JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

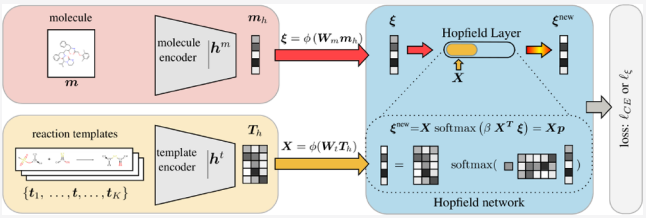
Improving Few- and Zero-Shot Reaction Template Prediction Using Modern Hopfield Networks

Philipp Seidl, Philipp Renz, Natalia Dyubankova, Paulo Neves, Jonas Verhoeven, Jörg K. Wegner, Marwin Segler, Sepp Hochreiter, and Günter Klambauer*

Cite This: J. Chem. Inf. Model. 2022, 62, 2111−2120 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Finding synthesis routes for molecules of interest is essential in the discovery of new drugs and materials. To find such routes, computer-assisted synthesis planning (CASP) methods are employed, which rely on a single-step model of chemical reactivity. In this study, we introduce a template-based single-step retrosynthesis model based on Modern Hopfield Networks, which learn an encoding of both molecules and reaction templates in order to predict the relevance of templates for a given molecule. The template representation allows generalization across different reactions and significantly improves the performance of template relevance prediction, especially for templates with few or zero training examples. With inference speed up to orders of magnitude faster than baseline methods, we improve or match the state-of-the-art performance for top-k exact match accuracy for $k \geq 3$ in the retrosynthesis benchmark USPTO-50k. Code to reproduce the results is available at github.com/ml-jku/mhn-react.



INTRODUCTION

The design of a new molecule starts with an initial idea of a chemical structure with hypothesized desired properties.[1] Desired properties might be the inhibition of a disease or a virus in drug discovery or thermal stability in material science.[2,3] From the design idea of the molecule, a virtual molecule is constructed, the properties of which can then be predicted by means of computational methods.[4,5] However, to test its properties and to finally make use of it, the molecule must be made physically available through chemical synthesis. Finding a synthesis route for a given molecule is a multistep process that is considered highly complex.[6,7]

To aid in finding synthesis routes, chemists have resorted to computer-assisted synthesis planning (CASP) methods.[6,8] Chemical synthesis planning is often viewed in the retrosynthesis setting in which a molecule of interest is recursively decomposed into less complex molecules until only readily available precursor molecules remain.[9] Such an approach relies on a single-step retrosynthesis model, which, given a product, tries to propose sets of reactants from which it can be synthesized. Early methods modeled chemical reactivity using rule-based expert systems.[8] These methods, however, require extensive manual curation.[9−11] Recently, there have been increased efforts to model chemical reactivity from reaction databases using machine learning methods.[9,12−15]

These efforts to model chemical reactions encompass a variety of different approaches. In one line of methods,[14,16−20] the simplified molecular-input line-entry system (SMILES) representation[21] of the reactants given that of the product is predicted, using architectures initially proposed for the translation between natural languages.[22,23] Others exploit the graph structure of molecules and model the task using graph neural networks.[24,25] A prominent line of work makes use of *reaction templates* which are graph transformation rules that encode connectivity changes between atoms during a chemical reaction.

In a template-based approach, reaction templates are first extracted from a reaction database or hand-coded by a chemist. If the product side of a template is a subgraph of a molecule, the template is called applicable to the molecule and can be used to transform it to a reactant set. However, even if a template can be applied to a molecule, the resulting reaction might not be viable in the laboratory.[11] Hence, a core task, which we refer to as template-relevance prediction, in such an approach is to predict with which templates a molecule can be combined with to yield a viable reaction. In prior work, this problem has often been tackled using machine learning methods that are trained at this task on a set of recorded reactions.[9,11,26−32]

Template-based methods usually view the problem as a classification task in which the templates are modeled as distinct categories. However, this can be problematic as automatic template extraction leads to many templates that are

Special Issue: From Reaction Informatics to Chemical Space

Received: September 6, 2021
Published: January 15, 2022

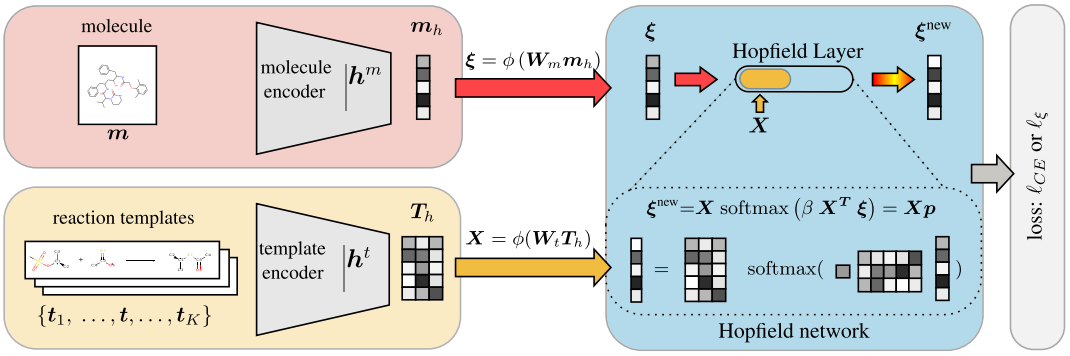


Figure 1. Simplified depiction of our approach. Standard approaches only encode the molecule and predict a fixed set of templates. In our modern Hopfield network (MHN)-based approach, the templates are also encoded and transformed to stored patterns via the template encoder. The Hopfield layer learns to associate the encoded input molecule, the state pattern $\boldsymbol{\xi}$, with the memory of encoded templates, the stored patterns $\mathbf{X}$. Multiple Hopfield layers can operate in parallel or can be stacked using different encoders.

represented by few training samples,[9,26] Somnath et al.[25] argued that template-based approaches suffer from bad performance, particularly for rare reaction templates. Segler[9] and Struble et al.[10] noted that machine learning (ML) has not been applied successfully for CASP in low-data regimes. To address the low-data issue, Fortunato et al.[26] pretrained their template-relevance model to predict which templates are applicable and then fine-tuned it on recorded reactions in a database. This improved template-relevance prediction, especially for rare templates, as well as the average applicability of the top-ranked templates. Overall, a challenge of template-based methods arises from modeling reaction templates as distinct categories, which leads to many classes with few examples (see the section entitled "Methods").

To avoid the above-mentioned problems, we propose a new model that does not consider templates as distinct categories, but can leverage structural information about the template. This allows for generalization over templates and improves performance in the tasks defined in ref 26, especially for templates with few training samples and even for unseen templates. This model learns to associate relevant templates to product molecules using a modern Hopfield network (MHN).[33,34] To this end, we adapted MHNs to associate objects of different modalities, namely input molecules and reaction templates. A depiction of our approach is illustrated in Figure 1.

In contrast to popular ML approaches, in which variable or input-dependent subsets of the data are associated,[22,33,35,36] our architecture maintains a fixed set of representations, considered as a static memory independent of the input.

In this study, we propose a template-based method, which are often reported to be computationally expensive, because of the NP-complete subgraph-isomorphism calculations involved in template execution.[24−26,28] To address this issue Fortunato et al.,[26] Bjerrum et al.[28] trained neural networks to predict which templates are applicable, given a molecule to filter inapplicable templates during inference. We find that using a substructure screen, i.e., a fast check of a necessary condition for a graph to be a subgraph of another improves inference speed, which may also be of interest for other template-based methods.

The main advance of our model over Fortunato et al.,[26] Hasic and Ishida,[37] or other template-based methods, is that by representing and encoding reaction templates we are able to predict relevant templates, even if few training data is available, which is a common issue in reaction datasets.

This work is structured as follows: In the "Methods" section, we propose a template relevance model that predicts template relevance by applying a multimodal learning approach using a modern Hopfield network. In the sections entitled "Template Prediction" and "Single-Step Retrosynthesis", we demonstrate that our architecture improves predictive performance for template relevance prediction and single-step retrosynthesis. In the section enetitled "Inference Speed", we show that our method is several times faster than baseline methods.

SINGLE-STEP RETROSYNTHESIS

The goal of *single-step retrosynthesis* is to predict sets of molecules that react to a given product.[7,38] Since a molecule can be synthesized in various ways, this represents a one-to-many task. Performance in this setting is usually measured by *reactant top-k accuracy* using a reaction database. This metric measures the fraction of samples for which, given the product of a recorded reaction, the recorded reactants are among the top-k predictions. Given the one-to-many setting, small values of k might not be an optimal choice as there might exist scenarios where a good model receives low scores. Choosing a large k might result in a metric that is overly easy to optimize.

Template-based approaches predict reactant sets via reaction templates. A reaction template encodes atom connectivity changes during a chemical reaction and can be used to transform a product molecule to reactants, $\boldsymbol{m} \xrightarrow{t} \boldsymbol{r}$, where $\boldsymbol{m}$ is a product molecule, $\boldsymbol{r}$ represents a set of reactants and $\boldsymbol{t}$ a reaction template. The product side of a template encodes at which position in a molecule the template can be applied. A necessary condition for this is that the product side of the template is a substructure of the molecule of interest. If this is the case, a template is said to be *applicable* to the molecule. The product subgraph is then transformed according to the reactant side of the template and an atom-mapping between the two sides. Templates can be either hand-coded or automatically extracted from reaction databases, which yields an ordered set of K unique templates $\boldsymbol{T} = \{\boldsymbol{t}^k\}_{k=1}^K$.

The aim of *template-relevance prediction* is to predict which templates result in a feasible reaction given a product. If this is the case, we say that a template is *relevant* to a molecule. While applicability is a necessary condition for relevance, it ignores the context of the whole molecule and thus substructures that might conflict with the encoded reaction (see Figure 1 in

https://doi.org/10.1021/acs.jcim.1c01065
J. Chem. Inf. Model. 2022, 62, 2111−2120

Segler and Waller[11]). In practice, applicability gives poor performance at relevance prediction (see Table 1, presented later in this work). To evaluate template-relevance predictions, we use *template top-k accuracy*, which given the product of a recorded reaction measures the fraction of samples for which the template extracted from the recorded reaction is among the top-k predicted ones.

Given relevance predictions for a product, reactant sets are obtained by executing top-scoring templates. We do not permit relevance prediction to rely on applicability calculations, because it is relatively slow to compute. Via this constraint, template top-k accuracy also incorporates information about the models ability to filter out nonapplicable templates. This information might be lost in reactant accuracy as template execution relies on a check for applicability. Other differences between the reactant/template accuracy can arise from multiple locations in which the correct template may be applied or incorrect templates leading to the correct reactants.

Multistep retrosynthesis can be achieved by applying single-step retrosynthesis recursively. One can decompose the desired molecule into less-complex molecules until only readily available precursor molecules remain.

METHODS

Motivation. Many template-based methods[9,11,26−28] consider a classification problem and predict templates using

$$\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{W}\boldsymbol{h}^m(\boldsymbol{m})) \tag{1}$$

where $\boldsymbol{h}^m(\boldsymbol{m})$ is a neural network (NN) that maps a molecule representation to a vector of size d_m, which we call *molecule encoder*. $\boldsymbol{W}$ is a randomly initialized matrix, the last layer of the NN mapping from the molecule encoder to the predictive score of the template classes. Multiplication with $\boldsymbol{W} \in \mathbb{R}^{K\times d_m}$ yields a score for each template $\boldsymbol{t}_1, \cdots, \boldsymbol{t}_K$. These scores are then normalized using the softmax function, which yields the vector $\hat{\boldsymbol{y}} \in \mathbb{R}^K$. In this setting, different templates are viewed as distinct categories or classes, which makes the model ignorant of similarities between classes, which prevents generalization over templates. The high fraction of samples in reaction datasets that have a unique template can be problematic because they cannot contribute to performance. This problem might also appear for templates occurring only a few times, but to a lesser extent.

Instead of learning the rows of $\boldsymbol{W}$ independently, one could map each template to a vector of size d_t using a *template encoder*, $\boldsymbol{h}^t$, and concatenate them row-wise to obtain $\boldsymbol{T}_h = \boldsymbol{h}^t(\boldsymbol{T}) \in \mathbb{R}^{K\times d_t}$. If $d_m = d_t$, replacing $\boldsymbol{W}$ in the equation above yields

$$\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{T}_h\boldsymbol{h}^m(\boldsymbol{m})) \tag{2}$$

which associates the molecule m with each template via the dot product of their representations. This allows generalization across templates, because the structure of the template is used to represent the class and the model can leverage similarities between templates. We adapt modern Hopfield networks[33] to generalize this association of the two modalities, molecules and reaction templates.

Modern Hopfield Networks. By going from eq 1 to eq 2, we have recast the problem of classifying a given molecule into a reaction template class into a content-based retrieval problem. Given a molecule $\boldsymbol{m}$, the correct address, or index, of the molecule's associated template $\boldsymbol{t}$ in a database of templates $\boldsymbol{T}$ must be retrieved based on the chemical structure of the molecule. Such content-addressable, so-called associative memory systems realized as neural networks are called Hopfield networks.[39,40] Their storage capacity can be considerably increased by polynomial terms in the energy function.[41−48] In contrast to these binary memory networks, we use continuous associative memory networks with very high storage capacity. These modern Hopfield networks for deep learning architectures have an energy function with continuous states and can retrieve samples with only one update.[33,49]

For tackling retrieval problems, modern Hopfield networks perform several operations with so-called patterns, i.e., vector representations of the data points. A retrieval model based on a modern Hopfield network can be considered as a function g that returns the position $\hat{\boldsymbol{y}}$ of the retrieved pattern

$$\hat{\boldsymbol{y}} = g(\boldsymbol{h}^m(\boldsymbol{m}), \boldsymbol{h}^t(\boldsymbol{T})) \tag{3}$$

The structure of the function g can be relatively complex,[33,34] but consists of two main components: (a) a mapping to a d-dimensional associative space using linear embeddings $\boldsymbol{W}_m$ and $\boldsymbol{W}_t$ followed by a a non/linear activation ϕ. With these mappings, the state pattern $\boldsymbol{\xi} = \phi(\boldsymbol{W}_m\boldsymbol{h}^m(\boldsymbol{m}))$ and stored patterns $\boldsymbol{X} = \phi(\boldsymbol{W}_t\boldsymbol{h}^t(\boldsymbol{T}))$ are obtained. (b) An update function that performs the following operation,

$$\boldsymbol{\xi}^{\text{new}} = \boldsymbol{X}\boldsymbol{p} = \boldsymbol{X}\ \text{softmax}(\beta\boldsymbol{X}^{\text{T}}\boldsymbol{\xi}) \tag{4}$$

where $\boldsymbol{\xi}^{\text{new}}$ is the retrieved pattern and the stochastic vector $\mathbf{p}$ associates the state pattern with the stored patterns and $\beta > 0$ is a scaling parameter (inverse temperature). The stored patterns $\boldsymbol{X}$ can be considered a memory of reaction templates. Other components, such multiple mappings to associative spaces in parallel, so-called heads, and iterative refinement of the retrieved patterns across multiple layers in the form of stacking are suggested by the powerful transformer architectures,[22] which are also based on Hopfield networks.[33,34] Multiple layers of parallel heads have been shown to be necessary for high predictive quality at natural language processing tasks.[22] This design of the function g allows one to build a DL architecture that is potentially able to retrieve a correct reaction template from an arbitrary set of templates given a molecule.

All these above-mentioned operations and architectural components are implemented in the so-called "Hopfield layer",[33] whose design we use in our model and whose concrete settings are determined during hyperparameter selection. The matrices $\boldsymbol{W}_t$ and $\boldsymbol{W}_m$ that associate molecules and templates are learned during training of the model. In the following, we provide details on the architecture.

Model Architecture. Our model architecture consists of three main parts: (a) a molecule encoder, (b) a reaction template encoder, and (c) one or more stacked or parallel Hopfield layers. First, we use a molecule encoder function that learns a relevant representation for the task at hand. For this, we use a fingerprint-based, e.g., extended connectivity fingerprint (ECFP),[50] fully connected NN, $\boldsymbol{h}^m_{\boldsymbol{w}}(\boldsymbol{m})$ with weights $\boldsymbol{w}$. The molecule encoder maps a molecule to a representation $\boldsymbol{m}_h = \boldsymbol{h}^m_{\boldsymbol{w}}(\boldsymbol{m})$ of dimension d_m.

Second, we use the reaction template encoder $\boldsymbol{h}^t_{\boldsymbol{v}}$ with parameters $\boldsymbol{v}$ to learn relevant representations of templates. Here, we also use a fully connected NN with *template fingerprints* as input. These fingerprints are described in Section S3 in the Supporting Information. This function is

https://doi.org/10.1021/acs.jcim.1c01065
J. Chem. Inf. Model. 2022, 62, 2111−2120

applied to all templates $\boldsymbol{T}$ and the resulting vectors are concatenated column-wise into a matrix $\boldsymbol{T}_h = \boldsymbol{h}_v^t(\boldsymbol{T})$ with shape (d_t, K).

Finally, we use a single or several stacked or parallel Hopfield layers $g(.,.)$ to associate a molecule with all templates in the memory. Hopfield layers consist of the option of layer normalization[51] for $\boldsymbol{\xi}$ and $\boldsymbol{X}$, which is included as a hyperparameter. We also consider the scaling parameter β as a hyperparameter. The Hopfield layer then employs the update rule described by eq 4 through which the updated representation of the product molecule $\boldsymbol{\xi}^{\text{new}}$ and the vector of associations p is obtained. If multiple Hopfield layers are stacked, $\boldsymbol{\xi}^{\text{new}}$ enters the next Hopfield layer, for which additional template encoders supply the template representations. Parallel Hopfield layers use the same template encoder, but learn different projections $\boldsymbol{W}_t\boldsymbol{W}_m$, which is analogous to the heads in Transformer networks.

The simple model (eq 2) is a special case of our MHN and recovered if (a) $\boldsymbol{W}_t$ and $\boldsymbol{W}_m$ are the identity matrices and $d_t = d_m = d$, (b) the Hopfield network is constrained to a single update, (c) Hopfield networks are not stacked, i.e., there is only a single Hopfield layer, (d) the scaling parameter $\beta = 1$, (e) layer norm learns zero mean and unit variance and does not use its adaptive parameters, and (f) the activation function ϕ is the identity. The standard deep neural network (DNN) model (eq 1) is recovered if additionally the reaction templates are one-hot encoded, and the template encoder is linear.

In this study, we tested fingerprint-based fully connected networks for the molecule and template encoder. In principle, one could use any mapping from molecules/templates to vector-valued representations for these components, for example, raw fingerprints, graph neural networks[52] or SMILES arbitrary target specification (SMARTS)-based RNNs,[53] or Transformers.[22]

Loss Function and Optimization. Given a training pair $(\boldsymbol{m},\boldsymbol{t})$ and the set of all templates $\boldsymbol{T}$, the model should assign high probability to $\boldsymbol{t}$, based on $\boldsymbol{m}$ and $\boldsymbol{T}$. We encode this objective by the negative log-likelihood: $-\log p(\boldsymbol{t}|\boldsymbol{m},\boldsymbol{T})$. The probability of each template in $\boldsymbol{T}$ is encoded by the corresponding element of the vector of associations $\boldsymbol{p}$ of the last Hopfield layer. In the simple case of a single correct template, this is equivalent to the cross-entropy loss $l_{\text{CE}}(\boldsymbol{y},\boldsymbol{p})$ between the one-hot encoded label vector $\boldsymbol{y}$ and the predictions $\boldsymbol{p}$. In the case of multiple parallel Hopfield layers, we use average pooling across the vectors $\boldsymbol{p}$ supplied from each layer. We provide a general definition of the loss in terms of retrieved patterns and details in the section entitled "Related Work".

The parameters of the model are adjusted on a training set using stochastic gradient descent on the loss with respect to $\boldsymbol{W}_t,\boldsymbol{W}_m,\boldsymbol{w},v$ via the AdamW optimizer.[54] We train our model for a maximum of 100 epochs and then select the best model with respect to the minimum cross-entropy loss in the case of template-relevance prediction or maximum top-1 accuracy for single-step retrosynthesis on the validation set.

We use dropout regularization in the molecule encoder $\boldsymbol{h}^m$ for the template encoder $\boldsymbol{h}^t$, as well as for the representations in the Hopfield layers. We employ L2 regularization on the parameters. A detailed list of considered and selected hyperparameters is given in Tables S2 and S3 in the Supporting Information.

We added a computationally inexpensive fingerprint-based substructure screen as a post-processing step that can filter out a part of the nonapplicable templates. For each product and the product side of each template, we calculated a bit-vector using the "PatternFingerprint" function from RDKit.[55] Each bit set in this vector specifies the presence of a substructure. For a template to be applicable, every bit set in the template fingerprint also must be set in the product fingerprint, which is a necessary condition for subgraphs to match. We chose a fingerprint size of 4096, as we did not observe significant performance gains for larger sizes, as can be seen in Figure S2 in the Supporting Information.

Related Work. From the perspective of attention-based machine learning,[22] our model can be seen as an attention mechanism that learns to attend elements or patterns of an external memory. The model proposed in Dai et al.[29] is similar to our approach, because it also makes use of the templates' structures and could even be seen as a special case, in which the memory of reaction templates is assembled based on logical operators. There are further restrictions on the structures of the encoder networks, which our work demonstrates are unnecessary. Because of the fact that representations of data from two different modalities, reactions and molecules, are learned, our approach also resembles the contrastive learning approaches taken in CLIP[36] and ConVIRT,[56] in which associated pairs of images and texts are contrasted against nonassociated pairs. Our adaption of MHN to maintain a static memory complements previous contrastive learning[35,57] approaches using a memory.[58−61] To embed molecule and reaction representations in the same latent space by maximizing the cosine similarity of reactions relevant to a given molecule has also been suggested by Segler.[9] We also considered a contrastive learning setting using the InfoNCE loss;[35] however, this led to slightly inferior results (see the Supporting Information).

Data and Preprocessing. All datasets used in this study are derived from the United States Patent and Trademark Office (USPTO) dataset, extracted from the U.S. patent literature between 1976 and September 2016 by Lowe.[62] This dataset contains 1.8 million text-mined reaction equations in SMILES notation[21] and consists of reactions recorded in the years from 1976 to 2016. Reaction conditions and process actions are not included. For evaluating template relevance prediction, we use the preprocessing procedure described in ref 26. Templates are extracted using rdchiral.[63] This results in two datasets: *USPTO-sm*, which is based on USPTO-50k,[64] and *USPTO-lg*, which is based on *USPTO-410k*.[65] *USPTO-50k* contains only the 50 most populated reaction types, yielding a much simpler dataset than *USPTO-lg*, which is more diverse and entails multireactant reactions, which leads to many different templates

For evaluating single-step retrosynthesis, we use *USPTO-50k* as preprocessed in ref 31. For this set, we also extract templates using rdchiral,[63] but only for the train and validation split, to prevent test data leakage. Figure 2 displays the fraction of samples for different template frequencies for *USPTO-sm*/*USPTO-lg*. A detailed description of the datasets and their preprocessing can be found in Section S3 in the Supporting Information.

EXPERIMENTS AND RESULTS

Template Prediction. In this section, we evaluate different models in the setup by Fortunato et al.[26] Here, the aim is to predict the correct reaction templates, with template top-k accuracy used as a metric. In contrast to reactant prediction,

https://doi.org/10.1021/acs.jcim.1c01065
J. Chem. Inf. Model. 2022, 62, 2111−2120

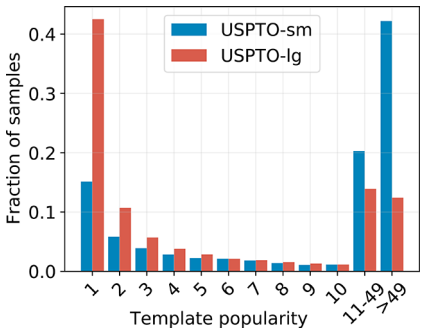


Figure 2. Histogram showing the fraction of samples for different template frequencies. The leftmost red bar indicates that over 40% of chemical reactions of *USPTO-lg* have a unique reaction template. The majority of reaction templates are rare.

this allows a more fine-grained analysis of the template ranking obtained by the models, because it ignores errors stemming from multiple potential application locations. The evaluation of our model in the full reactant prediction task is delayed to the next section. In their study, Fortunato et al.[26] mainly compared two models. First, a fully connected network with a softmax output in which each output unit corresponds to a reaction template, conceptually similar to the model introduced in ref 11. We refer to this model as DNN. The second method is identical to the above, but instead of randomly initializing the weights, pretraining on a template applicability task (DNN +pretrain) is done. We extend this model by the addition of a template encoder and the MHN to associate the entities. We refer to models of the latter type as MHN while calling the former DNN. We also introduced the fingerprint filter (FPF) as a post-processing step. The choice of (a) model type, (b) the use of the FPF, and (c) the pretraining results in $2^3 = 8$ model variants. For all model variants, hyperparameters were adjusted on the validation set, as described in Section S3. We start with a general performance analysis and then investigate how rare templates affect the performance.

Overall Performance. The upper section of Table 1 shows the performance of these eight variants on *USPTO-sm* and *USPTO-lg*. Overall, it can be seen that the use of MHN and FPF yields large performance improvements over the methods evaluated in ref 26. Most notably, the top-100 accuracy increases by 10% on *USTPO-sm* and 18% on *USPTO-lg*. The plain MHN model without both FPF or pretraining has higher top-*k* accuracy for most values of *k* and both datasets, except for top-1 accuracy on *USPTO-lg*, showing the isolated performance gains by the model type. We will further investigate where these performance differences stem from below. Furthermore, the FPF yields non-negligible accuracy improvements for all models. Interestingly, pretraining and the FPF seem to complement each other in predicting applicability for the DNN models, rather than one of them being redundant. While pretraining yields non-negligible performance increases for the DNN models, the effect on the MHN model performance seems rather limited.

In the lower part of Table 1, we show the performance of a simple popularity baseline. This baseline predicts templates based on their occurrence in the training set. The last row shows that a plain applicability check is not sufficient for high performance. We include additional results in Section S3.

Rare Templates: Few- and Zero-Shot Learning. Given the propensity of rare template samples in the used datasets, we next show how the predictive performance varies with template popularity. Figure 3 shows the top-100 accuracy for different subsets of the test set, which were grouped according to the number of training samples with the same template. For improved clarity, we only include four of the above methods: DNN, DNN+pretrain, MHN+FPF, and the popularity baseline. Figure S1 in the Supporting Information shows all eight model variants. Especially for samples with rare templates, the performance gap between our method and the compared ones is large. As expected, in the experiments on template relevance prediction (see the section entitled "Template Prediction" and Figure 3), the DNN models and the popularity baseline perform poorly for samples with templates not seen during training. The MHN model, on the other hand, achieves far above random accuracy on these samples by generalizing over templates. Because of the large fraction of rare template samples, the overall performance is strongly dependent on these.

Table 1. Template Top-*k* Accuracy (%) of Different Method Variants on *USPTO-sm* and *USPTO-lg**

				USPTO-sm			USPTO-lg		
Ref.	Model	Filter	Pretrain	Top-1	Top-10	Top-100	Top-1	Top-10	Top-100
11	DNN	-	no	38.1[a]	64.1[a]	76.5[a]	16.0[b]	35.7[b]	50.7[b]
26	DNN	-	yes	38.5[a]	69.1[a]	85.8[a]	20.8[b]	41.7[b]	54.2[b]
	DNN	FPF	no	39.0[a]	67.6[a]	84.6[a]	17.1[b]	38.1[b]	53.6[b]
	DNN	FPF	yes	38.9[a]	71.2[a]	90.6[a]	**21.5**[b]	43.0[b]	56.0[b]
	MHN	-	no	39.9[a]	75.7[a]	91.9[a]	16.7[b]	43.6[b]	71.4[b]
	MHN	-	yes	**40.4**[a]	76.2[a]	91.8[a]	16.7[b]	43.5[b]	71.4[b]
(ours)	MHN	FPF	no	**40.5**[a]	**78.7**[a]	**95.9**[a]	16.9[b]	**44.2**[b]	**72.4**[b]
	MHN	FPF	yes	**41.3**[a]	**78.8**[a]	**95.7**[a]	17.0[b]	**44.1**[b]	**72.3**[b]
	Pop	-	no	0.0[a]	8.6[a]	28.9[a]	0.1[b]	0.8[b]	3.5[b]
	Pop	FPF	no	1.5[a]	17.6[a]	53.1[a]	0.3[b]	1.9[b]	7.5[b]
	Pop	App[c]	no	9.4[a]	39.6[a]	80.3[a]	1.1[b]	5.1[b]	16.5[b]

*"Model" indicates how the templates were ranked. "Filter" specifies if and how templates were excluded from the ranking via FPF or an applicability check (App). Pre-train indicates whether a model was pre-trained on the applicability task. Error bars represent confidence intervals on binomial proportions. The gray rows indicate methods specifically proposed here or in prior work. [a]Width of 95% confidence interval <1.3%. [b]Width of 95% confidence interval <0.4%. [c]Note that the applicability filter violates modeling constraints from the section entitled "Single-Step Retrosynthesis".

https://doi.org/10.1021/acs.jcim.1c01065
J. Chem. Inf. Model. 2022, 62, 2111−2120

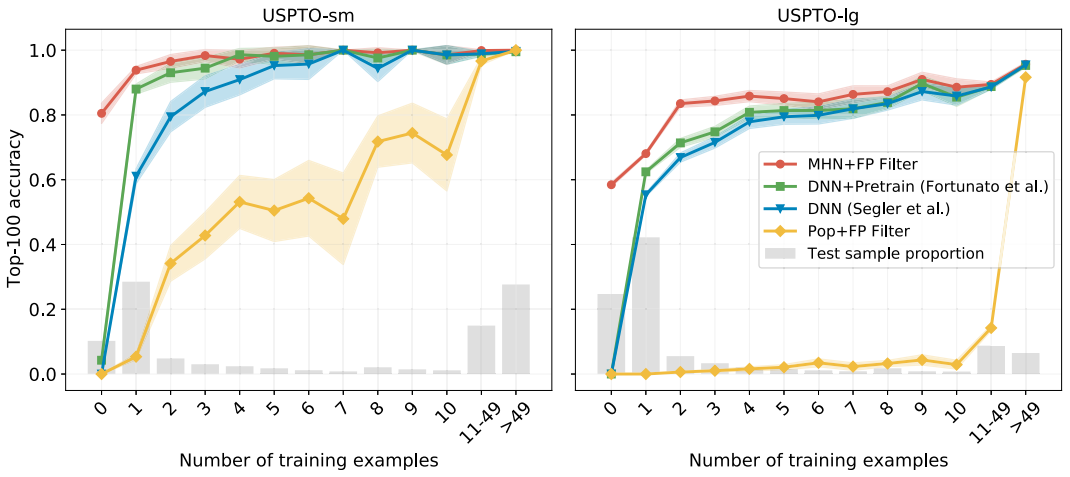


Figure 3. Top-100 accuracy for different template popularity on the *USPTO-sm*/*USPTO-lg* datasets. The gray bars represent the proportion of samples in the test set. Error bars represent 95% confidence intervals on binomial proportion. Our method performs especially well on samples with reaction templates with few training examples.

Table 2. Reactant Top-*k* Accuracy (%) on *USPTO-50k* Retrosynthesis[a]

Abbr.	Ref.	Cat.	Top-1	Top-3	Top-5	Top-10	Top-20	Top-50
MHNreact	ours	tb	$51.8^{\pm.2}$	$\mathbf{74.6}^{\pm.3}$	$\mathbf{81.2}^{\pm.2}$	$\mathbf{88.1}^{\pm.2}$	$\mathbf{92.0}^{\pm.1}$	$\mathbf{94.0}^{\pm.0}$
Neuralsym	11	tb	$45.2^{\pm.2}$	$67.9^{\pm.5}$	$75.8^{\pm.2}$	$83.5^{\pm.2}$	$89.1^{\pm.1}$	$93.5^{\pm.1}$
Pop		tb	18.4	38.7	48.6	63.0	75.8	89.8
Transformer		tf	43.7	59.7	65.1	70.1	73.5	75.0
Dual-TB	66	tb	**55.2**	**74.6**	80.5	86.9		
Dual-TF	66	tf	53.3	69.7	73.0	75.0		
ATx100	20	tf	53.5		81.0	85.7		
GLN	29	tb	52.5	69.0	75.6	83.7	89.0	92.4
RetroPrime	67	tf	51.4	70.8	74.0	76.1		
G2G	24	tf	48.9	67.6	72.5	75.5		
MEGAN	68	tf	48.1	70.7	78.4	86.1	90.3	93.2
GET-LT1	69	tf	44.9	58.8	62.4	65.9		
Neuralsym	11,29	tb	44.4	65.3	72.4	78.9	82.2	83.1
GOPRO	70	tf	43.8	57.2	61.4	66.6		
SCROP	71	tf	43.7	60.0	65.2	68.7		
LV-Trans	72	tf	40.5	65.1	72.8	79.4		
Trans	19	tf	37.9	57.3	62.7			
Retrosim	31	tb	37.3	54.7	63.3	74.1	82.0	85.3

[a]Data taken from refs 11, 19, 20, 24, 29, 31, and 66−72. Bold values indicate values within 0.1 of the maximum value, green denotes a value within 1 percentage point of the maximum value, and yellow denotes a value within 3 percentage points to the maximum value. Error bars represent standard deviations across five reruns. Category ("Cat.") indicates whether a method is template-based (tb) or template-free (tf). Methods in the upper part have been (re-)implemented in this work.

Learning from Rare Templates. Next, we analyze the effect on performance of rare template samples in the training set, as opposed to those in the test set. In a classification setting, it is only useful to include classes if they are recurring, i.e., represented by more than one sample. However, in the *USPTO-sm*/*USPTO-lg* datasets, many templates occur only once (see Figure 2). If the templates are modeled as categories, as done in the DNN approach, a large fraction of samples cannot contribute to performance. However, this does not hold for models that can generalize across templates, as our MHN model is able to do. To show the effect of the rare template samples on learning, we use the following experiment on *USPTO-sm*: We removed all samples with templates that are *exactly once* in the training set and *not* in the test set and retrain the best DNN and MHN models of the template relevance prediction experiment. After removal of these samples, the top-10 accuracy rose from 71.2 ± 0.2 to 72.3 ± 0.2 for the DNN +pretrain model and dropped from 78.8 ± 0.4 to 73.7 ± 0.3 for the MHN model. As expected, the performance does not drop for the DNN model, but even improved marginally, which we attribute to the model knowing which templates do not occur in the test set. In contrast, the performance for the MHN model decreased. This shows that the increased performance of our approach is in part caused by the larger fraction of data that can be leveraged for learning.

Single-Step Retrosynthesis. Next, we compare our method to previously suggested ones in the single-step retrosynthesis task using the *USPTO-50k* and *USPTO-full* datasets. We followed the preprocessing procedure of ref 13 and used rdchiral[63] to extract reaction templates. Following ref

https://doi.org/10.1021/acs.jcim.1c01065
J. Chem. Inf. Model. 2022, 62, 2111−2120

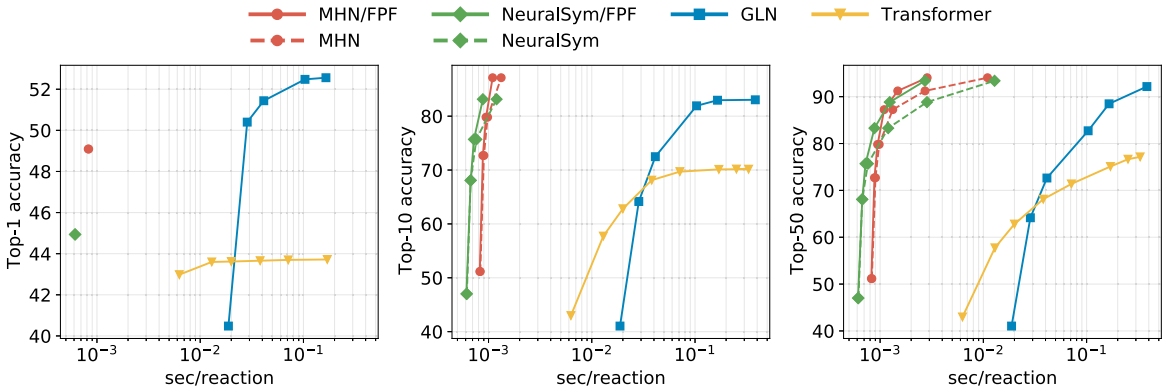


Figure 4. Reactant top-k accuracy versus inference speed for different values of k. Upper left is better. For Transformer/GLN, the points represent different beam sizes. For MHN/NeuralSym, the points reflect different numbers of generated reactant sets, namely, {1, 3, 5, 10, 20, 50}. In case of a Transformer, the points depict different beam sizes: {1, 3, 5, 10, 20, 50, 75, 100}, from left to right.

13, we shuffled the data and assigned 80%/10%/10% of the samples in each reaction class into train/validation/test set, respectively. This is similar to USPTO-sm above but varies in details discussed in section S3. Also, in contrast to template relevance, we optimize for maximum top-1 accuracy, which results in different selected hyperparameters. In addition, we report results of a single run on the *USPTO-full* dataset preprocessed by Tetko et al.[20] (see Table S4 in the Supporting Information). We first compare the predictive performance of our method to previous ones and then investigate its inference speed.

Predictive Performance. Table 2 shows the reactant top-k accuracies on *USPTO-50k* for different methods. These methods include, among others, transformer-based,[20,68] graph-to-graph[67] or template-based ones.[29] Some methods[18,25,37,73−77] that also report results on *USPTO-50k* have been omitted here, either because of test set leakage or different evaluation conditions, as detailed in Section S3. We reimplemented and improved the NeuralSym method as described in Section S3 and added the popularity baseline described in the section entitled "Template Prediction". Hyperparameter selection on the validation set returned an MHN model with two stacked Hopfield layers, which we refer to as *MHNreact* (see Section S3). We ranked reactant sets by the score of the template used to produce them. If a template execution yielded multiple results, all were included in the prediction in random order. Our method achieved state-of-the-art performance for $k \geq 3$ and approaches it for $k = 1, 3$. Together with Dual-TB,[66] this puts template-based methods ahead of other approaches at all considered values of k.

Without the canonicalization procedure of the product-SMILES from the mapped reaction smiles, we obtained a significant increase in performance (Top-1 accuracy of 59.04%). This might suggest leakage, as observed in ref 73, or signal getting lost from canonicalization procedure. This is apparent when using mini-hash fingerprint (MHFP),[78] a SMILES-based fingerprint. For all our experiments, we canonicalize the product-SMILES.

Inference Speed. Aside from predictive performance, inference speed is also vital for retrosynthesis methods. Therefore, CASP methods are often evaluated by their ability to find a route in a given time.[9,28,79] Template-based methods are sometimes reported to be slow;[24,68] however, we found that inference speed was not reported in mentioned studies and generally are seldom reported, despite their importance. Accuracy can be traded for inference speed for many models. For some, this tradeoff is achieved by varying the beam size.[20,29] In template-based approaches, the number of executed templates can be varied and traded off against speed. We compared inference speed of our MHN method with the following baselines. We obtained results for a graph logic network (GLN) from their paper.[24] We trained a Transformer baseline using the code of ref 14, as a representative of transformer-based methods.[19,20,72] In addition, we also include the NeuralSym[11] model that we implemented in the comparison. The results are displayed in Figure 4. At comparable or better performance, our method achieves inference speed of up to two magnitudes faster, compared to the Transformer and GLN. While NeuralSym is faster than our model for some fixed values of accuracy, MHN yields better maximum accuracy with comparable speed.

Computation Time and Resources. All experiments were run on different servers with diverse Nvidia GPUs (Titan V 12GB, P40 24 GB, V100 16GB, A100 20GB MIG), using PyTorch 1.6.0.[80] We estimate the total run time, for all experiments we performed for this study, including baselines, to be ∼1000 GPU hours. A single MHN model can be trained on *USPTO-50k* within ∼5 min on a V100.

DISCUSSION AND CONCLUSION

In this work, we have reformulated the problem of template-based reaction and retrosynthesis prediction as a retrieval problem. We introduced a deep learning architecture for reaction template prediction, based on using modern Hopfield networks. The proposed architecture consists of a molecule encoder, a reaction template encoder network, and Hopfield layers. The best network architectures that were found during hyperparameter selection are typically relatively lightweight, with one or two stacked Hopfield layers, compared to the sizable Transformer architectures.

The retrieval approach enables generalization across templates, which makes zero-shot learning possible and improves few-shot learning. On the single-step retrosynthesis benchmark *USPTO-50k*, our MHN model reaction reaches the state-of-the-art at top-k accuracy for $k \geq 3$. Furthermore, we falsify the common claim of template-based methods being slow.

We note that the current *USPTO-50k* benchmark and its great emphasis on top-1 accuracy for single-step retrosynthesis might only reflect part of what is needed for single-step

https://doi.org/10.1021/acs.jcim.1c01065
J. Chem. Inf. Model. 2022, 62, 2111−2120

retrosynthesis. In cases where a molecule can be made by multiple routes, top-1 accuracy might be too ambiguous; however, the evaluation unrealistically expects to use the one that is present in the dataset. We further argue that there is a tradeoff between having diverse results and having accurate results.[81,82] Unfortunately, it is currently hard to measure diversity, because of not having multiple correct ground-truth answers per product molecule.

Our experiments are currently limited by several factors. We did not investigate the importance of radius around the reaction center used for template extraction. We currently do not rerank reactants based on a secondary model, such as an in-scope filter[9] or dual models,[66] which could increase performance. There would also be several other hyperparameters to be explored, such as the template encoding, whose exploration could lead to an improvement of our method. The results for inference speed are dependent highly on implementation and may potentially be improved by relatively simple means, which was not the primary focus and is left for future work.

Nevertheless, we envision that our approach will be used to improve CASP systems or synthesis-aware generative models.[83−87]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c01065.

> Additional background information and related work (Section S2); details on experiments (Section S3); and additional results for USPTO-full, as well as, e.g., results for a chronological split for USPTO-50k (Section S4); visualizations on a down projection of the association space of reaction templates (Section S5); showcasing of several exemplary test-set samples of USPTO-50k and a reactant ranking comparison, to illustrate differences between methods (Section S6); an alternative view on the loss and an extended formulation of the algorithm as pseudocode (Section S7); details on experiments (e.g., dataset preprocessing, training, and hyperparameter selection, feature extraction from molecules, as well as details on template-fingerprint featurization can be found in the Supporting Information document (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Günter Klambauer − *ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning, Johannes Kepler University Linz, Linz, Austria 4040*; Email: klambauer@ml.jku.at

Authors

Philipp Seidl − *ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning, Johannes Kepler University Linz, Linz, Austria 4040*; orcid.org/0000-0003-4333-2040

Philipp Renz − *ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning, Johannes Kepler University Linz, Linz, Austria 4040*; orcid.org/0000-0002-3323-7632

Natalia Dyubankova − *Janssen Pharmaceutica NV, High Dimensional Biology and Discovery Data Sciences, Janssen Research & Development, Beerse, Belgium 2340*

Paulo Neves − *Janssen Pharmaceutica NV, High Dimensional Biology and Discovery Data Sciences, Janssen Research & Development, Beerse, Belgium 2340*

Jonas Verhoeven − *Janssen Pharmaceutica NV, High Dimensional Biology and Discovery Data Sciences, Janssen Research & Development, Beerse, Belgium 2340*

Jörg K. Wegner − *Janssen Research & Development, LLC, In-Silico Discovery and External Innovation (ISD&EI), Cambridge, Massachusetts 02142, United States*; orcid.org/0000-0002-1852-9434

Marwin Segler − *Microsoft Research, Cambridge, United Kingdom CB1 2FB*; orcid.org/0000-0001-8008-0546

Sepp Hochreiter − *ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning, Johannes Kepler University Linz, Linz, Austria 4040; Institute of Advanced Research in Artificial Intelligence, Wien, Austria 1030*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c01065

Funding

This work was supported by Flanders Innovation & Entrepreneurship (VLAIO) with the project grant (No. HBC.2018.2287) (MaDeSMart).

Notes

The authors declare no competing financial interest.

Data and Software Availability: Python code and instructions to reproduce the predictive results of template prediction as well as single-step retrosynthesis are available at https://github.com/ml-jku/mhn-react under the BSD 2-Clause license. Instructions to prepare the programming-environment, as well as to download the data and run the training, inference, and evaluation procedure can be found there. The code was run on different servers with diverse Nvidia GPUs using PyTorch 1.6.0.[80]

■ ACKNOWLEDGMENTS

The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. IARAI is supported by Here Technologies. We thank the projects AI-MOTION (No. LIT-2018-6-YOU-212), AI-SNN (No. LIT-2018-6-YOU-214), DeepFlood (No. LIT-2019-8-YOU-213), Medical Cognitive Computing Center (MC3), PRIMAL (No. FFG-873979), S3AI (No. FFG-872172), DL for granular flow (No. FFG-871302), ELISE (No. H2020-ICT-2019-3, ID: 951847), AIDD (No. MSCA-ITN-2020, ID: 956832). We thank Janssen Pharmaceutica, Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Laboratories (SAL), FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Software Competence Center Hagenberg GmbH, TÜV Austria, and the NVIDIA Corporation.

■ REFERENCES

(1) Lombardino, J. G.; Lowe, J. A. The Role of the Medicinal Chemist in Drug Discovery — Then and Now. *Nat. Rev. Drug Discovery* **2004**, *3*, 853−862.

(2) Lu, Z.; Chen, X.; Liu, X.; Lin, D.; Wu, Y.; Zhang, Y.; Wang, H.; Jiang, S.; Li, H.; Wang, X.; Lu, Z. Interpretable machine-learning strategy for soft-magnetic property and thermal stability in Fe-based metallic glasses. *npj Comput. Mater.* **2020**, *6*, 1−9.

(3) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-scale

comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.* **2018**, *9*, 5441−5451.
(4) McCammon, J. A. Computer-Aided Molecular Design. *Science* **1987**, *238*, 486−491.
(5) Ng, L. Y.; Chong, F. K.; Chemmangattuvalappil, N. G. Challenges and Opportunities in Computer-Aided Molecular Design. *Comput. Chem. Eng.* **2015**, *81*, 115−129.
(6) Corey, E. J.; Wipke, W. T. Computer-Assisted Design of Complex Organic Syntheses. *Science* **1969**, *166*, 178−192.
(7) Strieth-Kalthoff, F.; Sandfort, F.; Segler, M. H.; Glorius, F. Machine learning the ropes: principles, applications and directions in synthetic chemistry. *Chem. Soc. Rev.* **2020**, *49*, 6154−6168.
(8) Szymkuc, S.; Gajewska, E. P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B. A. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angew. Chem., Int. Ed.* **2016**, *55*, 5904−5937.
(9) Segler, M. H. S.; Preuss, M.; Waller, M. P. Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI. *Nature* **2018**, *555*, 604−610.
(10) Struble, T. J.; Alvarez, J. C.; Brown, S. P.; Chytil, M.; Cisar, J.; DesJarlais, R. L.; Engkvist, O.; Frank, S. A.; Greve, D. R.; Griffin, D. J.; Hou, X.; Johannes, J. W.; Kreatsoulas, C.; Lahue, B.; Mathea, Miriam; Mogk, G.; Nicolaou, C. A.; Palmer, A. D.; Price, D. J.; Robinson, R. I.; Salentin, S.; Xing, L.; Jaakkola, T.; Green, W. H.; Barzilay, R.; Coley, C. W.; Jensen, K. F. Current and Future Roles of Artificial Intelligence in Medicinal Chemistry Synthesis. *J. Med. Chem.* **2020**, *63*, 8667−8682.
(11) Segler, M. H. S.; Waller, M. P. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chemistry* **2017**, *23*, 5966−5971.
(12) Coley, C. W.; Green, W. H.; Jensen, K. F. Machine Learning in Computer-Aided Synthesis Planning. *Acc. Chem. Res.* **2018**, *51*, 1281−1289.
(13) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Cent. Sci.* **2017**, *3*, 1237−1245.
(14) 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 Cent. Sci.* **2019**, *5*, 1572−1583.
(15) Klambauer, G.; Hochreiter, S.; Rarey, M. Machine Learning in Drug Discovery. *J. Chem. Inf. Model.* **2019**, *59*, 945.
(16) Nam, J.; Kim, J. Linking the neural machine translation and the prediction of organic chemistry reactions. *arXiv (Machine Learning)*, **2016**. 1612.09529.
(17) Schwaller, P.; Gaudin, T.; Lányi, D.; Bekas, C.; Laino, T. Found in Translation": Predicting Outcomes of Complex Organic Chemistry Reactions Using Neural Sequence-to-Sequence Models. *Chem. Sci.* **2018**, *9*, 6091−6098.
(18) Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; Pande, V. Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Cent. Sci.* **2017**, *3*, 1103−1113.
(19) Karpov, P.; Godin, G.; Tetko, I. V. A transformer model for retrosynthesis. *Int. Conf. on Artif. Neur. Netw.* **2019**, *11731*, 817−830.
(20) 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.
(21) Weininger, D. SMILES, a Chemical Language and Information System. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.
(22) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, 5998−6008.
(23) Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, 3104−3112.
(24) Shi, C.; Xu, M.; Guo, H.; Zhang, M.; Tang, J. A graph to graphs framework for retrosynthesis prediction. *Int. Conf. Mach. Learn.* **2020**, 8818−8827.
(25) Somnath, V. R.; Bunne, C.; Coley, C. W.; Krause, A.; Barzilay, R. Learning graph models for template-free retrosynthesis. *arXiv (Machine Learning)*, **2020**. 2006.07038.
(26) Fortunato, M. E.; Coley, C. W.; Barnes, B. C.; Jensen, K. F. Data Augmentation and Pretraining for Template-Based Retrosynthetic Prediction in Computer-Aided Synthesis Planning. *J. Chem. Inf. Model.* **2020**, *60*, 3398−3407.
(27) Wei, J. N.; Duvenaud, D.; Aspuru-Guzik, A. Neural Networks for the Prediction of Organic Chemistry Reactions. *ACS Cent. Sci.* **2016**, *2*, 725−732.
(28) Bjerrum, E. J.; Thakkar, A.; Engkvist, O. Artificial Applicability Labels for Improving Policies in Retrosynthesis Prediction. *ChemRxiv*, **2020**. DOI: 10.26434/chemrxiv.12249458.v1.
(29) Dai, H.; Li, C.; Coley, C.; Dai, B.; Song, L. Retrosynthesis Prediction with Conditional Graph Logic Network. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8872−8882.
(30) Baylon, J. L.; Cilfone, N. A.; Gulcher, J. R.; Chittenden, T. W. Enhancing Retrosynthetic Reaction Prediction with Deep Learning Using Multiscale Reaction Classification. *J. Chem. Inf. Model.* **2019**, *59*, 673.
(31) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Cent. Sci.* **2017**, *3*, 1237−1245.
(32) Ishida, S.; Terayama, K.; Kojima, R.; Takasu, K.; Okuno, Y. Prediction and Interpretable Visualization of Retrosynthetic Reactions Using Graph Convolutional Networks. *J. Chem. Inf. Model.* **2019**, *59*, 5026−5033.
(33) Ramsauer, H.; Schäfl, B.; Lehner, J.; Seidl, P.; Widrich, M.; Gruber, L.; Holzleitner, M.; Adler, T.; Kreil, D.; Kopp, M. K.; Klambauer, G.; Brandstetter, J.; Hochreiter, S. Hopfield Networks is All You Need. *Int. Conf. Learn. Rep.*, **2021**. https://openreview.net/forum?id=tL89RnzIiCd.
(34) Widrich, M.; Schäfl, B.; Pavlovic, M.; Ramsauer, H.; Gruber, L.; Holzleitner, M.; Brandstetter, J.; Sandve, G. K.; Greiff, V.; Hochreiter, S.; Klambauer, G. Modern Hopfield Networks and Attention for Immune Repertoire Classification. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18832−18845.
(35) Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. *Int. Conf. Mach. Learn.* **2020**, 1597−1607.
(36) Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; Sutskever, I. Learning transferable visual models from natural language supervision. *arXiv (Machine Learning)*, **2021**. 2103.00020.
(37) Hasic, H.; Ishida, T. Single-Step Retrosynthesis Prediction Based on the Identification of Potential Disconnection Sites Using Molecular Substructure Fingerprints. *J. Chem. Inf. Model.* **2021**, *61*, 641−652.
(38) Segler, M. H. World programs for model-based learning and planning in compositional state and action spaces. *arXiv (Machine Learning)*, **2019**. 1912.13007.
(39) Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U. S. A.* **1982**, *79*, 2554−2558.
(40) Graves, A.; Wayne, G.; Danihelka, I. Neural Turing Machines. *arXiv (Machine Learning)*, **2014**. 1410.5401.
(41) Chen, H. H.; Lee, Y. C.; Sun, G. Z.; Lee, H. Y.; Maxwell, T.; Giles, C. L. High order correlation model for associative memory. *AIP Conf. Proc.* **1986**, *151*, 86−99.
(42) Psaltis, D.; Park, C. H. Nonlinear discriminant functions and associative memories. *AIP Conf. Proc.* **1986**, *151*, 370−375.
(43) Baldi, P.; Venkatesh, S. S. Number of stable points for spin-glasses and neural networks of higher orders. *Phys. Rev. Lett.* **1987**, *58*, 913−916.
(44) Gardner, E. Multiconnected neural network models. *J. Phys. A* **1987**, *20*, 3453−3464.
(45) Abbott, L. F.; Arian, Y. Storage capacity of generalized networks. *Phys. Rev. A* **1987**, *36*, 5091−5094.

(46) Horn, D.; Usher, M. Capacities of multiconnected memory models. *J. Phys. (Paris)* **1988**, *49*, 389−395.
(47) Caputo, B.; Niemann, H. Storage Capacity of Kernel Associative Memories. *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*. Berlin, Heidelberg, 2002; p 51−56.
(48) Krotov, D.; Hopfield, J. J. Dense associative memory for pattern recognition. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1172−1180.
(49) Ramsauer, H.; Schäfl, B.; Lehner, J.; Seidl, P.; Widrich, M.; Adler, T.; Gruber, L.; Holzleitner, M.; Pavlovic, M.; Sandve, G. K.; Greiff, V.; Kreil, D.; Kopp, M.; Klambauer, G.; Brandstetter, J.; Hochreiter, S. Hopfield networks is all you need. *arXiv (Machine Learning)*, **2020**. 2008.02217.
(50) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(51) Ba, J. L.; Kiros, J. R.; Hinton, G. E. Layer normalization. *arXiv (Machine Learning)*, **2016**. 1607.06450.
(52) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. *Int. Conf. Mach. Learn.* **2017**, 1263−1272.
(53) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-Scale Comparison of Machine Learning Methods for Drug Target Prediction on ChEMBL. *Chem. Sci.* **2018**, *9*, 5441.
(54) Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv (Machine Learning)*, **2017**. 1711.05101.
(55) Landrum, G. *RDKit: Open-Source Cheminformatics*. 2006, accessed on Jan. 1, 2020.
(56) Zhang, Y.; Jiang, H.; Miura, Y.; Manning, C. D.; Langlotz, C. P. Contrastive learning of medical visual representations from paired images and text. *arXiv (Machine Learning)*, **2020**. 2010.00747.
(57) Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. *Proc. IEEE* **2006**, *2*, 1735−1742.
(58) Wu, Z.; Xiong, Y.; Yu, S. X.; Lin, D. Unsupervised Feature Learning via Non-parametric Instance Discrimination. *Proc. IEEE* **2018**, 3733−3742.
(59) Misra, I.; van der Maaten, L. Self-supervised learning of pretext-invariant representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, 2020; pp 6707−6717, .
(60) He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA2020; pp 9726−9735, .
(61) Tian, Y.; Krishnan, D.; Isola, P. Contrastive multiview coding. *Comp. Vision ECCV* **2020**, *12356*, 776−794.
(62) Lowe, D. M. Extraction of chemical structures and reactions from the literature. Ph.D. Thesis, University of Cambridge, 2012.
(63) Coley, C. W.; Green, W. H.; Jensen, K. F. RDChiral: An RDKit Wrapper for Handling Stereochemistry in Retrosynthetic Template Extraction and Application. *J. Chem. Inf. Model.* **2019**, *59*, 2529−2537.
(64) Schneider, N.; Stiefl, N.; Landrum, G. A. What's What: The (Nearly) Definitive Guide to Reaction Role Assignment. *J. Chem. Inf. Model.* **2016**, *56*, 2336−2346.
(65) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A Graph-Convolutional Neural Network Model for the Prediction of Chemical Reactivity. *Chem. Sci.* **2019**, *10*, 370−377.
(66) Sun, R.; Dai, H.; Li, L.; Kearnes, S.; Dai, B. Energy-based View of Retrosynthesis. *arXiv (Machine Learning)*, **2020**. 2007.13437.
(67) Wang, X.; Li, Y.; Qiu, J.; Chen, G.; Liu, H.; Liao, B.; Hsieh, C.-Y.; Yao, X. RetroPrime: A Diverse, Plausible and Transformer-Based Method for Single-Step Retrosynthesis Predictions. *Chem. Eng. J.* **2021**, *420*, 129845.
(68) Sacha, M.; Błaz, M.; Byrski, P.; Dabrowski-Tumanski, P.; Chrominski, M.; Loska, R.; Włodarczyk-Pruszynski, P.; Jastrzebski, S. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *J. Chem. Inf. Model.* **2021**, *61*, 3273−3284.
(69) Mao, K.; Zhao, P.; Xu, T.; Rong, Y.; Xiao, X.; Huang, J. Molecular Graph Enhanced Transformer for Retrosynthesis Prediction. *bioRxiv*, **2020**. DOI: 10.1101/2020.03.05.979773.
(70) Mann, V.; Venkatasubramanian, V. Retrosynthesis Prediction Using Grammar-Based Neural Machine Translation: An Information-Theoretic Approach. *ChemRxiv*, **2021**. DOI: 10.26434/chemrxiv.14410442.v1.
(71) Zheng, S.; Rao, J.; Zhang, Z.; Xu, J.; Yang, Y. Predicting Retrosynthetic Reactions Using Self-Corrected Transformer Neural Networks. *J. Chem. Inf. Model.* **2020**, *60*, 47−55.
(72) Chen, B.; Shen, T.; Jaakkola, T. S.; Barzilay, R. Learning to Make Generalizable and Diverse Predictions for Retrosynthesis. *arXiv (Machine Learning)*, **2019**. 1910.09688.
(73) Yan, C.; Ding, Q.; Zhao, P.; Zheng, S.; Yang, J.; Yu, Y.; Huang, J. RetroXpert: Decompose Retrosynthesis Prediction like a Chemist. *arXiv (Machine Learning)*, **2020**. 2011.02893.
(74) Lee, H.; Ahn, S.; Seo, S.-W.; Song, Y. Y.; Hwang, S.-J.; Yang, E.; Shin, J. RetCL: A Selection-Based Approach for Retrosynthesis via Contrastive Learning. *arXiv (Machine Learning)*, **2021**. 2105.00795.
(75) Guo, Z.; Wu, S.; Ohno, M.; Yoshida, R. A Bayesian Algorithm for Retrosynthesis. *J. Chem. Inf. Model.* **2020**, *60*, 4474−4486.
(76) Ishiguro, K.; Ujihara, K.; Sawada, R.; Akita, H.; Kotera, M. Data Transfer Approaches to Improve Seq-to-Seq Retrosynthesis. *arXiv (Machine Learning)*, **2020**. 2010.00792.
(77) Ucak, U. V.; Kang, T.; Ko, J.; Lee, J. Substructure-based neural machine translation for retrosynthetic prediction. *J. Cheminf.* **2021**, *13*, 4.
(78) Probst, D.; Reymond, J.-L. A probabilistic molecular fingerprint for big data settings. *J. Cheminf.* **2018**, *10*, 1−12.
(79) Chen, B.; Li, C.; Dai, H.; Song, L. Retro*: learning retrosynthetic planning with neural guided A* search. *Int. Conf. Mach. Learn.* **2020**, 1608−1616.
(80) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026−8037.
(81) Kunaver, M.; Pozrl, T. Diversity in Recommender Systems, A Survey. *Knowl.-Based Syst.* **2017**, *123*, 154−162.
(82) Isufi, E.; Pocchiari, M.; Hanjalic, A. Accuracy-diversity trade-off in recommender systems via graph convolutions. *Inf. Process. Manage.* **2021**, *58*, 102459.
(83) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M.; Hernández-Lobato, J. M. A Model to Search for Synthesizable Molecules. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 7937−7949.
(84) Gottipati, S. K.; Sattarov, B.; Niu, S.; Pathak, Y.; Wei, H.; Liu, S.; Thomas, K. M. J.; Blackburn, S.; Coley, C. W.; Tang, J.; Chandar, S.; Bengio, Y. Learning To Navigate The Synthetically Accessible Chemical Space Using Reinforcement Learning. *arXiv (Machine Learning)*, **2020**. 2004.12485.
(85) Horwood, J.; Noutahi, E. Molecular Design in Synthetically Accessible Chemical Space via Deep Reinforcement Learning. *ACS Omega* **2020**, *5*, 32984−32994.
(86) Renz, P.; Van Rompaey, D.; Wegner, J. K.; Hochreiter, S.; Klambauer, G. On Failure Modes in Molecule Generation and Optimization. *Drug Discovery Today: Technol.* **2019**, *32*−*33*, 55−63.
(87) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M. H.; Hernández-Lobato, J. M. Barking up the right tree: an approach to search over molecule synthesis dags. *arXiv Preprint* **2020**, arXiv:2012.11522.