FS-Mol: A Few-Shot Learning Dataset of Molecules

Megan Stanley
Microsoft Research

John Bronskill
Microsoft Research
University of Cambridge

Krzysztof Maziarz
Microsoft Research

Hubert Misztela
Novartis

Jessica Lanini
Novartis

Marwin Segler
Microsoft Research

Nadine Schneider
Novartis

Marc Brockschmidt
Microsoft Research

Abstract

Small datasets are ubiquitous in drug discovery as data generation is expensive and can be restricted for ethical reasons (e.g. in vivo experiments). A widely applied technique in early drug discovery to identify novel active molecules against a protein target is modeling quantitative structure-activity relationships (QSAR). It is known to be extremely challenging, as available measurements of compound activities range in the low dozens or hundreds. However, many such related datasets exist, each with a small number of datapoints, opening up the opportunity for few-shot learning after pretraining on a substantially larger corpus of data. At the same time, many few-shot learning methods are currently evaluated in the computer-vision domain. We propose that expansion into a new application, as well as the possibility to use explicitly graph-structured data, will drive exciting progress in few-shot learning. Here, we provide a few-shot learning dataset (FS-Mol) and complementary benchmarking procedure. We define a set of tasks on which few-shot learning methods can be evaluated, with a separate set of tasks for use in pretraining. In addition, we implement and evaluate a number of existing single-task, multi-task, and meta-learning approaches as baselines for the community. We hope that our dataset, support code release, and baselines will encourage future work on this extremely challenging new domain for few-shot learning.

1 Introduction

Deep Learning has led to tremendous progress in a variety of domains, such as computer vision [6], natural language processing [11], molecular design [49], chemical synthesis planning [45], and most recently protein folding [26]. In all of these cases, progress has been driven by large single-task datasets (e.g., 14M images in ImageNet; $\sim$500B tokens for GPT-3), often enabled by a combination of substantial manual labeling together with elegant methods able to learn from unlabeled data.

Computer-aided drug discovery has long made use of machine learning to predict the biological activity of molecules [33, 52, 56]. Such models are often most useful in the hit-to-lead and early lead optimization phase of a drug discovery project. After the biological target (often a protein) has been identified, initial hit molecules, which modulate the target but do not have yet have all required properties, need to be optimized towards a drug candidate suitable for clinical testing. This optimization phase proceeds by the time-consuming and expensive iterative procedure of designing new molecules, synthesizing them, and measuring their properties. Prediction using ML models reduces the number of molecules requiring synthesis and wet-lab measurement, considerably reducing the cost of drug discovery [44]. In the typical life cycle of a drug discovery project, only a few hundred molecules can be made and tested. With such small datasets, current deep learning approaches are in practice often not more effective than support vector machines, random forests or gradient-boosted tree models based on molecular feature representations, which have been tuned over decades [59].

35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks.

However, data is gathered across many different targets (i.e. tasks) and different drug discovery projects, suggesting the possibility of knowledge transfer between low-data tasks. A common framework for such settings is few-shot learning [10, 18, 41, 53, 58, 64]. For example, meta-learning [18, 21, 53, 58] refers to algorithms which learn to learn; given an unseen task, such methods utilize their prior knowledge from previous training episodes to generalize and make predictions in a few-shot manner. The efficacy of this paradigm for enhancing molecular property prediction in low-data regimes has shown some promise [7, 9, 35, 36], but we note that there is no dataset specifically designed to benchmark few-shot learning methods for molecules and thus allow easy model comparison. In particular, this would require a dataset that provides both a large range of diverse tasks on which to perform an initial pretraining step as well as a set of low-data tasks on which to evaluate, with both of these chosen to be similar in nature.

From a machine learning point of view, few-shot learning has been primarily focused on the computer vision domain [29, 55, 58]. However, state-of-the-art performance has become relatively saturated [12, 24, 37, 54], and many approaches profit from pretraining on rich, highly related large datasets such as ImageNet [43], which are not available in other few-shot domains. To drive further algorithmic innovation and solutions to real-world scientific problems, we propose that few-shot learning should expand into alternate domains of applicability.

With our new dataset, we aim to inspire the development of machine learning methods that generalize well across a diverse distribution of tasks and adapt efficiently to minimal new data. We also consider it essential that models developed using such a dataset and its benchmarks should have value in real-world applications, and hence we propose to use Quantitative Structure-Activity Relationships (QSAR) as the data domain, in which the task is to predict the activity in inhibition or activation of a specific target protein given the structure of a molecule.

In this work, we make three key contributions:

- A Few-Shot Learning Dataset of Molecules (henceforth abbreviated to FS-Mol) that can demonstrate the utility of few-shot learning methods in an important domain, namely QSAR, which does not provide an obvious generic pretraining corpus (such as in NLP or computer vision). The proposed dataset is specifically designed to replicate the challenges of machine learning in the very low data regime of drug-discovery projects.
- A fixed benchmarking procedure on this dataset that allows to easily compare new few-shot learning methods in an apples-to-apples scenario. We hope this will encourage the development of novel methods for few-shot learning on structured data.
- The establishment of baselines against the new benchmark with representative few-shot learning methods.

2 Motivation

Our main goal is to provide a dataset, benchmarks and baselines to encourage development of machine learning methods for realistic drug discovery tasks. We here focus on the challenge of a novel QSAR task at the very early stages of a lead optimization project, where there there is only very little available data. This will enable the community to explore the effectiveness of different machine learning techniques in this domain, and develop new methods to meet the unique challenges of the data. Concretely, we are interested in helping to answer two core research questions:

(RQ1) Does knowledge learned from a large dataset transfer gainfully to previously unseen tasks?

(RQ2) How does the performance vary with the size of the training data available in a new task?

2.1 Background: Few-Shot Learning

Standard supervised single-task methods rely on one dataset $\mathcal{D}$ to train a model, and evaluation proceeds by examining the predictions made on a held-out test set. Such approaches are known to work well with large sets of training examples drawn from the same distribution as the test set. In the few-shot learning scenario, no large training set exists. Instead, we aim to leverage an advantageous initialization (for example obtained from *pretraining* on related tasks), followed by training (*adaptation*) on very few datapoints from a previously unseen task.

The methods used during pretraining can fall into one of three categories. *Self-supervised* methods make use of a very large unlabelled dataset on which proxy tasks such as prediction of masked parts of the input are defined [16]. *Multitask* [13] and *meta-learning* methods assume the availability of a set of related supervised training tasks $\mathcal{D}_{train} = \{\mathcal{T}_t\}_{t=1}^{K}$. In general, each task $\mathcal{T}_t$ is composed of a *support set* $\mathcal{T}_{t,support}$, and a *query set* $\mathcal{T}_{t,query}$. The support set consists of examples with features $\mathbf{x}_i^t$ and labels y_i^t which can be used to train a model, and the model is given the features $\mathbf{x}_j^t$, to predict the label y_j^t of the query set.[1] At what we will term *few-shot testing* time, a model is given access to an unseen new task $\mathcal{T}_u$. It is then expected to predict labels on $\mathcal{T}_{u,query}$, while having access to the features and labels of $\mathcal{T}_{u,support}$. Consequently, evaluating these methods requires two disjoint sets of tasks: $\mathcal{D}_{train}$ (used for pretraining) and $\mathcal{D}_{test}$ (used to evaluate pre-trained models).

2.2 Desired Attributes of a QSAR Few-Shot Dataset and Benchmark

Given our research questions and the structure of most few-shot learning techniques, we can hence derive design requirements for our new dataset.

Concretely, to be amenable to few-shot learning, it should provide a large set $\mathcal{D}_{train}$ of training tasks useful for pretraining, and a disjoint set $\mathcal{D}_{test}$ of test tasks that are related to the training tasks. To enable an analysis of the extent of generalization to new tasks, $\mathcal{D}_{test}$ should contain both tasks that are very similar to the training data, as well as others that require more specialization at few-shot testing time. Finally, our test tasks should be chosen such that we can evaluate the amenability of evaluated methods to different support set sizes.

At the same time, we aim to construct a benchmark that is relevant for real-world drug-discovery projects. To this end, the number of samples for some tasks should be small, reflecting early-stage projects with access to measurements for fewer than 100 compounds. The considered molecules should be drug-like, and the tasks themselves should include a broad range of drug targets. Finally, the labels should be drawn from real measurements, to reflect the noise observed in wet-lab measurements for a novel target.

Overall, a benchmark on this dataset should capture the efficacy of the few-shot learning method in generalization to new tasks, i.e., does the adaptation method perform better than a single-task method exposed to the set $\mathcal{T}_{u,support}$ alone (answering **(RQ1)**), and can this adaptation use a minimal amount of data, answering **(RQ2)**.

3 The Few-Shot Learning Dataset of Molecules

We construct our dataset as a careful selection of data from ChEMBL27 [3]. We provide an overview in this section, but refer to our source code release (and in particular `ExtractDataset.ipynb`) at `https://github.com/microsoft/FS-Mol/` for an exact, executable and reproducible definition.

Selection of molecular property prediction tasks ChEMBL contains the results of many experiments, termed "assays", each having a unique experiment ID. We retained only those measurements referring to small molecule activity (IC50 or EC50) [48], and removed all compounds with a molecular weight ≥ 900 Dalton in order to ensure only drug-like molecules are included. We then applied a standard cleaning and canonicalization procedure to all compounds (see details in our code release) and stored them as SMILES strings [61]. Assays were then selected to have at least 32 datapoints and not more than 5000 datapoints. The reason why we remove large assays is that they often come from high-throughput screens (HTS), and thus contain a high percentage of inactive compounds, and are very noisy, rendering the challenge more complex. We further exclude all assays that are not associated with a specific target protein ID. We view each selected, filtered assay as a single task in our few-shot learning dataset, as described in section 2.1. We only consider assays with a single protein target (where the same target may be the subject of several separate assays), and treat assays as separate tasks to avoid inter-assay noise often seen when combining measurements [27].

Split into pretraining and test tasks As part of our dataset release, we provide a split into pretraining tasks $\mathcal{D}_{train}$ and (few-shot) test tasks $\mathcal{D}_{test}$. In order to derive disjoint task sets, we require that all selected assays are associated with a specific protein target. We avoid an overlap

[1] In the single-task scenario, the "support" set is called the training data, and the "query" set is the test data.

of very related tasks between pretraining and the test set by splitting tasks such that protein targets are used either only in the training or at most once in the test data.[2] We identified few-shot testing tasks from the subset of tasks that address enzyme targets. This choice enabled us to partition the set $\mathcal{D}_{test}$ by EC (Enzyme Commission) number [2] to permit sub-benchmarks within the overall benchmark set. The best few-shot learners are those able to perform well across all sub-benchmarks. Few-shot testing tasks were required to contain > 128 compounds to allow comparison of model adaptation performance across a range of available support set sizes, necessary to answer **(RQ2)**. Our final $\mathcal{D}_{test}$ comprises 157 tasks, while $\mathcal{D}_{train}$ has 4938 tasks. We additionally also provide a $\mathcal{D}_{valid}$ set, consisting of 40 tasks selected in the same manner as $\mathcal{D}_{test}$, to aid the development of few-shot learning methods. In this way, our proposed meta-testing tasks closely mimic the new-lead optimization problem, where a completely unseen task is presented for adaptation.

We encourage the use of the set $\mathcal{D}_{train}$ not only for pretraining as described in section 2.1, but also as a task-specific phase following other large-data pretraining methods [23]. To enable this, the code implementing the dataset extraction and cleaning protocol outlined above can easily be adapted to select a larger set of assays than our chosen $\mathcal{D}_{train}$ (for example by also considering assays that are not annotated with a protein target).

Binary Classification Task While the raw ChEMBL data provides activity as a floating point number, treating this as a regression target is known to be extremely hard (for reasons including measurement noise, narrow measurement range and that low/high values are often only encoded as a boundary constant). Instead, many practitioners only consider a binary classification task into active/inactive compounds, which is substantially more robust and often good enough to make decisions in the drug discovery process. While in practice thresholds for this would be defined on a by-project basis, we opt for an automated thresholding procedure based on the IC_{50} or EC_{50} value available for each compound. The median value over compounds in an assay defines the threshold, but the range of allowed thresholds are fixed to $5 \leq \text{pXC} \leq 7$ for enzyme targets and $4 \leq \text{pXC} \leq 6$ for all other protein targets, where $\text{pXC} = -\log_{10}(\text{XC}_{50})$. Should a median be found outside this range, a threshold $\text{pXC} = 5$ is applied, in keeping with fixed-threshold approaches taken elsewhere [31]. In this way, we ensure that label classes are more balanced to avoid further issues with highly imbalanced data diluting the comparison across different methods; assays where the median falls outside the prescribed range will be filtered if their classes are strongly imbalanced. These represent either very late stage optimization or very early or high-throughput screens. We include only those for which the active ratio falls between 0.30 and 0.70 (see Figure 1b).

Dataset statistics FS-Mol consists of a total of 5120 separate assays, with 233,786 unique compounds. While assays address unique targets to prevent few-shot testing/pretraining overlap, many compounds are measured in multiple assays; $\mathcal{D}_{test}$ contains 27520 compounds, of which 15732 are unseen in $\mathcal{D}_{train}$ and $\mathcal{D}_{valid}$. The resulting task sizes are displayed in Figure 1a, where the mean number of compounds per task is 94, far below alternative datasets, reflecting the highly specific nature of the protein targets and the assays used to explore them.

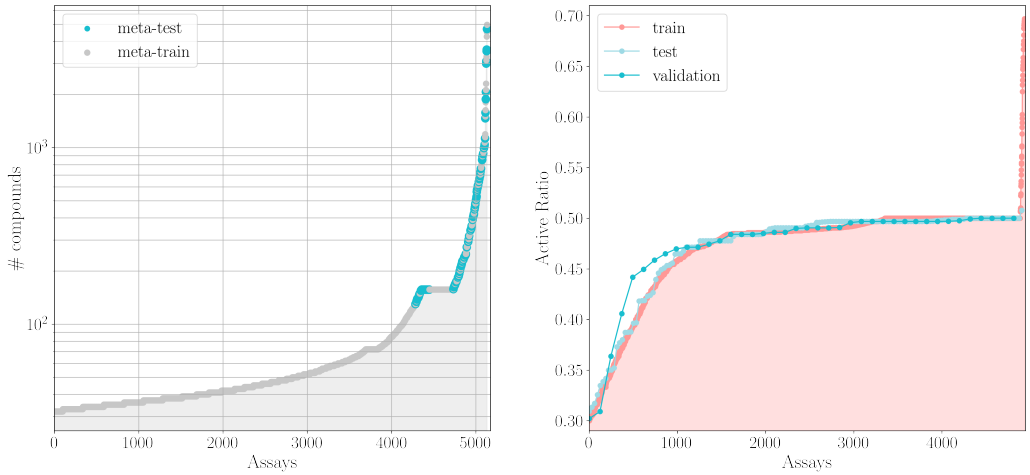
Features To encourage development of diverse approaches, our released dataset and supporting code provide three alternative featurization methods: (1) SMILES strings [61] for each compound, which may be used for NLP-inspired approaches [17, 25, 62] or to derive an arbitrary featurization; (2) Extended Connectivity Fingerprints (ECFP) and key molecular physical descriptors [42], which are standard choices in many machine learning approaches to QSAR; and (3) molecular graphs of atoms and bonds, to be used with methods such as graph neural networks.

4 Related Work

Given the long history of computer-assisted drug discovery, a substantial array of datasets is available for use in machine learning approaches to molecular property prediction. We summarize FS-Mol and related datasets in Table 1. Details on the various related datasets can be found in Appendix B.

We assess the available datasets according to their immediate "out of the box" suitability for the few-shot benchmarking scenario we are interested in.

[2] We note that it is possible, however, for the same target to be addressed by multiple assays, or tasks, within the pretraining set.



(a) Number of compounds in each assay. Meta-testing assays are drawn from larger assays to allow full support-set size comparison. Assays are ordered by size.

(b) The distribution of the proportion of active compounds in all sub-tasks in the dataset. Assays are ordered by proportion of active compounds.

Figure 1: FS-Mol statistics. Assays have sizes ranging from 32-5000, and most contain 30-50% active compounds.

Table 1: Key small molecule activity datasets for multiple tasks. All datasets are set as binary classification problems, although some additionally include underlying activity values (before thresholding).

	Datasets			
	ExCAPE-ML	PCBA	LSC	FS-Mol
# measurements	49,316,517	34,017,170	5,100,411	489,133
# compounds	955,386	437,929	449,391	233,786
# tasks	526	128	1310	5120
Mean # compounds / task	93,758	265,759	3872	94
Median # compounds / task	1820	309,562	320	46
Mean inactive:active / task	268:1	46:1	7:1	1:1
Raw values available?	Yes	No	No	Yes
Source	PubChem/ChEMBL	PubChem	ChEMBL18	ChEMBL27

We require: (i) a large set of related QSAR prediction tasks to enable transfer to a diverse set of unseen prediction targets; (ii) a relatively small number of compounds per task to simulate the early lead optimization phase of a drug discovery project; (iii) well-balanced (i.e. active/inactive) classes, as this reflects a standard approach in QSAR modelling; (iv) a well-defined, context-aware $\{\mathcal{D}_{train}, \mathcal{D}_{test}\}$ task split. Examination of Table 1 reveals that FS-Mol is the only dataset that meets all these requirements. While FS-Mol contains fewer compounds and measurements that all the other compared datasets, it is the only one with a sufficiently large number of tasks where each contain relatively few compounds. We also note that our extraction pipeline enables easy addition of data as more becomes available in future editions of ChEMBL [3]. In addition, all FS-Mol tasks are well balanced by design and the $\{\mathcal{D}_{train}, \mathcal{D}_{test}\}$ split is explicit and fixed.

5 A Molecular Few-Shot Learning Benchmark

We now define a benchmarking methodology able to evaluate the utility of few-shot learning methods.

5.1 Evaluation Methodology

Per-Task Dataset Splitting In the setting of an early-stage drug discovery project, in which some leads (molecular structures demonstrating activity against the target) are known, models are used to evaluate variations of these leads and therefore choose promising development directions. Therefore temporal information regarding inclusion of a molecule in the dataset is the gold-standard to define dataset splits [46]. The data present in ChEMBL does not provide proper temporal information, and hence there is no inherent division of each task into support and query sets.

In the absence of temporal information, two options are regularly considered: (a) random splitting, using several different seeds to reduce the noise introduced by "lucky" splits, and (b) scaffold splitting, in which molecules with similar scaffolds (common, significant substructures) are grouped together, and groups are either entirely in the support or in the query set. The scaffold splitting scenario is clearly much harder and requires better generalization behavior of the considered models, but is also unrealistic in a real-world scenario of modeling project-specific data, where new molecules to be evaluated are variations of existing leads and usually share their scaffolds. Hence, we have decided to instead use repeated random splits for our evaluations.

Concretely, we perform ten-fold stratified random sampling for every task in $\mathcal{D}_{test}$ to create support and query sets. We perform this process for five support set sizes 16, 32, 64, 128, 256, in view of answering **(RQ2)**. Should there be insufficient overall samples to perform a split for a requested support set size, this evaluation point is passed over. The support sets can be used to train a single-task model, or to adapt a pre-trained model at few-shot adaptation time.

Task-Level Metric As core metric to evaluate considered models, we propose area under the precision-recall curve (AUPRC), which is sensitive to the class balance of our query sets. It also allows for a simple comparison to a trivial baseline: the AUPRC of a random classifier is equal to the percentage of positive points in the query set. To help answer **(RQ1)**, we use this observation to focus on the improvement of a given learned model f over that trivial baseline, namely ΔAUPRC $=$ AUPRC$(f(\mathcal{T}_{t,query})) - act_{t,query}/|\mathcal{T}_{t,query}|$, where $act_{t,query}$ refers to the number of active compounds in $\mathcal{T}_{t,query}$.

Dataset-Level Evaluation The task-level evaluation above can be extended to the entire set of test tasks, helping to reduce the noise stemming from particular tasks or "lucky" parameter choices. To this end, we consider the mean ΔAUPRC at different support set sizes (allowing identification of models that are particularly well-suited to very small datasets) and the mean rank in comparison with other methods. Additionally, to reflect the differences in our considered tasks, we also break out evaluation results into smaller categories defined by the different EC numbers.

6 Experimental Baseline Evaluation

In this section, for reference for the community, we present results of standard methods on our new dataset to illustrate the potential and shortcomings of these techniques.

6.1 Baseline Methods

We provide a set of results for all three categories of few-shot learning, with representative methods of the use of this dataset in each. Model implementations used for the experiments are available in our source code release at `https://github.com/microsoft/FS-Mol`.

Single-Task Methods The reigning industry standard in in-silico modeling of very small tasks for drug discovery is the use of random forests (called RF below) and k-nearest neighbour models (kNN) on top of manually curated features (namely, extended connectivity fingerprints [42] and phys-chem descriptors [1]). To answer **(RQ1)**, we include these models in our few-shot learning evaluation. We used typical hyperparameter search configurations for these classes of models based on the extensive

industrial experience of some of the authors. Building on top of the standard scikit-learn [38] library we trained on the support set of each of the test tasks, with hyperparameter choice following a grid-search and validation procedure as detailed in the supporting code and documentation. Our best performing single-task method is RF, which outperforms kNN in all of our experiments.

Additionally, we also consider a graph neural network (GNN) baseline trained from scratch on each individual task, primarily to illustrate that this is not a promising method. Concretely, we use a GNN with 8 layers, a hidden dimension of 128 and a gated readout function similar to Gilmer et al. [20], referring to it as GNN-ST below. For this, as well as for the other GNN-based models introduced below, we determined hyperparameters after a small search in an author-defined space, considering around $\sim$30 configurations per model.

Multi-Task Pretraining A commonly successful approach to few-shot learning is to pretrain a model on a selection of related tasks such that a common model "trunk" learns to extract relevant features, and different "heads" on top of this trunk specialize to the set of known tasks. Such an architecture can then be fine-tuned to a new task by "re-heading", i.e., keeping the trained trunk of the model and adding a freshly initialized head for a new task. Fine-tuning then only needs to adapt the parameters of this new head. This idea has been applied to a number of molecular activity tasks, e.g., by Hu et al. [23].

Here, we represent this approach using a GNN-based multi-task model (called GNN-MT below). Concretely, we use a shared GNN model of 10 layers, hidden dimension 128 and using principal neighborhood message aggregation [14] for all tasks, and then have task-specific gated graph readout functions (as Gilmer et al. [20]) and a task-specific MLP with one hidden layer of dimension 512 to produce an activity label. We train the model on the support sets of all tasks in $\mathcal{D}_{train}$ over multiple epochs. We employ an early stopping criterion based on the suitability for specialization to a new task. More precisely, to evaluate the model during training, we iterate over the tasks in our validation set, and initiate a fine-tuning training process on each task's support set, starting from the pre-trained shared GNN and a freshly initialized graph readout layer and final MLP. For each validation task, we record the ΔAUPRC after fine-tuning, and stop training of the model once the mean of these values has stopped improving. To evaluate the trained model, we follow the same fine-tuning strategy for the tasks in our test set.

Self-Supervised Pretraining After the success of self-supervised methods such as BERT [16], similar methods have been developed for use on molecular data. In particular, Hu et al. [23] introduced the idea of pretraining GNNs by masking and reconstructing of node features and substructures. Maziarka et al. [32] implements a similar idea in the *Molecule Attention Transformer* using a Transformer architecture [57] and reports substantially stronger results. We refer to this as MAT below and treat it as a representative of self-supervised methods, but point out that this area is very active and more recent methods may perform even better. To test it on our dataset, we use the released pre-trained model and code, and fine-tune it on the support set of the test tasks (splitting out 20% of the per-task support set as a validation set to allow training with early stopping). We performed a small hyperparameter search on the validation tasks to identify the learning rate that showed best results in fine-tuning.

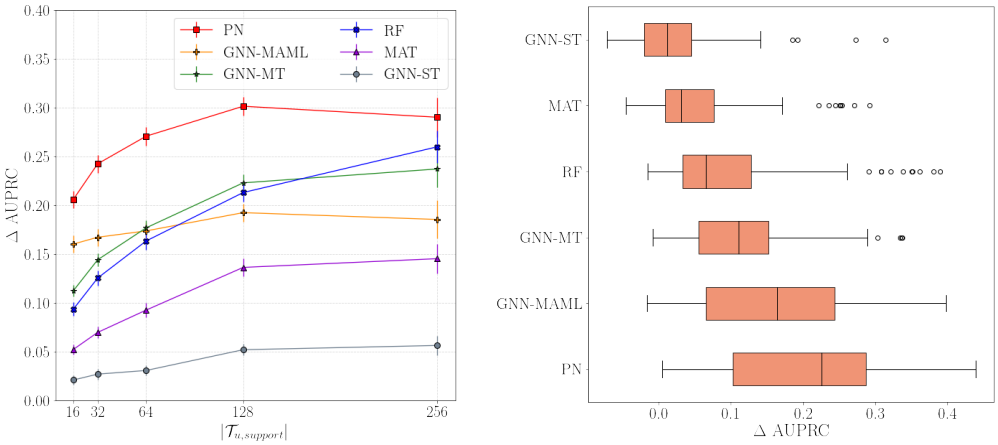
Meta-Learning Methods Meta-learning methods are designed specifically for the few-shot learning setting, aiming to "learn how to learn". In particular, they learn models that are suitable for rapid adaptation to a new task with very small support sets. Commonly, they take an episodic approach to pretraining: in each episode e, a batch of tasks is sampled uniformly $\{\mathcal{T}_k\}_{k=1}^{N_e}$ from $\mathcal{D}_{train}$ and used to train the meta-learner [18]. We choose two key methods as examples here: *optimization-based* meta-learners adapt by taking gradient steps on the support set of a new task [18, 39] and are then used to classify final query examples; *metric-based* approaches compute per-class embeddings of the support set, and classify query examples according to their distance from each [47, 58].

Concretely, we consider Model-Agnostic Meta-Learning [18] procedure on top of a GNN operating on the molecular graph as a representative optimization-based approach and refer to it as GNN-MAML. For this, we use the same GNN as in the GNN-ST case, aiming in particular to evaluate if GNN-MAML succeeds in learning a GNN model that can be rapidly adapted. We use the same procedure to evaluate and fine-tune GNN-MAML as we use for the GNN-MT model.

As a metric-based approach we apply a prototypical network [47] operating on ECFP fingerprints and features extracted by a GNN consuming the molecular graph. We use Mahalanobis distance to measure how similar a new query sample is to representations of the inactive/active classes in the support set of each task, and refer to this model as PN below.

6.2 Empirical Results on FS-Mol

We present the results of the discussed baseline methods on FS-Mol in Figure 2. A complete set of results for each task in $\mathcal{D}_{test}$ is available as supplemental material with our source code release at `https://github.com/microsoft/FS-Mol`.



(a) Mean performance on unseen tasks $\mathcal{T}_u$ as the support set size available for adaptation is increased. We include errors in the means for each point.

(b) Performance across all independent unseen tasks from $\mathcal{D}_{test}$, at support set size 16. The boxes represent the interquartile range across tasks, the extended lines are the (5, 95) percentiles and additional points represent outliers.

Figure 2: Experimental baseline results on all few-shot testing tasks from $\mathcal{D}_{test}$

In Figure 2a, we show how performance varies with the size of the used support set as an aggregation across tasks. Regarding **(RQ1)**, the results indicate that while GNN-MAML is able to provide gains when only very little training data is present, the random forests often used in real-world applications are doing very well for small-to-medium-sized training sets. However, PN provides a significant performance increase over RF methods, in particular on tasks with few datapoints. Other pre-trained methods (GNN-MT and MAT) either perform comparably to the single-task baseline or under-perform it substantially. The plot also answers **(RQ2)**, showing that availability of more training data specific to the task can lead to dramatic improvements for most methods. However, for our meta-learning model GNN-MAML, leveraging information from additional datapoints seems to be challenging, pointing to an interesting question for future research. We note that as $|\mathcal{T}_{u,support}|$ increases, some tasks from $\mathcal{D}_{test}$ can no longer be included as insufficient datapoints are available; the resulting smaller pool of few-shot testing tasks happen to derive a smaller benefit from both the meta-trained and multitask models in this case. This leads to the slight decrease in performance on the aggregation over tasks at $|\mathcal{T}_{u,support}| = 256$.

The results also demonstrate that GNN-ST is clearly outperformed by its pre-trained variants GNN-MT and GNN-MAML **(RQ1)**. Similarly, MAT, which uses self-supervised pretraining unrelated to specific tasks, shows some improvement over GNN-ST, but underperforms the GNN-MT, GNN-MAML, and PN models that were pre-trained using task-specific information. The detailed results for support set size 16 in Figure 2b further support these observations, indicating that PN substantially outperforms all other methods in this scenario. However, we stress that performance is highly task-dependent, as evidenced by the broad range of improvements over the full set of few-shot testing tasks in Figure 2b. While in certain tasks, few-shot learning methods clearly outperform single-task, this is sometimes not the case. The degree to which information transfer can occur to a specific new task in few-shot learning is an open question [5, 8].

Table 2: Baseline results on few-shot learning methods on FS-Mol with support set size 16. Results are broken down by few-shot testing class. The reported metric is ΔAUPRC, accounting for the variation in percentage of active compounds in each few-shot testing task. The mean rank is calculated by *autorank* [22], following [15]. Errors in the mean are reported across aggregate task categories, errors across multiple support set splits are present for classes containing single tasks.

few-shot task categories			ΔAUPRC					
Class	Description	#tasks	RF	GNN-ST	GNN-MT	MAT	GNN-MAML	PN
1	oxidoreductases	7	0.081±0.032	0.013±0.019	0.045±0.013	0.062±0.024	0.046±0.023	**0.086±0.029**
2	kinases	125	0.082±0.006	0.013±0.004	0.113±0.006	0.043±0.005	0.178±0.009	**0.217±0.009**
3	hydrolases	20	0.158±0.026	0.062±0.019	0.129±0.025	0.095±0.019	0.106±0.024	**0.196±0.031**
4	lysases	2	0.218±0.172	0.161±0.112	0.189±0.100	0.139±0.105	0.218±0.147	**0.229±0.201**
5	isomerases	1	0.119±0.029	-0.014±0.015	0.083±0.054	0.040±0.044	0.006±0.021	**0.117±0.047**
6	ligases	1	0.027±0.069	-0.011±0.003	0.046±0.050	0.010±0.075	0.001±0.017	**0.058±0.074**
7	translocases	1	**0.102±0.053**	0.002±0.043	0.042±0.031	0.067±0.035	-0.002±0.021	0.055±0.021
	all enzymes	157	0.093±0.007	0.021±0.005	0.112±0.006	0.052±0.005	0.160±0.009	**0.206±0.008**
	mean rank		3.55	5.22	3.17	4.74	2.75	**1.56**

Table 2 breaks these results up by EC category. In particular, it shows that the largest category of tasks, kinases, which is also most common in the training data (1497 training tasks), clearly profits from meta-learning. Classes 6 and 7, in contrast, are represented in the training set by fewer than 40 tasks each. These results match the intuition that transfer of learned knowledge between more related tasks is more successful [8]. By similar reasoning, we expect the addition of more training tasks to improve few-shot learners. We confirm this empirically in Appendix C, where we show that randomly sub-sampling $\mathcal{D}_{train}$ makes transfer to $\mathcal{D}_{test}$ less effective.

7 Discussion & Conclusion

We presented FS-Mol, an up-to-date molecular dataset suitable for evaluation of few-shot learning methods in an important domain outside of computer vision and NLP. In particular, our dataset is chosen to match the modeling task in early-stage drug discovery projects, where the aim is to identify and optimize leads for a specific protein target given very little data. FS-Mol has enabled us to address key research questions. Specifically, it is now possible to evaluate if, and by how much, (new) few-shot methods improve over standard baselines in a realistic QSAR evaluation setting. Our baselines were chosen to be representative of industry practice, so that progress on FS-Mol is more likely to translate into advances in tools for practitioners. Finally, our experiments also allow us to determine the support set sizes for which a few-shot method is likely to be useful, for example showing that MAML is most successful at very small support set sizes.

However, we note that transfer of results to realistic projects is not guaranteed to be successful. In particular, it may be the case that current drug-discovery projects are sufficiently novel such that transferring knowledge from the public ChEMBL corpus does not provide any benefits. This may be alleviated by pre-training few-shot models on more recent, proprietary data that better matches current industry projects. We note that it is an ongoing area of research to understand when transfer-learning is likely to assist on an unseen task [8]. In addition, the few-shot baselines we provide checkpoints and results for are only a representative set, rather than a complete survey of the current state of the field, and so variations of the methods chosen by us may already provide significant boosts.

We are releasing FS-Mol, along with the preprocessing pipeline used to produce it and an extensible implementation of key baseline approaches, illustrating how to benchmark new methods. We hope that the dataset and the implementation will drive further research in few-shot learning, with a focus towards methods that are useful in this important application domain.

Societal Impact While the automation of components of the drug-discovery process could lead to redundancies in the pharmaceutical industry, better QSAR models alone are unlikely to have a large impact, as the tools developed by using this dataset are considerably more likely to be assistive tools for, rather than replacements of, professionals. Finally, we note the potential positive impact of reducing the time and cost required bring a drug to market and the potential of developing more effective treatments using large-scale computational methods rather than the current labour-intensive design-synthesize-measure cycle.

References

[1] Rdkit: Open-source cheminformatics. URL http://www.rdkit.org.

[2] Enzyme nomenclature 1992 : recommendations of the nomenclature committee of the international union of biochemistry and molecular biology on the nomenclature and classification of enzymes. 1992.

[3] ChEMBL: towards direct deposition of bioassay data, 2019. URL https://doi.org/10.1093/nar/gky1075.

[4] Melloddy: Machine learning ledger orchestration for drug discovery, 2021. URL https://www.melloddy.eu/.

[5] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. Fowlkes, S. Soatto, and P. Perona. Task2vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6429–6438, 2019. doi: 10.1109/ICCV.2019.00653.

[6] G. E. H. Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2012. doi: 10.1145/3065386. URL https://doi.org/10.1145/3065386.

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

[8] D. Alvarez-Melis and N. Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 2020-December(NeurIPS), 2020. ISSN 10495258.

[9] I. I. Baskin. Is one-shot learning a viable option in drug discovery? *Expert opinion on drug discovery*, 14(7):601–603, 2019.

[10] L. Bertinetto, P. H. Torr, J. Henriques, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, and et al. Language models are few-shot learners. *arxiv*, 2020. URL https://arxiv.org/abs/2005.14165v1.

[12] D. Chen, Y. Chen, Y. Li, F. Mao, Y. He, and H. Xue. Self-supervised learning for few-shot image classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1745–1749, 2021. doi: 10.1109/ICASSP39728.2021.9413783.

[13] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177.

[14] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Velickovic. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006. doi: 10.5555/1248547.1248548. URL https://doi.org/10.5555/1248547.1248548.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[17] B. Fabian, T. Edlich, H. Gaspar, M. Segler, J. Meyers, M. Fiscato, and M. Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

[18] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *34th International Conference on Machine Learning, ICML 2017*, 3:1856–1868, 2017.

[19] M. Gamelo, D. Rosenbaum, C. J. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. J. Rezende, and S. M. Eslami. Conditional neural processes. *35th International Conference on Machine Learning, ICML 2018*, 4:2738–2747, 2018.

[20] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML)*, 2017.

[21] J. Gordon, J. Bronskill, S. Nowozin, M. Bauer, and R. E. Turner. Meta-learning probabilistic inference for prediction. *7th International Conference on Learning Representations, ICLR 2019*, pages 1–22, 2019.

[22] S. Herbold. Autorank: A python package for automated ranking of classifiers. *Journal of Open Source Software*, 5(48):2173, 2020. doi: 10.21105/joss.02173. URL `https://doi.org/10.21105/joss.02173`.

[23] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. S. Pande, and J. Leskovec. Pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[24] Y. Hu, V. Gripon, and S. Pateux. Leveraging the Feature Distribution in Transfer-based Few-Shot Learning. 2020. URL `http://arxiv.org/abs/2006.03806`.

[25] S. Jastrzębski, D. Leśniak, and W. M. Czarnecki. Learning to smile (s). *arXiv preprint arXiv:1602.06289*, 2016.

[26] J. Jumper, R. Evans, A. Pritzel, and et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021. doi: 10.1038/s41586-021-03819-2. URL `https://doi.org/10.1038/s41586-021-03819-2`.

[27] T. Kalliokoski, C. Kramer, A. Vulpetti, and P. Gedeck. Comparability of mixed ic50 data – a statistical analysis. *PLOS ONE*, 8(4):1–12, 04 2013. doi: 10.1371/journal.pone.0061007. URL `https://doi.org/10.1371/journal.pone.0061007`.

[28] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton. PubChem in 2021: new data content and improved web interfaces. *Nucleic acids research*, 49(D1):D1388–D1395, 2021.

[29] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum. One shot learning of simple visual concepts. *In Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2011.

[30] D. Massiceti, L. Zintgraf, J. Bronskill, L. Theodorou, M. T. Harris, E. Cutrell, C. Morrison, K. Hofmann, and S. Stumpf. ORBIT: A Real-World Few-Shot Dataset for Teachable Object Recognition. 2021. doi: 10.25383/city.14294597. URL `http://arxiv.org/abs/2104.03841%0Ahttp://dx.doi.org/10.25383/city.14294597`.

[31] A. Mayr, G. Klambauer, T. Unterthiner, M. Steijaert, J. K. Wegner, H. Ceulemans, D. A. Clevert, and S. Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical Science*, 9(24):5441–5451, 2018. ISSN 20416539. doi: 10.1039/c8sc00148k.

[32] L. Maziarka, T. Danel, S. Mucha, K. Rataj, J. Tabor, and S. Jastrzebski. Molecule attention transformer. In *Graph Representation Learning Workshop at NeurIPS 2019 (GRL)*, 2019.

[33] E. N. Muratov, J. Bajorath, R. P. Sheridan, I. V. Tetko, D. Filimonov, V. Poroikov, T. I. Oprea, I. I. Baskin, A. Varnek, A. Roitberg, et al. Qsar without borders. *Chemical Society Reviews*, 49 (11):3525–3564, 2020.

[34] A. Nakamura and T. Harada. Revisiting fine-tuning for few-shot learning. *ArXiv*, abs/1910.00216, 2019.

[35] C. Q. Nguyen, C. Kreatsoulas, and K. M. Branson. Meta-Learning GNN Initializations for Low-Resource Molecular Property Prediction. In *Graph Representation Learning and Beyond Workshop at ICML (GRL+)*, 2020. URL http://arxiv.org/abs/2003.05996.

[36] A. Pappu and B. Paige. Making graph neural networks worth it for low-data molecular machine learning. *CoRR*, abs/2011.12203, 2020. URL https://arxiv.org/abs/2011.12203.

[37] E. Park. Meta-Curvature. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[39] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *International Conference on Learning Representations (ICLR)*, 2020. URL http://arxiv.org/abs/1909.09157.

[40] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande. Massively Multitask Networks for Drug Discovery. (Icml), 2015. URL http://arxiv.org/abs/1502.02072.

[41] J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, and R. E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 32(i), 2019. ISSN 10495258. doi: 10.17863/CAM.47574.

[42] D. Rogers and M. Hahn. Extended-Connectivity Fingerprints. 50(5):742–754. doi: 10.1021/ci100050t. URL https://doi.org/10.1021/ci100050t.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

[44] P. Schneider, W. P. Walters, A. T. Plowright, N. Sieroka, J. Listgarten, R. A. Goodnow, J. Fisher, J. M. Jansen, J. S. Duca, T. S. Rush, M. Zentgraf, J. E. Hill, E. Krutoholow, M. Kohler, J. Blaney, K. Funatsu, C. Luebkemann, and G. Schneider. Rethinking drug design in the artificial intelligence era. 19(5):353–364. ISSN 1474-1784. doi: 10.1038/s41573-019-0050-3. URL https://doi.org/10.1038/s41573-019-0050-3.

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

[46] R. P. Sheridan. Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. 53(4):783–790. ISSN 1549-9596. doi: 10.1021/ci400084k. URL https://doi.org/10.1021/ci400084k.

[47] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4080–4090, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[48] M. J. Stewart and I. D. Watson. Standard units for expressing drug concentrations in biological fluids. 16(1):3–7. doi: 10.1111/j.1365-2125.1983.tb02136.x. URL https://pubmed.ncbi.nlm.nih.gov/6882621.

[49] F. Strieth-Kalthoff, F. Sandfort, M. H. Segler, and F. Glorius. Machine learning the ropes: principles, applications and directions in synthetic chemistry. *Chemical Society Reviews*, 49 (17):6154–6168, 2020.

[50] N. Sturm, A. Mayr, T. Le Van, V. Chupakhin, H. Ceulemans, J. Wegner, J. F. Golib-Dzib, N. Jeliazkova, Y. Vandriessche, S. Böhm, V. Cima, J. Martinovic, N. Greene, T. Vander Aa, T. J. Ashby, S. Hochreiter, O. Engkvist, G. Klambauer, and H. Chen. Industry-scale application and evaluation of deep learning for drug target prediction. *Journal of Cheminformatics*, 12 (1):1–13, 2020. doi: 10.1186/s13321-020-00428-5. URL https://doi.org/10.1186/s13321-020-00428-5.

[51] J. Sun, N. Jeliazkova, V. Chupakhin, J.-F. Golib-Dzib, O. Engkvist, L. Carlsson, J. Wegner, H. Ceulemans, I. Georgiev, V. Jeliazkov, N. Kochev, T. J. Ashby, and H. Chen. ExCAPE-DB: an integrated large scale dataset facilitating Big Data analysis in chemogenomics. 9(1):17. ISSN 1758-2946. doi: 10.1186/s13321-017-0203-5. URL https://doi.org/10.1186/s13321-017-0203-5.

[52] D. Sydow, L. Burggraaff, A. Szengel, H. W. T. van Vlijmen, A. P. IJzerman, G. J. P. van Westen, and A. Volkamer. Advances and Challenges in Computational Target Prediction. 59(5): 1728–1742. doi: 10.1021/acs.jcim.8b00832. URL https://doi.org/10.1021/acs.jcim.8b00832.

[53] S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[54] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola. Rethinking Few-Shot Image Classification: A Good Embedding is All You Need? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12359 LNCS:266–282, 2020. ISSN 16113349. doi: 10.1007/978-3-030-58568-6_16.

[55] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. J. Swersky, P.-A. Manzagol, and H. Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations (submission)*, 2020.

[56] A. Varnek and I. Baskin. Machine learning methods for property prediction in chemoinformatics: quo vadis? *Journal of chemical information and modeling*, 52(6):1413–1437, 2012.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[58] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, (Nips):3637–3645, 2016. ISSN 10495258.

[59] W. P. Walters and R. Barzilay. Applications of Deep Learning in Molecule Generation and Molecular Property Prediction. 54(2):263–270, 2021. ISSN 0001-4842. doi: 10.1021/acs.accounts.0c00699. URL https://doi.org/10.1021/acs.accounts.0c00699.

[60] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, Z. Zhou, L. Han, K. Karapetyan, S. Dracheva, B. A. Shoemaker, E. Bolton, A. Gindulyte, and S. H. Bryant. PubChem's BioAssay database. *Nucleic Acids Research*, 40(D1), 2012. ISSN 03051048. doi: 10.1093/nar/gkr1132.

[61] D. Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. 28(1):31–36, 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005. URL https://pubs.acs.org/doi/abs/10.1021/ci00057a005.

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

[63] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. MoleculeNet: a benchmark for molecular machine learning. 9(2):513–530. doi: 10.1039/C7SC02664A. URL http://dx.doi.org/10.1039/C7SC02664A.

[64] L. Zintgraf, K. Shiarlis, V. Kurin, K. Hofmann, and S. Whiteson. Fast context adaptation via meta-learning. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, pages 13262–13276, 2019. ISBN 9781510886988.