Context-enriched molecule representations improve few-shot drug discovery

Anonymous Author(s)
Affiliation
Address
email

Abstract

A central task in computational drug discovery is to construct models from known active molecules to find further promising molecules for subsequent screening. However, typically only very few active molecules are known. Therefore, few-shot learning methods have the potential to improve the effectiveness of this critical phase of the drug discovery process. We introduce a new method for few-shot drug discovery. Its main idea is to enrich a molecule representation by knowledge about known context or reference molecules. Our novel concept for molecule representation enrichment is to associate molecules from both the support set and the query set with a large set of reference (context) molecules through a modern Hopfield network. Intuitively, this enrichment step is analogous to a human expert who would associate a given molecule with familiar molecules whose properties are known. The enrichment step reinforces and amplifies the covariance structure of the data and simultaneously removes spurious correlations arising from the decoration of molecules. We analyze our novel method on FS-Mol, which is the only established few-shot learning benchmark dataset for drug discovery. An ablation study shows that the enrichment step of our method is key to improving the predictive quality. In a domain shift experiment, our new method is more robust than other methods. On FS-Mol, our new method achieves a new state-of-the-art and outperforms all other few-shot methods.

1 Introduction

To improve human health, combat diseases and tackle pandemics, there is a steady need of discovering new drugs in a fast and efficient way. The drug discovery process is time-consuming and cost-intensive (Arrowsmith, 2011). Deep Learning methods have recently been shown to reduce time and monetary resources (Chen et al., 2018; Walters and Barzilay, 2021). They diminish the required number of both wet-lab measurements and molecules that must be synthesized (Merk et al., 2018; Schneider et al., 2020). However, they use only the molecular information about the ligands after being trained on a large training set. At inference time, they yield highly accurate property and activity prediction (Mayr et al., 2018; Yang et al., 2019), generative (Segler et al., 2018a; Gómez-Bombarelli et al., 2018), or synthesis models (Segler et al., 2018b; Seidl et al., 2021).

Deep Learning methods in drug discovery usually require large amounts of biological measurements. To train Deep Learning-based activity and property prediction models with high predictive performance, hundreds or thousands of data points per task are required. For example, well-performing predictive models for activity prediction tasks of ChEMBL have been trained with an average of 3,621 activity points per task, i.e. drug target, by Mayr et al. (2018). The ExCAPE-DB dataset provides on average 42,501 measurements per task (Sun et al., 2017; Sturm et al., 2020). Wu et al. (2018) published a large scale benchmark for molecular machine learning, including among others prediction

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

models for the SIDER dataset with an average of 5,187 data points, Tox21 (Huang et al., 2016b; Mayr et al., 2016) with on average 9,031, and ClinTox (Wu et al., 2018) with 1,491 measurements per task. However, for typical drug design projects, the amount of available measurements is very limited (Stanley et al., 2021; Waring et al., 2015; Hochreiter et al., 2018), since in-vitro experiments are expensive and time-consuming. Therefore, methods that need only few measurements to build precise property and activity prediction models are desirable. This problem of learning from few data points is the focus of machine learning areas like meta-learning (Schmidhuber, 1987; Bengio et al., 1991; Hochreiter et al., 2001) and few-shot learning (Miller et al., 2000; Bendre et al., 2020; Wang et al., 2020).

Few-shot learning tackles the low-data problem that is ubiquitous in drug discovery. Few-shot learning methods have been predominantly developed and tested on image datasets (Bendre et al., 2020; Wang et al., 2020), and have recently been adapted to drug discovery problems (Stanley et al., 2021; Altae-Tran et al., 2017). They are usually categorized into three groups according to their main approach (Bendre et al., 2020; Wang et al., 2020; Adler et al., 2020). a) Data-augmentation-based approaches counter the problem of too few data by augmenting the available samples and generating new, more diverse data points (Chen et al., 2020; Zhao et al., 2019; Antoniou and Storkey, 2019). b) Embedding-based and nearest neighbour approaches learn representations in a meaningful embedding space. If the embeddings have a proper representation, then models can be constructed from only few new data points by comparing these embeddings. For Matching Networks (Vinyals et al., 2016), e.g., an attention mechanism, which relies on these embeddings, is the basis for the predictions. Prototypical Networks (Snell et al., 2017) create prototype representations for each class, using the above mentioned representations in the embedding space. c) Optimization-based or fine-tuning methods utilize a meta-optimizer that focuses on efficiently navigating the parameter space. For example, with MAML the meta-optimizer learns initial weights that can be adapted to a novel task by few optimization steps (Finn et al., 2017). Nguyen et al. (2020) evaluated the applicability of MAML and its variants to graph neural networks (GNNs) and Guo et al. (2021) also combine GNNs and meta-learning. Altae-Tran et al. (2017) suggested an approach called Iterative Refinement Long Short-Term Memory, in which query and support set embeddings can share information and update their embeddings. Recently, Stanley et al. (2021) generated a benchmark dataset for few-shot learning methods in drug discovery and provided some baseline results. Surprisingly, almost all these few-shot learning methods, are worse than a naive baseline, that does not even use the support set (see Section 4). We hypothesize that this under-performance of these methods stems from disregarding the context, both in terms of similar molecules and similar activities. Therefore, we propose a method that informs the representations of the query and support set with a large number of context molecules covering the chemical space.

Enriching molecule representations with context using associative memories. Especially in situations with few data, humans extract co-occurrences and covariances by associating current perceptions with memories (Bonner and Epstein, 2021; Potter, 2012). In drug discovery, a human expert, when shown a small set of active molecules, also associates them with known molecules and their properties to suggest further active molecules (Gomez, 2018; He et al., 2021). In an analogous manner, our novel concept for few-shot learning uses associative memories to extract co-occurrences and the covariance structure of the original data and to amplify them in the representations (Fürst et al., 2021). We use Modern Hopfield Networks (MHNs) as a associative memory, since they can store a large set of context molecule representations (Ramsauer et al., 2020, Theorem 3). The representations that are retrieved from the MHNs replace the original representations of the query and support set molecules. Those retrieved representations have amplified co-occurrences and covariance structures, while peculiarities and spurious co-occurrences of the query and support set molecules are averaged out.

In this work, our contributions are the following:

- We propose a new architecture **MHNfs** for few-shot learning in drug discovery.
- We achieved a new state-of-the-art on the benchmarking dataset FS-Mol.
- We introduce a novel concept to enrich the molecule representations with context by associating them with a large set of context molecules.
- We add a naive baseline to the FS-Mol benchmark that yields better results than almost all other published few-shot learning methods.

- We provide results of an ablation study and an domain shift experiment to further demonstrate the effectiveness of our new method.

2 Problem setting

Drug discovery projects revolve around models $g(\boldsymbol{m})$ that can predict a molecular property or activity $\hat{y}$ given a representation $\boldsymbol{m}$ of an input molecule from a chemical space $\mathcal{M}$. We consider machine-learning models $\hat{y} = g_{\boldsymbol{w}}(\boldsymbol{m})$ with parameters $\boldsymbol{w}$ that have been selected using a training set. Typically, Deep Learning based property prediction uses a molecule encoder $f^{\mathrm{ME}} : \mathcal{M} \to \mathbb{R}^d$. The molecule encoder can process different symbolic or low-level representations of molecules, such as molecular descriptors (Bender et al., 2004; Unterthiner et al., 2014; Mayr et al., 2016), SMILES (Weininger, 1988; Mayr et al., 2018; Winter et al., 2019; Segler et al., 2018a), or molecular graphs (Merkwirth and Lengauer, 2005; Kearnes et al., 2016; Yang et al., 2019; Jiang et al., 2021) and can be pre-trained on related property prediction tasks.

For few-shot learning, the goal is to select a high-quality predictive model based on a small set of molecules $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ with associated measurements $\boldsymbol{y} = \{y_1, \ldots, y_N\}$. The measurements are usually assumed to be binary $y_n \in \{-1, 1\}$, corresponding to the molecule being inactive or active. The set $\{\boldsymbol{X}, \boldsymbol{y}\}$ is called the *support set* that contains samples from a prediction task. Goal is to construct a model that correctly predicts $\boldsymbol{y}$ for a $\boldsymbol{x}$ that is not in the support set, which means the model that generalizes well.

Standard supervised machine learning approaches fail at this task (Stanley et al., 2021). These approaches learn the parameters $\boldsymbol{w}$ of the model $g_{\boldsymbol{w}}$ from the support set in a supervised manner. However, they heavily overfit to the support set when N is small. Therefore, few-shot learning methods are necessary to construct models from the support set that generalize well to new data.

3 MHNfs: Hopfield-based molecular context enrichment for few-shot drug discovery

We aim at increasing the generalization capabilities of few-shot learning methods in drug discovery by enriching the molecule representations with molecular context. This context, i.e. a large set of molecules that covers the chemical space, is provided to the model within a context module, in which the query and support set molecules perform a retrieval from the context set and thereby enrich their representations. We detail this in the following.

3.1 Model architecture

We propose to include the molecular context via three consecutive modules that feed into each other. The first module, a) the *context module* f^{CM}, enriches molecule representations by retrieving from a large set of molecules. The second module, b) the *cross-attention module* f^{CAM} (Hou et al., 2019; Chen et al., 2021), enables the effective exchange of information between the query molecule and the support set molecules. Finally the prediction for the query molecule is computed by using the usual c) *similarity module* f^{SM} (Koch et al., 2015; Altae-Tran et al., 2017):

$$\begin{aligned}
\text{context module:} \quad & \boldsymbol{m}' = f^{\mathrm{CM}}(\boldsymbol{m}, \boldsymbol{C}) \\
& \boldsymbol{X}' = f^{\mathrm{CM}}(\boldsymbol{X}, \boldsymbol{C}), && (1) \\
\text{cross-attention module:} \quad & [\boldsymbol{m}'', \boldsymbol{X}''] = f^{\mathrm{CAM}}([\boldsymbol{m}', \boldsymbol{X}']), && (2) \\
\text{similarity module:} \quad & \hat{y} = f^{\mathrm{SM}}(\boldsymbol{m}'', \boldsymbol{X}'', \boldsymbol{y}), && (3)
\end{aligned}$$

where $\boldsymbol{m} \in \mathbb{R}^d$ is a molecule embedding from a trainable or fixed molecule encoder, and $\boldsymbol{m}'$ and $\boldsymbol{m}''$ are enriched versions of it. Similarly, $\boldsymbol{X} \in \mathbb{R}^{d \times N}$ contains the stacked embeddings of the support set molecules and $\boldsymbol{X}'$ and $\boldsymbol{X}''$ are their enriched versions. $\boldsymbol{C} \in \mathbb{R}^{d \times M}$ is a large set of stacked molecule embeddings, $\boldsymbol{y}$ are the support set labels, and $\hat{y}$ is the prediction for the query molecule. Square brackets indicate concatenation, for example $[\boldsymbol{m}', \boldsymbol{X}']$ is a matrix with $N + 1$ columns. The modules f^{CM}, f^{CAM}, and f^{SM} are detailed in the paragraphs below. An overview of our architecture is given

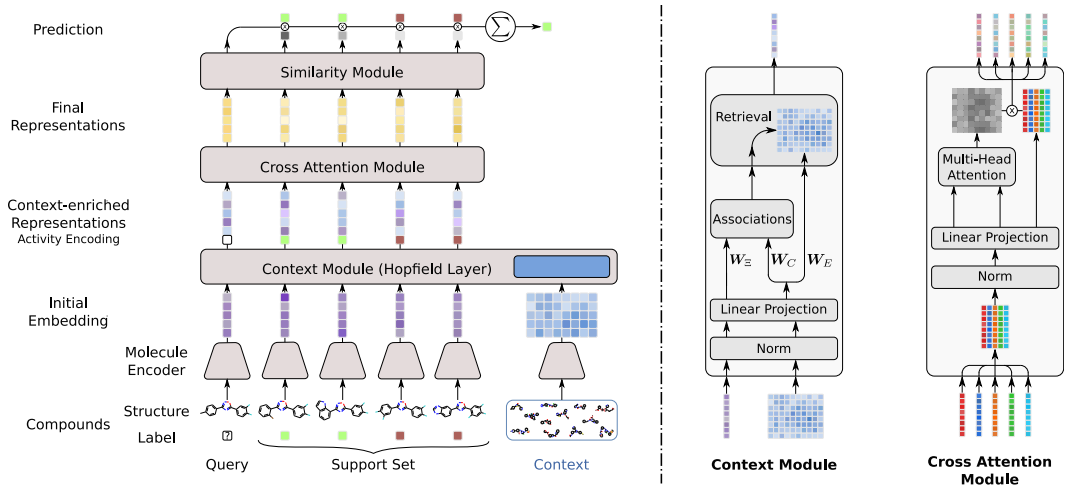

Figure 1: Schematic overview of our architecture. **Left:** All molecules are fed through a shared molecule encoder to obtain embeddings. Then, the context module (CM) enriches the representations by associating them with context molecules. The cross-attention module (CAM) enriches representations by mutually associating the query and support set molecules. Finally, the similarity module computes the prediction for the query molecule. **Right:** Detailed depiction of the operations in the CM and the CAM.

in Figure 1. The architecture also includes skip connections bypassing $f^{\mathrm{CM}}(.,.)$ and $f^{\mathrm{CAM}}(.)$, which are not shown in the figure.

A shared molecule encoder f^{ME} creates embeddings for the query molecule $\boldsymbol{m} = f^{\mathrm{ME}}(m)$, the support set molecules $\boldsymbol{x}_n = f^{\mathrm{ME}}(x_n)$, and the context molecules $\boldsymbol{c}_m = f^{\mathrm{ME}}(c_m)$. There are many possible choices for fixed or adaptive molecule encoders (see Section 2), of which we use descriptor-based fully-connected networks because of their computational efficiency and good accuracy (Dahl et al., 2014; Mayr et al., 2016, 2018). For notational clarity we denote the course of the representations through the architecture:

$$\underset{\text{symbolic or low-level repr.}}{m} \xrightarrow{f^{\mathrm{ME}}} \underset{\text{molecule embedding}}{\boldsymbol{m}} \xrightarrow{f^{\mathrm{CM}}} \underset{\text{context repr.}}{\boldsymbol{m}'} \xrightarrow{f^{\mathrm{CAM}}} \underset{\text{similarity repr.}}{\boldsymbol{m}''}, \tag{4}$$

$$\underset{\text{symbolic or low-level repr.}}{x_n} \xrightarrow{f^{\mathrm{ME}}} \underset{\text{molecule embedding}}{\boldsymbol{x}_n} \xrightarrow{f^{\mathrm{CM}}} \underset{\text{context repr.}}{\boldsymbol{x}'_n} \xrightarrow{f^{\mathrm{CAM}}} \underset{\text{similarity repr.}}{\boldsymbol{x}''_n}. \tag{5}$$

3.2 Context module (CM)

The context module associates the query and support set molecules with a large set of context molecules, and represents them as weighted average of context molecule embeddings. The context module is realised by a Modern Hopfield Network (MHN) (Ramsauer et al., 2020). An MHN is an associative memory which is capable of storing and retrieving patterns from a memory $\boldsymbol{M} \in \mathbb{R}^{e \times M}$ given a state pattern $\boldsymbol{\xi} \in \mathbb{R}^e$ that represents the query. The retrieved pattern $\boldsymbol{\xi}^{\mathrm{new}}$ is obtained by

$$\boldsymbol{\xi}^{\mathrm{new}} = \boldsymbol{M}\,\boldsymbol{p} = \boldsymbol{M}\ \mathrm{softmax}\left(\beta \boldsymbol{M}^T \boldsymbol{\xi}\right), \tag{6}$$

where $\boldsymbol{p}$ is called the vector of associations and β is a scaling factor or inverse temperature. Modern Hopfield Networks have been successfully applied to chemistry and computational immunology(Seidl et al., 2021; Widrich et al., 2020).

We use this mechanism in the form of a *Hopfield layer*, which first maps raw patterns to an associative space using linear transformations, and uses multiple simultaneous queries $\boldsymbol{\Xi} \in \mathbb{R}^{d \times N}$:

$$\mathrm{Hopfield}(\boldsymbol{\Xi}, \boldsymbol{C}) = (\boldsymbol{W}_E \boldsymbol{C})\ \mathrm{softmax}\left(\beta\ (\boldsymbol{W}_C \boldsymbol{C})^T\ (\boldsymbol{W}_\Xi \boldsymbol{\Xi})\right), \tag{7}$$

where $\boldsymbol{W}_E \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}_C, \boldsymbol{W}_\Xi \in \mathbb{R}^{e \times d}$ are trainable parameters of the Hopfield layer, $\mathrm{softmax}$ is applied column-wise, and β is a hyperparameter. Note that in principle the $\boldsymbol{\Xi}$ and $\boldsymbol{C}$ could have a different second dimension as long as the linear transformations map to the same dimension e. Note that all embeddings that enter this module are first layer normalized (Ba et al., 2016). Several of these Hopfield layers can run in parallel and we refer to them as "heads" in analogy to Transformers (Vaswani et al., 2017). From Eq. 7, Transformers arise as special cases of these layers (see below).

The context module of our new architecture uses a Hopfield layer, where the query patterns are the embeddings of the query molecule $\boldsymbol{m}$ and the support set molecules $\boldsymbol{X}$. The memory is composed of embeddings of a large set of M molecules from a chemical space, for example reference molecules, here called context molecules $\boldsymbol{C}$. Then the original embeddings $\boldsymbol{m}$ and $\boldsymbol{X}$ are replaced by the retrieved embeddings, which are weighted averages of context molecule embeddings:

$$\boldsymbol{m}' = \mathrm{Hopfield}(\boldsymbol{m}, \boldsymbol{C}) \quad \text{and} \quad \boldsymbol{X}' = \mathrm{Hopfield}(\boldsymbol{X}, \boldsymbol{C}). \tag{8}$$

Note that the embeddings of the query molecule and the support set molecules have not yet influenced each other. These updated representations $\boldsymbol{m}', \boldsymbol{X}'$ are passed to the cross-attention module.

3.3 Cross-attention module (CAM)

For embedding-based few-shot learning methods in the field of drug discovery, Altae-Tran et al. (2017) showed that the representations of the molecules can be enriched, if the architecture allows information exchange between query and support set molecules. Altae-Tran et al. (2017) uses an attention-enhanced LSTM variant which updates the query and the support set molecule representations in an iterative fashion, being aware of each other. We further develop this idea and combine it with the idea of using a transformer encoder layer (Vaswani et al., 2017) as a cross-attention module (Hou et al., 2019; Chen et al., 2021).

The cross-attention module updates the query molecule representation $\boldsymbol{m}'$ and the support set molecule representations $\boldsymbol{X}'$ by mutually exchanging information, using the usual Transformer mechanism:

$$[\boldsymbol{m}'', \boldsymbol{X}''] = \mathrm{Hopfield}([\boldsymbol{m}', \boldsymbol{X}'], [\boldsymbol{m}', \boldsymbol{X}']), \tag{9}$$

where $[\boldsymbol{m}', \boldsymbol{X}'] \in \mathbb{R}^{d \times (N+1)}$ is the concatenation of the representations of the query molecule $\boldsymbol{m}'$ with the support set molecules $\boldsymbol{X}'$ and we exploited that the Transformer is a special case of the Hopfield layer. Again, normalization is applied (Ba et al., 2016) and multiple Hopfield layers, i.e. heads, can run in parallel, can be stacked and equipped with skip-connections. The representations $\boldsymbol{m}''$ and $\boldsymbol{X}''$ are passed to the similarity module.

3.4 Similarity module (SM)

In this module, pairwise similarity values $k(\boldsymbol{m}'', \boldsymbol{x}_n'')$ are computed between the representation of the query molecule $\boldsymbol{m}''$ and each molecule $\boldsymbol{x}_n''$ in the support set as done recently (Koch et al., 2015; Altae-Tran et al., 2017). Based on these similarity values, the activity for the query molecule is predicted, building the weighted mean over the support set labels:

$$\hat{y} = \sigma\left(\tau^{-1} \frac{1}{N} \sum_{n=1}^{N} y_n' \; k(\boldsymbol{m}'', \boldsymbol{x}_n'')\right), \tag{10}$$

where our architecture employs dot product similarity of normalized representations $k(\boldsymbol{m}'', \boldsymbol{x}_n'') = {\boldsymbol{m}''}^T \boldsymbol{x}_n''$. $\sigma(.)$ is the sigmoid function and τ is a hyperparameter. Note that we use a balancing strategy for the labels $y_n' = \begin{cases} N/(2\sqrt{N_A}) & \text{if } y_n = 1 \\ -N/(2\sqrt{N_I}) & \text{else} \end{cases}$, where N_A is the number of actives and N_I is the number of inactives of the support set.

3.5 Architecture, hyperparameter selection, and training details

Hyperparameters. The main hyperparameters of our architecture are the number of heads, the embedding dimension and the dimension of the association space of the CAM and CM, the learning

rate schedule, the scaling parameter β and the molecule encoder. The following hyperparameters were selected by manual hyperparameter selection on the validation tasks. The molecule encoder consists of a single layer with output size $d = 1024$ and SELU activation (Klambauer et al., 2017). The CM consists of one Hopfield layer with 8 heads. The dimension e of the association space is set to 512 and $\beta = 1/\sqrt{e}$. As we use skip connections between all modules, the output dimension of the CM and CAM matches the input dimension. The CAM comprises one layer with 8 heads and an association-space dimension of 1088. For the input to the CAM, an activity encoding was added to the support set molecule representations to provide label information. The SM uses $\tau = 22.6$. For the context set, we randomly sample 5% from a large set of molecules, i.e. the molecules in the FS-Mol training split, for each batch. For inference, we used a fixed set of 5% of training set molecules as the context set for each seed. We hypothesize that these choices about the context could be further improved (see Section 5). For a table of considered and selected hyperparameters, see Supplementary.

Loss function, regularization and optimization. We use the Adam optimizer (Kingma and Ba, 2014) to minimize the cross-entropy loss between the predicted and known activity labels. We use a learning rate scheduler which includes a warm up phase, followed by a section with a constant learning rate, which is 0.0001, and a third phase in which the learning rate steadily decreases. As a regularization strategy, for the CM and the CAM a dropout rate of 0.5 is used. The molecule encoder has a dropout with rate 0.1 for the input and 0.5 elsewhere (see also Supplementary).

Compute time and resources. Training a single **MHNfs** model on the benchmarking dataset FS-Mol takes roughly 90 hours of wall-clock time on an A100 GPU. In total, roughly 15,000 GPU hours were consumed for this work.

4 Experiments

4.1 Benchmarking on FS-Mol

Experimental setup. Recently, the dataset FS-Mol (Stanley et al., 2021) was proposed to benchmark few-shot learning methods in drug discovery. It was extracted from ChEMBL27 and comprises in total 489,133 measurements, 233,786 compounds and 5,120 tasks. Per task, the mean number of data points is 94. The dataset is well balanced as the mean ratio of active and inactive molecules is close to 1. The FS-Mol benchmark dataset defines 4,938 training, 40 validation and 157 test tasks, guaranteeing disjoint task sets. Stanley et al. (2021) precomputed extended connectivity fingerprints (ECFP) (Rogers and Hahn, 2010) and key molecular physical descriptors, which were defined by RDKit (Landrum et al., 2006). While methods would be allowed to use other representations of the input molecules, such as the molecular graph, we use these provided descriptors as inputs for our methods. For the main benchmark, the support set size was fixed to 16, using a stratified random split. We use all these settings of FS-Mol and therefore ensure a fair method comparison.

Methods compared. Baselines for few-shot learning and our proposed method **MHNfs** were compared against each other. The **Frequent Hitters** model is a naive baseline that ignores the provided support set and therefore has to learn to predict the average activity of a molecule. This method can potentially discriminate so-called frequent-hitter molecules (Stork et al., 2019) against molecules that are inactive across many tasks. We added **Similarity Search** (Cereto-Massagué et al., 2015) as a baseline, which is a standard chemoinformatics technique used in situations in which a single or few actives are known. In the simplest case, molecules that are similar to given active molecules are searched by computing a fingerprint or descriptor-representation of the molecules and using a similarity measure $k(.,.)$, such as Tanimoto Similarity(Tanimoto, 1960). Thus, the Similarity Search as used in chemoinformatics can be formally written as $\hat{y} = 1/N \sum_{n=1}^{N} y_n \; k(\boldsymbol{m}, \boldsymbol{x}_n)$, where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\mathrm{n}$ come from a fixed molecule encoder, such as chemical fingerprint or descriptor calculation. A natural extension of Similarity Search with fixed chemical descriptors is **Neural Similarity Search or Siamese networks** (Koch et al., 2015) wich extend the classic similarity search by learning a molecule encoder: $\hat{y} = \sigma\left(\tau^{-1} \frac{1}{N} \sum_{n=1}^{N} y'_n \; f_{\boldsymbol{w}}^{\mathrm{ME}}(\boldsymbol{m})^T \; f_{\boldsymbol{w}}^{\mathrm{ME}}(\boldsymbol{x}_n)\right)$. Furthermore, we re-implemented the **IterRefLSTM** (Altae-Tran et al., 2017) in Pytorch. The **IterRefLSTM** model consists of three modules. First, a molecule encoder maps the query and support set molecules to its representations $\boldsymbol{m}$ and $\boldsymbol{X}$. Second, an attention-enhanced LSTM variant, the IterRefLSTM, iteratively updates the query and support set molecules, enabling information sharing between the

Table 1: Results on FS-MOL [ΔAUC-PR]. The best method is marked bold. Error bars represent standard errors across tasks according to Stanley et al. (2021). The metrics are also averaged across five training re-runs and ten draws of support sets. In brackets the number of tasks per category is reported.

Method	All [157]	Kin. [125]	Hydrol. [20]	Oxid.[7]
GNN-ST[a] (Stanley et al., 2021)	$.029 \pm .004$	$.027 \pm .004$	$.040 \pm .018$	$.020 \pm .016$
MAT[a] (Maziarka et al., 2020)	$.052 \pm .005$	$.043 \pm .005$	$.095 \pm .019$	$.062 \pm .024$
Random Forest[a] (Breiman, 2001)	$.092 \pm .007$	$.081 \pm .009$	$.158 \pm .028$	$.080 \pm .029$
GNN-MT[a] (Stanley et al., 2021)	$.093 \pm .006$	$.093 \pm .006$	$.108 \pm .025$	$.053 \pm .018$
Similarity Search	$.118 \pm .008$	$.109 \pm .008$	$.166 \pm .029$	$.097 \pm .033$
GNN-MAML[a] (Finn et al., 2017)	$.159 \pm .009$	$.177 \pm .009$	$.105 \pm .024$	$.054 \pm .028$
Frequent hitters	$.182 \pm .010$	$.207 \pm .009$	$.098 \pm .009$	$.041 \pm .005$
ProtoNet[a] (Snell et al., 2017)	$.207 \pm .008$	$.215 \pm .009$	$.209 \pm .030$	$.095 \pm .029$
Siamese Networks (Koch et al., 2015)	$.223 \pm .010$	$.241 \pm .010$	$.178 \pm .026$	$.082 \pm .025$
IterRefLSTM (Altae-Tran et al., 2017)	$.233 \pm .009$	$.249 \pm .010$	$.204 \pm .026$	$.091 \pm .028$
MHNfs (ours)	$\mathbf{.241} \pm .009$	$.259 \pm .010$	$.201 \pm .027$	$.087 \pm .019$

[a] metrics from Stanley et al. (2021).

molecules: $[\boldsymbol{m}', \boldsymbol{X}'] = \mathrm{IterRefLSTM}_L([\boldsymbol{m}, \boldsymbol{X}])$, where the hyperparameter L controls the number of iteration steps of the IterRefLSTM. Third, a similarity module computes attention weights based on the representations: $\boldsymbol{a} = \mathrm{softmax}\left(k\left(\boldsymbol{m}', \boldsymbol{X}'\right)\right)$. These representations are then used for the final prediction: $\hat{y} = \sum_{i=1}^{N} a_i y_i$. For further details, see Supplementary. The **Random Forest** baseline uses the chemical descriptors and is trained in standard supervised manner on the support set molecules for each task. The method **GNN-ST** is a graph neural network (Stanley et al., 2021; Gilmer et al., 2017) that is trained from scratch for each task. The **GNN-MT** uses a two step strategy, which is first to pre-train the model on a large dataset on related task and then adapting an output layer to the few-shot task via linear probing (Stanley et al., 2021; Alain and Bengio, 2016). The **Molecule Attention Transformer (MAT)** is pre-trained in a self-supervised fashion and fine-tuning is performed for the few-shot task (Maziarka et al., 2020). **GNN-MAML** is based on MAML (Finn et al., 2017), and uses a model-agnostic meta-learning strategy to find a general core model from which one can easily adapt to single tasks. **ProtoNet** (Snell et al., 2017) includes a molecule encoder, which maps query and support set molecules to representations in an embedding space. In this embedding space, prototypical representations of each class are built by taking the mean across all related support support set molecules for each class (details in Supplementary). For all methods the most important hyperparameters were adjusted on the validation tasks of FS-Mol.

Training and evaluation. For the model implementations, we used PyTorch (Paszke et al., 2019, BSD license), making use of PyTorch Lightning (Falcon et al., 2019, Apache 2.0 license) as a framework for training and test logic, hydra for config file handling (Yadan, 2019, Apache 2.0 license) and Weights & Biases (Biewald, 2020, MIT license) as an experiment tracking tool. We performed five training reruns with different seeds for all methods, except Classic Similarity Search as there is no variability across seeds. Each trained model was evaluated ten times by drawing support sets with ten different seeds.

Results. The results in terms of area-under-precision-recall curve (AUC-PR) are presented in Table 1, where the difference to a random classifier is reported (ΔAUC-PR). The standard error is reported across tasks. Surprisingly, the naive baseline **Frequent Hitters**, that neglects the support set, has outperformed most of the few-shot learning methods, except for the embedding-based methods **Siamese Networks**, **ProtoNet**, **IterRefLSTM**, and **MHNfs**. **IterRefLSTM**, which has not been included in the FS-Mol benchmark study, reaches the second best performance. **MHNfs** has outperformed all other methods with respect to ΔAUC-PR across all tasks, including the **IterRefLSTM** model (p-value 0.0014, paired Wilcoxon test).

4.2 Ablation study

MHNfs has two new main elements compared to the previous state-of-the-art method **IterRefLSTM**: i) the context module, and ii) the cross-attention module, which replaces the LSTM-like module. To assess the effects of these elements, we performed an ablation study. Therefore, we compared

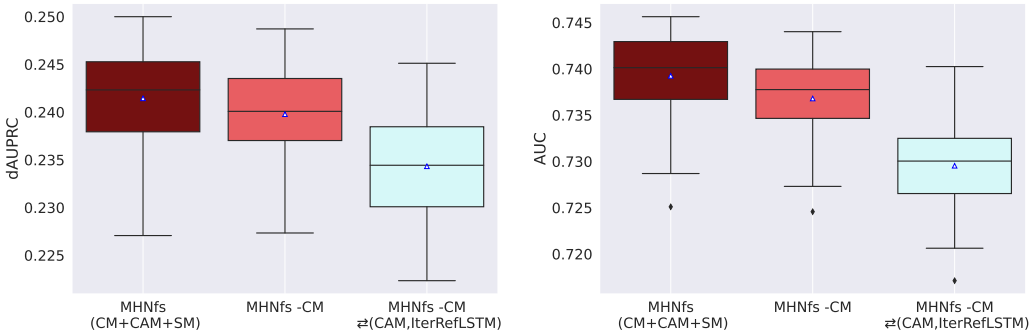

Figure 2: Results of the ablation study. The boxes show the median, mean and the variability of the average predictive performance of the methods across training re-runs and draws of support sets. The performance significantly drops when the context module is removed (light red bars), and when additionally the cross-attention module is replaced with the IterRefLSTM module (light blue bars). This indicates that our two newly introduced modules, CM and CAM, play a crucial role in MHNfs.

MHNfs to a method that does not have the context module ("MHNfs -CM") and to a method that does not have the context module and an LSTM-like module instead of the CAM ("MHNfs -CM ⇌(CAM,IterRefLSTM)"). For the ablation study, we used all 5 training reruns and evaluated each model 10 times on the test set with different support sets. The results of this ablation steps are presented in Figure 2. Both removing the CM and exchanging the CAM with the IterRefLSTM module were detrimental for the performance of the method (p-value 0.002 and 1.3e-8, respectively; paired Wilcoxon test). The difference was even more pronounced under domain shift (see Supplementary). A second ablation study to examine the overall effects of the context module, the cross-attention module, the similarity module, and the molecule encoder in the **MHNfs** model is reported in the Supplementary.

4.3 Domain shift experiment

Experimental setup. For the domain shift experiment, we used the Tox21 dataset. This dataset consists of 12,707 chemical compounds, for which measurements for up to 12 different toxic effects are reported (Mayr et al., 2016; Huang et al., 2016a). It was published with a fixed training, validation and test split. State-of-the-art supervised learning methods that have access to the full training set reach AUC performance values between 0.845 and 0.871 (Klambauer et al., 2017; Duvenaud et al., 2015; Li et al., 2017, 2021; Zaslavskiy et al., 2019; Alperstein et al., 2019). We re-cast Tox21 as a few-shot learning setting and draw small support sets from the 12 tasks. The compared methods were pre-trained on FS-Mol and then obtain small support sets from Tox21. Based on those support sets, the methods had to predict the activities of the Tox21 test set. Note that there is a strong domain shift from drug-like molecules of FS-Mol to environmental chemicals, pesticides, food additives of Tox21. The domain shift also concerns the outputs where a shift from kinases, hydrolases, and oxidoreductases of FS-Mol to nuclear receptors and stress responses of Tox21 is present.

Methods compared. The new method **MHNfs**, the runner-up method **IterRefLSTM**, and **Similarity Search**, since it has been widely used for such purposes for decades (Cereto-Massagué et al., 2015), were compared.

Training and evaluation. We followed the procedure of Stanley et al. (2021) for data-cleaning, preprocessing and extraction of the fingerprints and descriptors used in FS-Mol. After running the cleanup step, 8,423 molecules remained for the domain shift experiments. From the training set, 8 active and 8 inactive molecules per task were randomly selected to build the support set. The test set molecules were used as query molecules. The validation set molecules were not used at all. During test-time, a support set was drawn ten times for each task. Then, the performance of the model was evaluated for these support sets, using the area-under-precision-recall curve (AUC-PR), analogously to the FS-Mol benchmarking experiment reported as the difference to a random classifier (ΔAUC-PR), and the area-under-receiver-operating-characteristic-curve (AUC) metrics. The performance values report the mean over all combinations regarding the training reruns and the support set sampling iterations. Error bars indicate the standard deviation.

Table 2: Results of the domain shift experiment on Tox21 [AUC, ΔAUC-PR]. The best method is marked bold. Error bars represent standard deviation across training re-runs and draws of support sets

Method	AUC	ΔAUC-PR
Similarity Search (baseline)	.629 ± .015	.061 ± .008
IterRefLSTM (Altae-Tran et al., 2017)	.664 ± .018	.068 ± .009
MHNfs (ours)	**.679** ± .018	**.073** ± .008

Results. The Hopfield-based context retrieval method has significantly outperformed the IterRefLSTM-based model ($p_{\Delta\mathrm{AUPRC}}$-value 0.0002, p_{AUC}-value 9.4e-5, paired Wilcoxon test) and the Classic Similarity Search ($p_{\Delta\mathrm{AUPRC}}$-value 2.4e-9, p_{AUC}-value 7.6e-10, paired Wilcoxon test) and therefore showed robust performance on the toxicity domain, see Table 2. Notably, for none of these methods, any weight parameter was adjusted to the new dataset.

5 Conclusion and discussion

We have introduced a new architecture for few shot learning in drug discovery that is based on the novel concept to enrich molecule representations with context. The architecture, called **MHNfs**, comprises an adaptive molecule encoder, a context module, a cross-attention module and a similarity module. In a benchmarking experiment, the architecture was assessed for its ability to learn accurate predictive models from small sets of labelled molecules. Indeed, **MHNfs** outperformed all previously suggested methods on FS-Mol. In a domain shift study, the robustness and transferability of the learned models has been assessed and again **MHNfs** exhibited the best performance. It has not escaped our notice that the specific context module we have proposed could immediately be used for few-shot learning tasks in computer vision, but might be hampered by computational constraints.

Limitations. Our method is currently limited to small, organic drug-like molecules as inputs. Thus, the model cannot be used for molecules such as RNA, DNA or proteins. The output domain comprises biological effects focusing on effects on drug targets, such that the output of the trained models must be understood in that domain. While we hypothesize that the approach could also be successful for similar data in the materials science domain, this has not been assessed. Our study is also constrained by a limited amount of hyperparameter search for all methods. Deep Learning methods usually have a large number of hyperparameters, such as hidden dimensions, number of layers, learning rates, of which we were only able to explore the most important ones. The composition and choice of the context set is also underexplored and might be improved by selecting reference molecules with an appropriate strategy. Further considerations can be found in the section below.

Broader impact. *Impact on machine learning and related scientific fields.* We envision that with (a) the increasing availability of drug discovery and material science dataset, (b) further improved biotechnologies, and (c) accounting for characteristics of individuals, the drug and materials discovery process will be made more efficient. For machine learning and artificial intelligence, the novel way in which representations are enriched with context might strengthen the general research stream to include more context into Deep Learning systems. Our approach also shows that such a system is more robust against domain shifts, which could be a step towards Broad AI (Chollet, 2019; Hochreiter, 2022). *Impact on society.* If the approach proves itself useful, it could lead to a faster and more cost-efficient drug discovery process. Especially the COVID-19 pandemic has shown that it is crucial for humanity to speed up the drug discovery process to few years or even months. We hope that this work contributes to this effort and eventually leads to safer drugs developed faster. *Consequences of failures of the method.* As common with methods in machine learning, potential danger lies in the possibility that users rely too much on our new approach and use it without reflecting on the outcomes. Failures of the proposed method would lead to unsuccessful wet lab validation and negative wet lab tests. Since the proposed algorithm does not directly suggest treatment or therapy, human beings are not directly at risk of being treated with a harmful therapy. Wet lab and in-vitro testing would indicate wrong decisions by the system. *Leveraging of biases in the data and potential discrimination.* As for almost all machine learning methods, confounding factors, lab or batch effects, could be used for classification. This, might lead to biases in predictions or uneven predictive performance across different drug targets or bioassays.

References

Adler, T., Brandstetter, J., Widrich, M., Mayr, A., Kreil, D., Kopp, M., Klambauer, G., and Hochreiter, S. (2020). Cross-domain few-shot learning by representation fusion. *arXiv preprint arXiv:2010.06498*.

Alain, G. and Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Alperstein, Z., Cherkasov, A., and Rolfe, J. T. (2019). All smiles variational autoencoder. *arXiv preprint arXiv:1905.13343*.

Altae-Tran, H., Ramsundar, B., Pappu, A. S., and Pande, V. (2017). Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293.

Antoniou, A. and Storkey, A. (2019). Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. *arXiv preprint arXiv:1902.09884*.

Arrowsmith, J. (2011). Phase ii failures: 2008-2010. *Nature reviews drug discovery*, 10(5).

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bender, A., Mussa, H. Y., Glen, R. C., and Reiling, S. (2004). Similarity searching of chemical databases using atom environment descriptors (molprint 2d): evaluation of performance. *Journal of chemical information and computer sciences*, 44(5):1708–1718.

Bendre, N., Marín, H. T., and Najafirad, P. (2020). Learning from few samples: A survey. *arXiv preprint arXiv:2007.15484*.

Bengio, Y., Bengio, S., and Cloutier, J. (1991). Learning a synaptic learning rule. In *Seattle international joint conference on neural networks*.

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

Bonner, M. F. and Epstein, R. A. (2021). Object representations in the human brain reflect the co-occurrence statistics of vision and language. *Nat Commun*, 12(4081).

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Cereto-Massagué, A., Ojeda, M. J., Valls, C., Mulero, M., Garcia-Vallvé, S., and Pujadas, G. (2015). Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63.

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., and Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250.

Chen, H., Li, H., Li, Y., and Chen, C. (2021). Sparse spatial transformers for few-shot learning. *arXiv preprint arXiv:2109.12932*.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chollet, F. (2019). On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.

Dahl, G. E., Jaitly, N., and Salakhutdinov, R. (2014). Multi-task neural networks for qsar predictions. *arXiv preprint arXiv:1406.1231*.

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*.

Falcon, W. et al. (2019). Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3:6.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Fürst, A., Rumetshofer, E., Tran, V., Ramsauer, H., Tang, F., Lehner, J., Kreil, D., Kopp, M., Klambauer, G., Bitto-Nemling, A., et al. (2021). Cloob: Modern hopfield networks with infoloob outperform clip. *arXiv preprint arXiv:2110.11316*.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

Gomez, L. (2018). Decision making in medicinal chemistry: The power of our intuition. *ACS Medicinal Chemistry Letters*, 9(10):956–958.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276.

Guo, Z., Zhang, C., Yu, W., Herr, J., Wiest, O., Jiang, M., and Chawla, N. V. (2021). Few-shot graph learning for molecular property prediction. In *Proceedings of the web conference 2021*, pages 2559–2567.

He, J., You, H., Sandström, E., Nittinger, E., Bjerrum, E. J., Tyrchan, C., Czechtizky, W., and Engkvist, O. (2021). Molecular optimization by capturing chemist's intuition using deep neural networks. *Journal of cheminformatics*, 13(1):1–17.

Hochreiter, S. (2022). Toward a broad ai. *Communications of the ACM*, 65(4):56–57.

Hochreiter, S., Klambauer, G., and Rarey, M. (2018). Machine learning in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1723–1724.

Hochreiter, S., Younger, A. S., and Conwell, P. R. (2001). Learning to learn using gradient descent. In *International conference on artificial neural networks*, pages 87–94. Springer.

Hou, R., Chang, H., Ma, B., Shan, S., and Chen, X. (2019). Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems*, 32.

Huang, R., Xia, M., Nguyen, D.-T., Zhao, T., Sakamuru, S., Zhao, J., Shahane, S. A., Rossoshek, A., and Simeonov, A. (2016a). Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. *Frontiers in Environmental Science*, 3:85.

Huang, R., Xia, M., Sakamuru, S., Zhao, J., Shahane, S. A., Attene-Ramos, M., Zhao, T., Austin, C. P., and Simeonov, A. (2016b). Modelling the tox21 10 k chemical profiles for in vivo toxicity prediction and mechanism characterization. *Nature communications*, 7(1):1–10.

Jiang, D., Wu, Z., Hsieh, C.-Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J., and Hou, T. (2021). Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13(1):1–23.

Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in neural information processing systems 30*, pages 972–981.

Koch, G., Zemel, R., Salakhutdinov, R., et al. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

Landrum, G. et al. (2006). Rdkit: Open-source cheminformatics.

Li, J., Cai, D., and He, X. (2017). Learning graph-level representation for drug discovery. *arXiv preprint arXiv:1709.03741*.

Li, P., Li, Y., Hsieh, C.-Y., Zhang, S., Liu, X., Liu, H., Song, S., and Yao, X. (2021). Trimnet: learning molecular representation from triplet messages for biomedicine. *Briefings in Bioinformatics*, 22(4):bbaa266.

Mayr, A., Klambauer, G., Unterthiner, T., and Hochreiter, S. (2016). Deeptox: toxicity prediction using deep learning. *Frontiers in environmental science*, 3:80.

Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., Wegner, J. K., Ceulemans, H., Clevert, D.-A., and Hochreiter, S. (2018). Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24):5441–5451.

Maziarka, Ł., Danel, T., Mucha, S., Rataj, K., Tabor, J., and Jastrzębski, S. (2020). Molecule attention transformer. *arXiv preprint arXiv:2002.08264*.

Merk, D., Friedrich, L., Grisoni, F., and Schneider, G. (2018). De novo design of bioactive small molecules by artificial intelligence. *Molecular informatics*, 37(1-2):1700153.

Merkwirth, C. and Lengauer, T. (2005). Automatic generation of complementary descriptors with molecular graph networks. *Journal of chemical information and modeling*, 45(5):1159–1168.

Miller, E. G., Matsakis, N. E., and Viola, P. A. (2000). Learning from one example through shared densities on transforms. In *Proceedings ieee conference on computer vision and pattern recognition. cvpr 2000 (cat. no. PR00662)*, volume 1, pages 464–471. IEEE.

Nguyen, C. Q., Kreatsoulas, C., and Branson, K. M. (2020). Meta-learning gnn initializations for low-resource molecular property prediction. *arXiv preprint arXiv:2003.05996*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2019). Automatic differentiation in pytorch. In *Conference on neural information processing systems*.

Potter, M. (2012). Conceptual short term memory in perception and thought. *Frontiers in Psychology*, 3:113.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., et al. (2020). Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*.

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754.

Schmidhuber, J. (1987). *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München.

Schneider, P., Walters, W. P., Plowright, A. T., Sieroka, N., Listgarten, J., Goodnow, R. A., Fisher, J., Jansen, J. M., Duca, J. S., Rush, T. S., et al. (2020). Rethinking drug design in the artificial intelligence era. *Nature reviews drug discovery*, 19(5):353–364.

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. (2018a). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1):120–131.

Segler, M. H., Preuss, M., and Waller, M. P. (2018b). Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610.

Seidl, P., Renz, P., Dyubankova, N., Neves, P., Verhoeven, J., Wegner, J. K., Hochreiter, S., and Klambauer, G. (2021). Modern hopfield networks for few-and zero-shot reaction prediction. *arXiv preprint arXiv:2104.03279*.

Snell, J., Swersky, K., and Zemel, R. S. (2017). Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*.

Stanley, M., Bronskill, J. F., Maziarz, K., Misztela, H., Lanini, J., Segler, M., Schneider, N., and Brockschmidt, M. (2021). Fs-mol: A few-shot learning dataset of molecules. In *Conference on neural information processing systems workshop*.

Stork, C., Chen, Y., Sicho, M., and Kirchmair, J. (2019). Hit dexter 2.0: machine-learning models for the prediction of frequent hitters. *Journal of chemical information and modeling*, 59(3):1030–1043.

Sturm, N., Mayr, A., Le Van, T., Chupakhin, V., Ceulemans, H., Wegner, J., Golib-Dzib, J.-F., Jeliazkova, N., Vandriessche, Y., Böhm, S., et al. (2020). Industry-scale application and evaluation of deep learning for drug target prediction. *Journal of Cheminformatics*, 12(1):1–13.

Sun, J., Jeliazkova, N., Chupakhin, V., Golib-Dzib, J.-F., Engkvist, O., Carlsson, L., Wegner, J., Ceulemans, H., Georgiev, I., Jeliazkov, V., et al. (2017). Excape-db: an integrated large scale dataset facilitating big data analysis in chemogenomics. *Journal of cheminformatics*, 9(1):1–9.

Tanimoto, T. (1960). Ibm type 704 medical diagnosis program. *IRE transactions on medical electronics*, (4):280–283.

Unterthiner, T., Mayr, A., Klambauer, G., Steijaert, M., Wegner, J. K., Ceulemans, H., and Hochreiter, S. (2014). Multi-task deep networks for drug target prediction. In *Neural information processing system workshop*, volume 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638.

Walters, W. P. and Barzilay, R. (2021). Critical assessment of ai in drug discovery. *Expert opinion on drug discovery*, pages 1–11.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.

Waring, M. J., Arrowsmith, J., Leach, A. R., Leeson, P. D., Mandrell, S., Owen, R. M., Pairaudeau, G., Pennie, W. D., Pickett, S. D., Wang, J., et al. (2015). An analysis of the attrition of drug candidates from four major pharmaceutical companies. *Nature reviews drug discovery*, 14(7):475–486.

Weininger, D. (1988). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36.

Widrich, M., Schäfl, B., Ramsauer, H., Pavlović, M., Gruber, L., Holzleitner, M., Brandstetter, J., Sandve, G. K., Greiff, V., Hochreiter, S., et al. (2020). Modern hopfield networks and attention for immune repertoire classification. *arXiv preprint arXiv:2007.13505*.

Winter, R., Montanari, F., Noé, F., and Clevert, D.-A. (2019). Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical science*, 10(6):1692–1701.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. (2018). Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530.

Yadan, O. (2019). Hydra - a framework for elegantly configuring complex applications. Github. Visited 2022-04-25.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. (2019). Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388.

Zaslavskiy, M., Jégou, S., Tramel, E. W., and Wainrib, G. (2019). Toxicblend: Virtual screening of toxic compounds with ensemble predictors. *Computational Toxicology*, 10:81–88.

Zhao, A., Balakrishnan, G., Durand, F., Guttag, J. V., and Dalca, A. V. (2019). Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 8543–8553.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We refer e.g. to Section 4 in which we provide this information. All training details are provided in Supplementary Material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report error bars for all performance metrics. For all experiments the variability across re-runs, different support sets and prediction tasks was assessed.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.5.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Section 5.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]