc-OptRCS model end up in a single starting material, compared to 11.8 and 1.8% for the

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

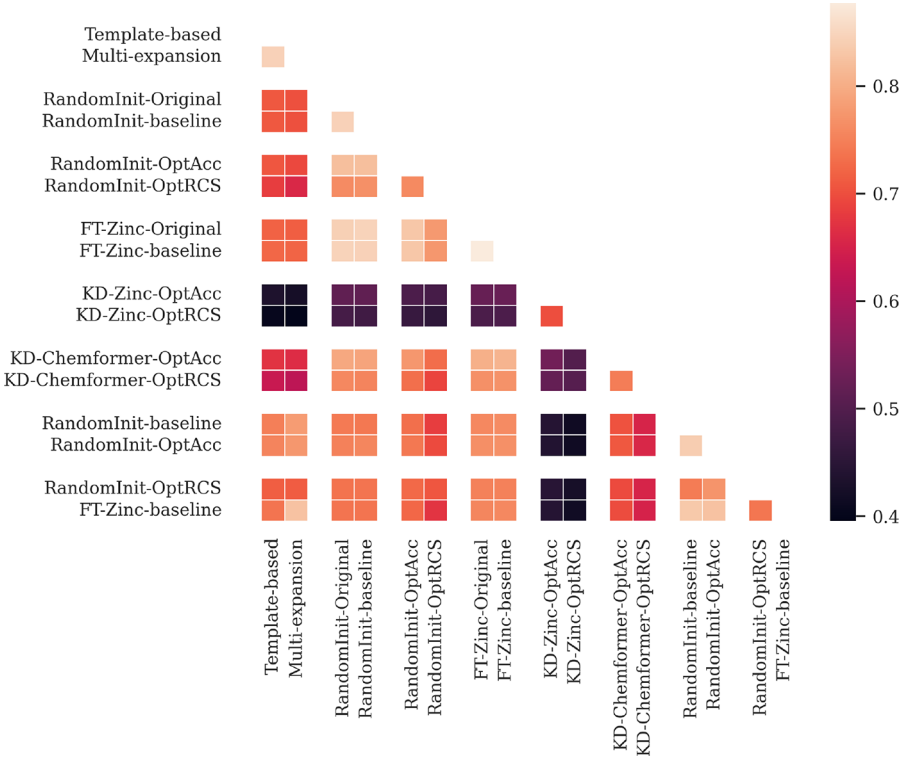


Figure 6. Comparison of route similarity of the highest ranked routes of each model. Route similarity indicates that most models have high route similarity and therefore high route quality, with the exception of KD-Zinc models, which predict significantly less reactants per reaction step.

KD-Chemformer-OptRCS and template-based models, respectively. Five illustrative examples are shown in Figure A2 to give intuition about the general differences between template-based and template-free methods. For some disconnections, the second reactant may be small such that it exists in the stock, e.g., a bromination agent or the reactant in the example above. However, it is likely that the second reactant in some cases requires further disconnections to reach the starting materials.

DISCUSSION

In this study, we adopted alternative transformer architectures and knowledge-distillation (KD) to boost the retrosynthesis prediction efficiency with Chemformer. Model training variations were compared to the original Chemformer, uncovering the complex interplay between the single-step model and multi-step search. Specifically, we focused on the speed-up in the single-step model variants and how this influenced multi-step retrosynthesis performance. This builds on earlier work with the AiZynthFinder tool[12,26] which has been used extensively in drug discovery projects.[4] The models developed herein can furthermore be used in other retrosynthesis frameworks such as Synthesus[14] or ASKCOS.[49]

Single-Step Retrosynthesis Prediction. *Alternative Model Architectures.* The initial findings indicated several important aspects. First, our investigation extended to alternative model architectures, including the Switch Transformer[20] and the Perceiver model.[23] The Switch Transformer showed similar accuracy as the baseline Transformer model but was significantly slower. This was possibly due to a suboptimal implementation and the usage of only a single GPU, making the sparsity effects of the original implementation less effective. The Perceiver model showed promise, having similar accuracy to the transformer trained from scratch while being faster. However, we found that these alternative architectures under-performed when coupled with KD. The Switch transformer and the Perceiver models were chosen as alternative architectures as these architectures were relatively closely related to, though projected to be faster than the original Transformer architecture. The fact that these model variants under-performed when trained with KD indicates that the benefits of KD may not be applicable to all model architectures and emphasizes the need for careful evaluation. One notable aspect here is that the Perceiver model, which uses different dimensions, was unable to use the feature-based KD, and thus solely relied on the response-based KD. Future work could focus on implementing and evaluating the relation-based KD,[16] which uses comparisons of in-between batch distances and angles to match between models, making them more impervious to dimensional changes.

Hyper-Parameter Optimization. During hyper-parameter optimization, we found that the Pareto front contained mostly smaller decoders and larger encoders. This is expected behavior as the inference calculations were based on autoregressive greedy search, which iteratively calls the decoder to make its predictions. Using Optuna, we trained 150 models, including three training settings from random initialization, KD using Chemformer, and KD using the pretrained Zinc model. However, the Optuna search allowed for duplicate parameter suggestions, resulting in on average seven to eight duplicate model settings. This is likely because some models had especially poor validation averages (around 0% accuracy margin). Additionally, no model reached a high accuracy, none higher than 46.7%. This is likely due to the theoretical maximum of the model parameters, where the original implementation parameters were set as the maximum.

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

Finally, the Pareto front mostly included pretrained KD models.

Fully Trained Models and KD. Translating the effects of hyper-parameter optimization to fully trained models, we observed mixed results. First, comparing optimized models with baseline implementations indicated that we were successful in creating increasingly faster models. Interestingly, KD models had around half of the baseline implementations but were similar or even faster than the fastest trained from scratch model with five times fewer model parameters. Another interesting finding is that speed did not necessarily translate one-to-one with decreases in carbon footprints. This is possibly due to the rather basic method of calculating carbon footprints through power usage over the inference time. However, we did observe a general trend where smaller, faster models translate to lower carbon emissions. Moreover, we also observed a substantial increase in the number of invalid smiles generated using smaller models, especially when using KD. This does indicate that a smaller decoder size comes with a cost, and KD might introduce competing objectives during training, meaning that the model might prioritize similarity to the original model over correct SMILES prediction. This could be the reason why the pretrained models dominated the Pareto front but had the worst performance during full-size training. Finally, we observed that using the last epoch for inference on USPTO-50k results in models more optimized on top-1 accuracy, where models using the optimal validation loss (around 100−200 epochs) result in models better optimized for top-10.

Overall in the fully trained models, KD generally leads to lower accuracy compared to standard training methods, which aligns with previous research,[15] possibly due to capacity issues.[50] Relation-based KD could play a bigger role possibly increasing accuracy in future research,[16] especially when considering alternative architecture. Additionally, using the top-10 accuracy in USPTO-50k might be a misleading benchmark, as we observed that the top-1 accuracy was inversely related to the top-10 accuracy in USPTO-50k but not in USPTO-full. This indicates that the long training on USPTO-50k does not translate well to beam search accuracy, as the model becomes overconfident in optimizing for low-diversity, high-accuracy responses over diversity.

Multi-step Retrosynthesis Is Less Dependent on Single-Step Expansion Models than Policy. *Impact of Single-Step Speed in Multi-step.* In this research we also investigated the relation between single-step speed and multi-step speed; however, this relation was less predictive than initially expected. We mostly find that multi-step search times are associated more with a higher number of model calls, which in turn is impacted by higher average policy probabilities. This average policy probability is a reflection of model confidence, thereby influencing the exploration/exploitation trade-off in the Monte Carlo search as it depends on the prediction probability. This is further supported by the smaller number of model calls made by the original models. Future research into increasing multi-step speed might prefer to look into optimizing the exploration/exploitation axis of the multi-step research.

Solvability and Exploration. Furthermore, there is also a relation between the percentage of solved targets (also referred to as solvability or success rate) and the number of calls, which can be interpreted as higher exploration, resulting in more solved routes. However, there also is a slight inverse relationship between the percentage solved and the single-step top-1 accuracy, meaning that the accuracy of the solved routes might not be as precise as routes from higher-accuracy single-step models. However, top-10 is positively related to the number of calls and solvability, especially when looking at using optimal validation loss. This indicates that future research interested in multi-step solvability should focus on top-10 accuracy, not top-1 accuracy in single-step models. Finally, we observe that the models trained on the larger USPTO-full model have an increased percentage solved, which is expected due to their higher chemical space coverage. However, this is also paired with a significantly decreased inference speed, even with identical model parameters. In previous research, the original Chemformer has been shown to under-perform on USPTO-50k compared to other models, but outperform on larger, more complex data sets.[12,13,51] This can likely be explained by the increase in the percentage of unique predictions from beam search, which might also translate to better solvability for more difficult targets. Although solvability does not correspond to route quality, it is a necessary condition to find solved routes,[14,46] and as we focus on comparison between models in this study we leave the assessment of route quality for future studies. We include several examples of retrosynthesis plans in the Support Information for the interested reader and publish all produced routes as a digital download.

Route Similarity. Because the template-based model is used extensively in drug discovery projects,[4] we are confident that this is a good baseline for comparing routes. Therefore, we used a route similarity metric[47] and found that most models produce highly similar routes compared with the baseline template-based model (around 0.7−0.8). In other words, the resulting routes are often very similar regardless of the solvability or speed. The exception to this was the routes generated with KD-Zinc models, which exhibit considerably lower similarity. The reason for this route disparity stems from the propensity of KD-Zinc models to predict only a single reactant for the reactions. Notably, these models were distilled from the pretrained model, which was trained to predict single molecules. These conclusions are supported by both route statistics and the illustrative example given in the route analysis case study.

In general, using single-step models in a multi-step setting, we need to further evaluate and determine the translation of effects of single-step characteristics on multi-step behavior. Specifically, in the field of retrosynthesis, multi-step behavior analysis is still an open question with end-users having various preferences. For example, a reduction in accuracy might be acceptable given the advantages in model size and efficiency gained. Specifically, as mentioned before, single-step retrosynthesis accuracy may not directly translate in a multi-step setting, where diversity might play a more important role[13] and speed might allow for higher exploration. In general, exploration versus time costs should be evaluated further, and exploration should be better calibrated based on single-step models, not necessarily viewed as independent.

CONCLUSIONS

In this research, we investigated the potential of hyper-parameter tuning, alternative model architectures, and knowledge distillation (KD) to accelerate both single-step and multi-step retrosynthesis prediction. The initial motivation was to reduce model size in a system that makes multiple model queries, hypothesizing that speed increases in a single-step

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781

setting would have a compounding effect in a multi-step setting. While KD offered promising speed improvements, these gains were often accompanied by reduced accuracy and the effectiveness of KD varied significantly across different model architectures. This suggests that the benefits of KD are architecture-dependent and that careful tuning is required to avoid undesirable trade-offs. Further investigations into multispeed settings uncovered important aspects of single-step retrosynthesis models that impact multi-step, including top-10 accuracy influence on solvability and the influence of exploration on multi-step speed. Future research into retrosynthesis prediction should focus on optimizing the exploration/exploitation dynamics, particularly in multi-step scenarios, and further refine KD techniques, alternative architectures, and hyper-parameter optimizations to balance speed and accuracy. Investigating hybrid methods that combine the strengths of KD with advanced model architectures may also prove useful in improving both the efficiency and the robustness of retrosynthesis predictions. Although reducing model size and improving single-step speed are important, our results indicate that multi-step search efficiency is more significantly influenced by the diversity and confidence of single-step models. In Monte Carlo-based (MC) multi-step retrosynthesis, these factors play a crucial role in balancing exploration and exploitation during the search process, often outweighing the direct impact of single-step model speed.

■ ASSOCIATED CONTENT

Data Availability Statement

All raw data, generated data and trained models are available on FigShare: 10.6084/m9.figshare.27908229.v1. Project home page: https://github.com/PeterHartog/fast-retro. Operating system(s): Platform independent. Programming language: Python 3. Other requirements: several open source python packages. License: MIT.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c01821

Appendix with Tables and Figures (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Peter B.R. Hartog − *Molecular AI, Discovery Sciences, R&D, AstraZeneca, 431 83 Mölndal, Sweden; Institute of Structural Biology, Molecular Targets and Therapeutics Center, Helmholtz Munich - German Research Center for Environmental Health (GmbH), 85764 Neuherberg, Germany;* orcid.org/0000-0001-6406-6234; Email: peter.hartog@astrazeneca.com

Samuel Genheden − *Molecular AI, Discovery Sciences, R&D, AstraZeneca, 431 83 Mölndal, Sweden;* orcid.org/0000-0002-7624-7363; Email: samuel.genheden@astrazeneca.com

Authors

Annie M. Westerlund − *Molecular AI, Discovery Sciences, R&D, AstraZeneca, 431 83 Mölndal, Sweden;* orcid.org/0000-0003-2288-5711

Igor V. Tetko − *Institute of Structural Biology, Molecular Targets and Therapeutics Center, Helmholtz Munich - German Research Center for Environmental Health (GmbH), 85764 Neuherberg, Germany;* orcid.org/0000-0002-6855-0012

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c01821

Author Contributions

Conceptualization, P.B.R.H., I.V.T., S.G.; Methodology, P.B.R.H., A.M.W., S.G.; Validation, P.B.R.H., A.M.W.; Writing original draft preparation, P.B.R.H.; Writing, review and editing, P.B.R.H., A.M.W., E.S., I.V.T.; Figures: P.B.R.H.; Visualization, P.B.R.H.; Supervision, S.G., I.V.T.; Project administration, S.G., I.V.T.; Funding acquisition, S.G., I.V.T.

Funding

This study has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Actions Innovative Training Network European Industrial Doctorate grant agreement "Advanced machine learning for Innovative Drug Discovery (AIDD)" No. 956832.

Notes

The authors declare no competing financial interest.
All authors have read and agreed to the published version of the manuscript.

■ ACKNOWLEDGMENTS

The author thank the doctoral candidates and supervisors from the Marie Skłodowska-Curie Innovative AIDD consortium for their support.

■ REFERENCES

(1) Strubell, E.; Ganesh, A.; McCallum, A. Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243* **2019**.

(2) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *J. Cheminform.* **2020**, *12*, 70.

(3) Kreutter, D.; Reymond, J. Multistep retrosynthesis combining a disconnection aware triple transformer loop with a route penalty score guided tree search. *Chemical Science* **2023**, *14*, 9959−9969.

(4) Shields, J. D.; Howells, R.; Lamont, G.; Leilei, Y.; Madin, A.; Reimann, C. E.; Rezaei, H.; Reuillon, T.; Smith, B.; Thomson, C.; et al. AiZynth impact on medicinal chemistry practice at AstraZeneca. *RSC Medicinal Chemistry* **2024**, *15*, 1085−1095.

(5) Zhong, Z.; Song, J.; Feng, Z.; Liu, T.; Jia, L.; Yao, S.; Hou, T.; Song, M. Recent advances in deep learning for retrosynthesis. *WIREs: Comput. Mol. Sci.* **2024**, *14*, No. e1694.

(6) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 2017, *30*.

(7) Karpov, P.; Godin, G.; Tetko, I. V. A transformer model for retrosynthesis. *International Conference on Artificial Neural Networks* **2019**, *11731*, 817−830.

(8) Tetko, I. V.; Karpov, P.; Van Deursen, R.; Godin, G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nat. Commun.* **2020**, *11*, 5575.

(9) Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. A. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science* **2019**, *5*, 1572−1583.

(10) Irwin, R.; Dimitriadis, S.; He, J.; Bjerrum, E. J. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology* **2022**, *3*, No. 015022.

(11) Hastedt, F.; Bailey, R. M.; Hellgardt, K.; Yaliraki, S. N.; del Rio Chanona, E. A.; Zhang, D. Investigating the Reliability and

Interpretability of Machine Learning Frameworks for Chemical Retrosynthesis. *Digit. Discov.* **2024**, *3*, 1194−1212.
(12) Westerlund, A. M.; Manohar Koki, S.; Kancharla, S.; Tibo, A.; Saigiridharan, L.; Kabeshov, M.; Mercado, R.; Genheden, S. Do Chemformers Dream of Organic Matter? Evaluating a Transformer Model for Multistep Retrosynthesis. *J. Chem. Inf. Model.* **2024**, *64*, 3021−3033.
(13) Torren-Peraire, P.; Hassen, A. K.; Genheden, S.; Verhoeven, J.; Clevert, D.-A.; Preuss, M.; Tetko, I. V. Models Matter: the impact of single-step retrosynthesis on synthesis planning. *Digital Discovery* **2024**, *3*, 558−572.
(14) Maziarz, K.; Tripp, A.; Liu, G.; Stanley, M.; Xie, S.; Gaiński, P.; Seidl, P.; Segler, M. H. S. Re-evaluating retrosynthesis algorithms with Syntheseus. *Faraday Discuss.* **2025**
(15) Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* **2015**.
(16) Park, W.; Kim, D.; Lu, Y.; Cho, M. Relational knowledge distillation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019; 3967−3976.
(17) Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient transformers: A survey. *ACM Computing Surveys* **2023**, *55*, 1−28.
(18) Eigen, D.; Ranzato, M.; Sutskever, I. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314* **2013**
(19) Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* **2017**
(20) Fedus, W.; Zoph, B.; Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.* **2022**, *23*, 5232−5270.
(21) Lee, J.; Lee, Y.; Kim, J.; Kosiorek, A.; Choi, S.; Teh, Y. W.Set transformer: A framework for attention-based permutation-invariant neural networks. *International conference on machine learning* 2019, 3744−3753.
(22) Wang, S.; Li, B. Z.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768* **2020**
(23) Jaegle, A.; Gimeno, F.; Brock, A.; Vinyals, O.; Zisserman, A.; Carreira, J. Perceiver: General perception with iterative attention. *International conference on machine learning*. 2021; 4651−4664.
(24) Jaegle, A.; Borgeaud, S.; Alayrac, J.-B.; Doersch, C.; Ionescu, C.; Ding, D.; Koppula, S.; Zoran, D.; Brock, A.; Shelhamer, E.et al.Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795* **2021**
(25) Andronov, M.; Andronova, N.; Wand, M.; Schmidhuber, J.; Clevert, D.-A. Accelerating the inference of string generation-based chemical reaction models for industrial applications. *arXiv preprint arXiv:2407.09685* **2024**
(26) Saigiridharan, L.; Hassen, A. K.; Lai, H.; Torren-Peraire, P.; Engkvist, O.; Genheden, S. AiZynthFinder 4.0: developments based on learnings from 3 years of industrial application. *J. Cheminform.* **2024**, *16*, 57.
(27) Sterling, T.; Irwin, J. J. ZINC 15−ligand discovery for everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324−2337.
(28) Genheden, S.; Norrby, P.-O.; Engkvist, O. AiZynthTrain: robust, reproducible, and extensible pipelines for training synthesis prediction models. *J. Chem. Inf. Model.* **2023**, *63*, 1841−1846.
(29) Lowe, D. *Chemical reactions from US patents (1976-Sep2016)*. 2017; https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873.
(30) eMolecules, *Chemical Building Blocks: Chemical Scaffolds*. 2023; https://www.emolecules.com/products/building-blocks, [Accessed 01−01−2023].
(31) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **1988**, *28*, 31−36.
(32) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Félix, E.; Magariños, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic acids research* **2019**, *47*, D930−D940.
(33) Bjerrum, E.; Rastemo, T.; Irwin, R.; Kannas, C.; Genheden, S. PySMILESUtils−Enabling deep learning with the SMILES chemical language. *ChemRxiv*. **2021**
(34) Wiseman, S.; Rush, A. M. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960* **2016**
(35) Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* **2019**
(36) Varuna Jayasiri, N. W. *labml.ai Annotated Paper Implementations*. 2020; https://nn.labml.ai/.
(37) Paszke, A.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8024−8035.
(38) Falcon, W.; *The PyTorch Lightning team, PyTorch Lightning*. 2019; https://github.com/Lightning-AI/lightning.
(39) Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*; Association for Computing Machinery: New York, NY, United States, 2019; 2623−2631.
(40) Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*; Curran Associates, Inc., 2011, *24*.
(41) Bergstra, J.; Yamins, D.; Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *Int. Conf. Mach. Learn.* **2013**, *28*, 115−123.
(42) Westerlund, A. M.; Barge, B.; Mervin, L.; Genheden, S. Data-driven approaches for identifying hyperparameters in multi-step retrosynthesis. *Mol. Inform.* **2023**, *42*, No. e202300128.
(43) Woltz, D. J.; Was, C. A. Availability of related long-term memory during and after attention focus in working memory. *Memory & Cognition* **2006**, *34*, 668−684.
(44) Yang, Z.; Cui, Y.; Chen, Z.; Che, W.; Liu, T.; Wang, S.; Hu, G. TextBrewer: An Open-Source Knowledge Distillation Toolkit for Natural Language Processing. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*; Association for Computational Linguistics, 2020; 9−16.
(45) Budennyy, S.; Lazarev, V.; Zakharenko, N.; Korovin, A.; Plosskaya, O.; Dimitrov, D.; Akhripkin, V.; Pavlov, I.; Oseledets, I.; Barsola, I.; et al. Eco2ai: carbon emissions tracking of machine learning models as the first step towards sustainable ai. *Dokl. Math.* **2022**, *106*, S118−S128.
(46) Genheden, S.; Bjerrum, E. PaRoutes: towards a framework for benchmarking retrosynthesis route predictions. *Digital Discovery* **2022**, *1*, 527−539.
(47) Genheden, S.; Shields, J. D. A Simple Similarity Metric for Comparing Synthetic Routes. *Digit. Discov.* **2025**
(48) Thakkar, A.; Kogej, T.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. J. Datasets and their influence on the development of computer assisted synthesis planning tools in the pharmaceutical domain. *Chemical science* **2020**, *11*, 154−168.
(49) Coley, C. W.; et al. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* **2019**, *365*, No. eaax1566.
(50) Cho, J. H.; Hariharan, B. On the efficacy of knowledge distillation. *Proceedings of the IEEE/CVF international conference on computer vision*; IEEE: Seoul, Korea, 2019; 4794−4802.
(51) Tu, H.; Shorewala, S.; Ma, T.; Thost, V. Retrosynthesis prediction revisited. *NeurIPS 2022 AI for Science: Progress and Promises*. 2022.

https://doi.org/10.1021/acs.jcim.4c01821
J. Chem. Inf. Model. 2025, 65, 1771−1781