A generalized framework for embedding-based few-shot learning methods in drug discovery

Johannes Schimunek [1]

Lukas Friedrich [2] **Daniel Kuhn** [2] **Friedrich Rippmann** [2]

Sepp Hochreiter [1] **Günter Klambauer** [1]

[1] ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning,
Johannes Kepler University Linz, Austria
[2] Computational Chemistry & Biologics, Merck Healthcare, Darmstadt, Germany

Abstract

Building models for molecular properties and activities based on very few measurements is a central problem in drug discovery. Almost all drug discovery projects start with one or few known active molecules and face the problem of selecting promising molecules for screening. Therefore, few-shot learning methods have been introduced to computer-aided drug design, which have the potential to improve this critical phase of the drug discovery process. However, it is currently unclear how they relate to each other and how they compare to classical chemoinformatics methods, such as Similarity Search. In this work, we present a generalized framework, by which we can explain the relation of few-shot learning methods in drug discovery. This framework reveals under-explored architectures and relations to classical chemoinformatics methods. Experiments on the few-shot benchmarking dataset FS-Mol show that classic chemoinformatics methods outperform several recent few-shot learning methods and suggest novel promising architecture designs.

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 (Mayr et al., 2018; Chen et al., 2018; Walters and Barzilay, 2021), diminishing the required number of synthesized molecules and wet-lab measurements (Merk et al., 2018; Schneider et al., 2020). These Deep Learning models usually just rely on molecular information about the ligands, on which they are trained to yield highly accurate property and activity prediction (Mayr et al., 2020; 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 are data-hungry (Marcus, 2018) and thus require large amounts of biological measurements. However, for Deep Learning-based activity and property prediction to reach 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). Wu et al. (2018) published a large scale benchmark for molecular machine learning, including among others prediction models for the SIDER dataset with an average of 5,187 data points, Tox21 (Huang et al., 2016; Mayr et al., 2016) with on average 9,031, and ClinTox (Wu et al., 2018) with 1,491

measurements per task. However, for many drug targets the amount of available measurements is very limited (Stanley et al., 2021; Altae-Tran et al., 2017; Waring et al., 2015) because of the resources required for in-vitro experiments. Therefore, methods that can exploit few measurements to build valuable property and activity prediction models are required. This problem of efficiently training models with few available data points is the focus of the machine learning areas of **meta-learning** (Schmidhuber, 1987; Bengio et al., 1990; Hochreiter et al., 2001) and **few-shot learning** (Miller et al., 2000; Bendre et al., 2020; Wang et al., 2020).

Few-shot learning methods tackle the central low-data problem 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 (Adler et al., 2020; Stanley et al., 2021; Altae-Tran et al., 2017). Few-shot learning methods are usually classified into four categories according to their main approach to tackle the low-data problem (Bendre et al., 2020; Wang et al., 2020). a) Data-augmentation-based approaches tackle the problem of little 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) In embedding-based and nearest neighbour approaches, representations in a meaningful embedding space are learned. By comparing these embeddings, low-shot predictions for new input data can be realized. For Matching Networks (Vinyals et al., 2016), e.g., an attention mechanism, which rely 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) In optimization-based or fine-tuning methods, a meta-optimizer focuses on efficiently navigating the parameter space, for example, by learning initial weights that can be adapted to a novel task by few optimization steps (Finn et al., 2017). d) Semantic-based approaches use additional given semantics together with the input data to learn or to optimize for new tasks (Schwartz et al., 2019; Li et al., 2019). In the field of drug discovery, Adler et al. (2020) proposed a method for cross-domain few-shot learning based on representation fusion. Nguyen et al. (2020) evaluated the applicability of MAML and MAML variants to 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 embeddings of the support set molecules are learned. Recently, Stanley et al. (2021) generated a benchmark dataset for few-shot learning methods in drug discovery and provided some baseline results. However, it is still unclear how these methods are connected to classic chemoinformatics techniques, such as Similarity Search, that have previously been used for few-shot learning (Cereto-Massagué et al., 2015). Therefore, we suggest a generalized framework for embedding-based few-shot learning models and present strong new baselines for the FS-Mol dataset (Stanley et al., 2021).

In this work, our contributions are the following:

- We present a framework for embedding-based few-shot learning methods in drug discovery, from which classic chemoinformatics and Deep Learning methods arise as special cases.
- We provide additional baseline methods for the few-shot learning dataset FS-Mol.
- Our experiments show that a neural variant of Similarity Search performs best on the benchmarking dataset FS-Mol.

2 Problem setting

Drug discovery projects revolve around models $g(\boldsymbol{m})$ that can predict a molecular property or activity $\hat{y}$ given an input molecule $\boldsymbol{m}$ from a chemical space $\mathcal{M}$. We consider machine-learning models $\hat{y} = g_{\boldsymbol{w}}(\boldsymbol{m})$ with adaptive parameters $\boldsymbol{w}$ that have been found on a training dataset. For property prediction based on Deep Learning, the models typically comprise an adaptive molecule encoder $\boldsymbol{h}^{\text{mol}} : \mathcal{M} \to \mathbb{R}^d$. The molecule encoder can operate on different 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 could be pre-trained on related property prediction tasks.

For few-shot learning, the goal is to generate a good 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 inactive and active molecules. This set $\boldsymbol{Z} = \{\boldsymbol{X}, \boldsymbol{y}\}$ is called the *support set* and assumed to be sampled from a prediction task. Because

N is small, the usual strategy to adapt the parameters $\boldsymbol{w}$ of the model $g_{\boldsymbol{w}}$ from scratch to the given support set does not perform well. Therefore, few-shot learning methods use more prudent strategies to utilize the support set, for example fine-tuning the parameters of a pre-trained model, which we will detail in the following.

3 Drug-target association models: A generalized framework for embedding-based few-shot learning methods in drug discovery

We will show that embedding-based few-shot learning models $g_{\boldsymbol{w}}(\boldsymbol{m}, \boldsymbol{Z})$ in drug discovery have the following principle structure, from which several classic chemoinformatics methods and few-shot learning methods arise as special cases:

$$\hat{y} = g_{\boldsymbol{w}}(\boldsymbol{m}, \boldsymbol{Z}) = \boldsymbol{h}^{\text{assoc}}\left(\boldsymbol{h}^{\text{mol}}(\boldsymbol{m}),\ \boldsymbol{h}^{\text{mem}}(\boldsymbol{Z})\right), \tag{1}$$

where $\boldsymbol{m}$ is the input molecule, $\boldsymbol{Z}$ is the support set, and $\hat{y}$ is the prediction for the given molecule. The association function $\boldsymbol{h}^{\text{assoc}}$, the molecule encoder function $\boldsymbol{h}^{\text{mol}}$, and the memory encoder function $\boldsymbol{h}^{\text{mem}}$ can be chemoinformatics operations, such as descriptor or similarity calculation, or Deep Learning layers. With particular choices of these three functions, both traditional and recent few-shot learning methods can be recovered. The support set $\boldsymbol{Z}$ can be considered as an external memory (Wang et al., 2020), which is accessed by different mechanisms. With this perspective, $g_{\boldsymbol{w}}(\boldsymbol{m}, \boldsymbol{Z})$ can be interpreted as a *drug-target association model*, in which a target corresponds to a few-shot task, which is represented by the samples in the memory $\boldsymbol{Z} = \{\boldsymbol{X}, \boldsymbol{y}\}$. In the following, we demonstrate that embedding-based few-shot methods arise as special cases from this framework.

From Eq. (1), it is evident that there are two basic categories of this architecture: a) Pooling across samples in the support set is performed by $\boldsymbol{h}^{\text{mem}}$ and $\boldsymbol{h}^{\text{assoc}}$ associates the given molecule $\boldsymbol{m}$ with a representation of the support set, or b) pooling is performed later. $\boldsymbol{h}^{\text{mem}}$ procures updated representations of the support set molecules and $\boldsymbol{h}^{\text{assoc}}$ associates the given molecule with the molecules in the support set and pools afterwards. Both strategies exist for the classic Similarity Search (see below) as well es for the Deep Learning based few-shot methods, such as Matching Networks and Prototypical Networks (see further below).

Similarity Search. Similarity Search (Cereto-Massagué et al., 2015) is a classic chemoinformatics technique used in situations in which a single or few actives are known. In the simplest case, molecules that are similar to a given active molecule are searched by computing a fingerprint or descriptor-representation $\boldsymbol{h}^{\text{desc}}(\boldsymbol{m})$ 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:

$$\hat{y} = 1/N \sum_{n=1}^{N} y_n\ k(\boldsymbol{h}^{\text{desc}}(\boldsymbol{m}), \boldsymbol{h}^{\text{desc}}(\boldsymbol{x}_n)), \tag{2}$$

where the function $\boldsymbol{h}^{\text{desc}}$ maps the molecule to its chemical descriptors or fingerprints and takes the role of both the molecule encoder $\boldsymbol{h}^{\text{mol}}$ and the memory encoder $\boldsymbol{h}^{\text{mem}}$. The association function $\boldsymbol{h}^{\text{assoc}}$ consists of a) the similarity measure $k(.,.)$ and then b) mean pooling across molecules weighted by their similarity and activity.

Notably, there are many variants of Similarity Search (Cereto-Massagué et al., 2015; Wang et al., 2010; Eckert and Bajorath, 2007; Geppert et al., 2008; Willett, 2014; Sheridan and Kearsley, 2002; Riniker and Landrum, 2013) of which some correspond to recent few-shot learning methods with a fixed molecule encoder. For example, Geppert et al. (2008) suggest to use centroid molecules, i.e. prototypes or averages of active molecules, which is equivalent to the idea of Prototypical Networks (Snell et al., 2017). Riniker and Landrum (2013) are aware of different fusion strategies for sets of active or inactive molecules, which corresponds to different pooling strategies of the support set. Overall, the variants of the classic Similarity Search are highly similar to embedding-based few-shot learning methods except that they have a fixed instead of a learned molecule encoder.

Neural Similarity Search and Siamese Neural Networks. In contrast to the classic Similarity Search that uses fixed molecule encoders, a neural variant arises naturally, in which the molecule encoder and the memory encoder are learned on a training set. If these encoder share weights, the approach is known as metric learning with Siamese Networks (Koch et al., 2015; Hertz et al.,

2006; Ye and Guo, 2018) and has been suggested for drug discovery by Torres et al. (2020), using Convolutional Neural Networks as molecule encoders. $\boldsymbol{h}^{\text{assoc}}$ consists of three functions which are firstly a distance or similarity metric, secondly mean pooling over the labels of the support set molecules weighted by their similarity values and thirdly the sigmoid function (Koch et al., 2015). In this work, we implemented a variant of Neural Similarity Search (Section 4) and our experiments show that this is the best-performing few-shot method on the FS-Mol dataset (Section 5).

Matching Networks. For Matching Networks (Vinyals et al., 2016), $\boldsymbol{h}^{\text{mol}}$ and $\boldsymbol{h}^{\text{mem}}$ map the input molecule and the support set molecules to embeddings. $\boldsymbol{h}^{\text{assoc}}$ consists of three parts. Firstly, the representation of the input molecule is updated, using an attention-enhanced LSTM (Hochreiter and Schmidhuber, 1997; Vinyals et al., 2015, 2016) variant. Secondly, attention weights are computed, based on the updated input molecule embedding and the support set molecule representations. In a third step, these attention weights are used to compute a weighted sum over the activity labels (Appendix A.1.1).

IterRefLSTM. Altae-Tran et al. (2017) modified the idea of Matching Networks by replacing the LSTM with their Iterative Refinement Long Short-Term Memory (IterRefLSTM). The use of the IterRefLSTM empowers $\boldsymbol{h}^{\text{assoc}}$ to update not only the embeddings for the input molecule but also adjust the representations of the support set molecules (Appendix A.1.2).

Prototypical Networks (ProtoNet) (Snell et al., 2017) also include a molecule encoder $\boldsymbol{h}^{\text{mol}}$ and a memory encoder $\boldsymbol{h}^{\text{mem}}$. In contrast to models which associate the input molecule with a set of molecules in the embedding space, the memory encoder returns prototypical representations of each class by class-wise building the mean across all related support set molecules in the embedding space. These prototypical representations build together with the query molecule embedding the inputs for $\boldsymbol{h}^{\text{assoc}}$ (Appendix A.1.3).

4 Neural Similarity Search variant

As noted above, a lot of related work (Koch et al., 2015; Hertz et al., 2006; Ye and Guo, 2018; Torres et al., 2020) already was done. We adapted these ideas, such that a fully-connected Deep Neural Network followed by a Layer Normalization operation, $\boldsymbol{h}_{\boldsymbol{w}}$, is used in a Siamese fashion to compute the representations for the input molecule and the support set molecules. Within the association function block, pairwise similarity values for the input molecule and each support set molecule are computed, associating both embeddings via the dot product. Based on these similarity values, the activity for the input molecule is predicted, building the weighted mean over the support set molecule labels:

$$\hat{y} = \boldsymbol{\sigma}\left(c \cdot 1/N \sum_{n=1}^{N} y_n \cdot \boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{m})^T \cdot \boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{x}_n)\right), \tag{3}$$

where $\boldsymbol{\sigma}$ is the sigmoid function and c is a constant set to 1 divided by the embedding space dimension. The hyperparameters for the fully-connected Deep Neural Network can be found in the Appendix (A.3). Figure 1 provides an schematic overview of the Neural Similarity Search variant.

5 Experiments and results

Dataset. We use the benchmark dataset FS-Mol (Stanley et al., 2021), that has been suggested to evaluate few-shot learning methods in drug discovery (details in Section A.2).

Methods compared. We implemented the classic chemoinformatics technique "Similarity Search" as baseline to compare it to the methods originally included in Stanley et al. (2021). Furthermore, we compared the neural variant of the similarity search ("Neural Similarity Search"). The architectures and hyperparameters are reported in the Appendix (A.3). All hyperparameters were selected on the validation set provided by FS-Mol.

Multi-Task Pretraining. We pretrained both architectures on the predefined training fold, using a mini-batch size of 4096. As all available data points in the training set were shuffled, each mini-batch consists of data points from multiple tasks.

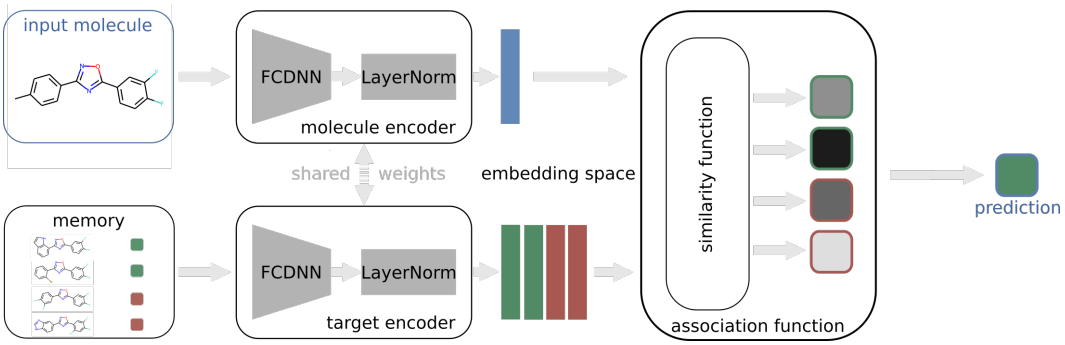

Figure 1: Schematic overview of the implemented Neural Similarity Search variant

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).

Method	All	Kinases	Hydrolases	Oxidored
Random Forest (Stanley et al., 2021)	.092 ± .007	.081 ± .009	.158 ± .028	.080 ± .029
GNN-ST (Stanley et al., 2021)	.029 ± .004	.027 ± .004	.040 ± .018	.020 ± .016
GNN-MT (Stanley et al., 2021)	.093 ± .006	.093 ± .006	.108 ± .025	.053 ± .018
MAT (Stanley et al., 2021)	.052 ± .005	.043 ± .005	.095 ± .019	.062 ± .024
GNN-MAML (Stanley et al., 2021)	.159 ± .009	.177 ± .009	.105 ± .024	.054 ± .028
ProtoNet (Stanley et al., 2021)	.207 ± .008	.215 ± .009	.209 ± .030	.095 ± .029
Similarity Search (this work)	.118 ± .011	.113 ± .008	.117 ± .009	.157 ± .012
Neural Similarity Search (this work)	**.223** ± .011	**.219** ± .011	**.223** ± .011	**.322** ± .011

Support sets for validation and test tasks. For each task, eight active and eight inactive molecules are drawn for the support set from all available data points. All other measurements are used for the performance evaluation.

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. Additionally, the variability across drawn support sets and training re-runs can be found in the Section A.4. Notably, the classic Similarity Search has outperformed most few-shot learning methods, such as single-task methods (RF, GNN-ST) and multi-task pretraining methods (GNN-MT, MAT). The variant of the Neural Similarity Search has outperformed all previously suggested methods (p-value .03, Binomial test across re-runs).

Conclusion. Our work has connected the chemoinformatics and the Deep Learning field concerning few-shot drug discovery methods. Our generalized framework readily suggest new architectures, of which we have implemented some as additional baselines for the FS-Mol benchmarking dataset. We envision that our work initiates ideas for new architectures for few-shot drug discovery methods.

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*.

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. (1990). *Learning a synaptic learning rule*. Citeseer.

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, 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.

Eckert, H. and Bajorath, J. (2007). Molecular similarity analysis in virtual screening: foundations, limitations and novel approaches. *Drug discovery today*, 12(5-6):225–233.

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.

Geppert, H., Horváth, T., Gärtner, T., Wrobel, S., and Bajorath, J. (2008). Support-vector-machine-based ranking significantly improves the effectiveness of similarity searching using 2d fingerprints and multiple reference compounds. *Journal of chemical information and modeling*, 48(4):742–746.

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.

Hertz, T., Hillel, A. B., and Weinshall, D. (2006). Learning a kernel function for classification with small training samples. In *Proceedings of the 23rd international conference on machine learning*, pages 401–408.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

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.

Huang, R., Xia, M., Sakamuru, S., Zhao, J., Shahane, S. A., Attene-Ramos, M., Zhao, T., Austin, C. P., and Simeonov, A. (2016). 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.

Li, A., Luo, T., Lu, Z., Xiang, T., and Wang, L. (2019). Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7212–7220.

Li, P. (2016). Generalized min-max kernel and generalized consistent weighted sampling. *arXiv preprint arXiv:1605.05721*.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

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., and Hochreiter, S. (2020). The lsc benchmark dataset: Technical appendix and partial reanalysis. Technical report, LIT AI Lab & Institute for Machine Learning, Johannes Kepler University Linz.

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.

Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., De Veij, M., Félix, E., Magariños, M. P., Mosquera, J. F., Mutowo, P., Nowotka, M., et al. (2019). Chembl: towards direct deposition of bioassay data. *Nucleic acids research*, 47(D1):D930–D940.

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*.

Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. (2005). Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110.

Riniker, S. and Landrum, G. A. (2013). Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of cheminformatics*, 5(1):1–17.

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.

Schwartz, E., Karlinsky, L., Feris, R., Giryes, R., and Bronstein, A. M. (2019). Baby steps towards few-shot learning with multiple semantics. *arXiv preprint arXiv:1906.01905*.

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*.

Sheridan, R. P. and Kearsley, S. K. (2002). Why do we need so many chemical similarity search methods? *Drug discovery today*, 7(17):903–911.

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.

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.

Torres, L., Monteiro, N., Oliveira, J., Arrais, J., and Ribeiro, B. (2020). Exploring a siamese neural network architecture for one-shot drug discovery. In *2020 IEEE 20th international conference on bioinformatics and bioengineering (bibe)*, pages 168–175. IEEE.

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*, volume 2014, pages 1–4. NeurIPS.

Vinyals, O., Bengio, S., and Kudlur, M. (2015). Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*.

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, X., Huan, J., Smalter, A., and Lushington, G. H. (2010). Application of kernel functions for accurate similarity search in large chemical databases. In *BMC bioinformatics*, volume 11, pages 1–14. BioMed Central.

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.

Willett, P. (2014). The calculation of molecular structural similarity: principles and practice. *Molecular informatics*, 33(6-7):403–413.

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.

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.

Ye, M. and Guo, Y. (2018). Deep triplet ranking networks for one-shot recognition. *arXiv preprint arXiv:1804.07275*.

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.

A Appendix

A.1 Drug-target association models: A generalized framework for embedding-based few-shot learning methods in drug discovery

To simplify notation, we use $\boldsymbol{h}^{\text{mem}}$ in two different settings, i. e. $\boldsymbol{h}^{\text{mem}}(\boldsymbol{X}) = \boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{X})$ and $\boldsymbol{h}^{\text{mem}}(\boldsymbol{Z}) = (\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{X}), \boldsymbol{y})$. The input argument indicates which function is meant.

A.1.1 Matching Networks

Let $\boldsymbol{h}^{\text{mol}} := \boldsymbol{h}_{\boldsymbol{w}_1}$ and $\boldsymbol{h}^{\text{mem}} := \boldsymbol{h}_{\boldsymbol{w}_2}$ be the encoder functions for the input molecule $\boldsymbol{m}$ and the support set molecules $\boldsymbol{X}$, respectively.

We denote:

$$\begin{aligned}\boldsymbol{m}^0 :=& \boldsymbol{h}^{\text{mol}}(\boldsymbol{m}),\\ \boldsymbol{H}^0 :=& \boldsymbol{h}^{\text{mem}}(\boldsymbol{X}),\end{aligned}$$

where $\boldsymbol{h}^{\text{mem}}$ is applied element-wise to all molecules stored in $\boldsymbol{X}$.

Furthermore, we define:

$$\begin{aligned}\boldsymbol{h}^{\text{assoc},1} &: \left(\boldsymbol{m}^0, \boldsymbol{H}^0\right) \mapsto (\boldsymbol{m}^*)\\ &\boldsymbol{m}^* = \text{attLSTM}_L\left(\boldsymbol{m}^0, \boldsymbol{H}^0\right).\end{aligned}$$

Here, $\boldsymbol{m}^*$ represents the updated representations of the input molecule. attLSTM denotes the attention-enhanced LSTM network (Vinyals et al., 2015, 2016). The hyperparameter $L \in \mathbb{N}$ defines the fixed number of unrolling steps of the attLSTM.

We introduce:

$$\begin{aligned}\boldsymbol{h}^{\text{assoc},2} &: \left(\boldsymbol{m}^*, \boldsymbol{H}^0\right) \mapsto \boldsymbol{a}\\ &\boldsymbol{a} = \text{softmax}\left(\boldsymbol{k}\left(\boldsymbol{m}^*, \boldsymbol{H}^0\right)\right).\end{aligned}$$

For the computation of the attention values $\boldsymbol{a}$, the softmax function and the cosine similarity function $\boldsymbol{k}$ is used. These attention values are used for the weighted sum over the support set labels to make the final prediction $\hat{y}$:

$$\begin{aligned}\boldsymbol{h}^{\text{assoc},3} &: (\boldsymbol{y}, \boldsymbol{a}) \mapsto \hat{y} \in \mathbb{R}\\ &\hat{y} = \sum_{i=1}^{N} a_i \cdot y_i.\end{aligned}$$

We combine $\boldsymbol{h}^{\text{assoc},1}$, $\boldsymbol{h}^{\text{assoc},2}$ and $\boldsymbol{h}^{\text{assoc},3}$ and name the resulting function $\boldsymbol{h}^{\text{assoc}}$.

A.1.2 IterRefLSTM

This approach is connected and thus similar to Matching Networks (Section A.1.1). Altae-Tran et al. (2017) replaced the attLSTM by their IterRefLSTM to be able to also update the support set molecule embeddings.

$\boldsymbol{h}^{\text{mol}} := \boldsymbol{h}_{\boldsymbol{w}_1}$ and $\boldsymbol{h}^{\text{mem}} := \boldsymbol{h}_{\boldsymbol{w}_2}$ are GNNs which work as encoder functions for the input molecule $\boldsymbol{m}$ and the support set molecules $\boldsymbol{X}$.

We denote:

$$\begin{aligned}\boldsymbol{m}^0 :=& \boldsymbol{h}^{\text{mol}}(\boldsymbol{m}),\\ \boldsymbol{H}^0 :=& \boldsymbol{h}^{\text{mem}}(\boldsymbol{X}),\end{aligned}$$

where $\boldsymbol{h}^{\text{mem}}$ is applied element-wise to all molecules stored in $\boldsymbol{X}$.

Furthermore, we define:

$$\begin{aligned}\boldsymbol{h}^{\text{assoc},1} &: \left(\boldsymbol{m}^0, \boldsymbol{H}^0\right) \mapsto (\boldsymbol{H}^*, \boldsymbol{m}^*)\\ &(\boldsymbol{H}^*, \boldsymbol{m}^*) = \text{IterRefLSTM}_L\left(\boldsymbol{m}^0, \boldsymbol{H}^0\right).\end{aligned}$$

Here, $\boldsymbol{m}^*$ and $\boldsymbol{H}^*$ contain the updated representations for the input molecule and the support set molecules. The IterRefLSTM denotes the function which updates these representations. For details, we refer to Altae-Tran et al. (2017). The hyperparameter $L \in \mathbb{N}$ controls the number of iteration steps of the IterRefLSTM.

$$\begin{aligned}\boldsymbol{h}^{\text{assoc},2} &: (\boldsymbol{m}^*, \boldsymbol{H}^*) \mapsto \boldsymbol{a} \\ \boldsymbol{a} &= \text{softmax}\left(\boldsymbol{k}\left(\boldsymbol{m}^*, \boldsymbol{H}^*\right)\right).\end{aligned}$$

For the computation of the attention values $\boldsymbol{a}$, the softmax function used. $\boldsymbol{k}$ is a similarity metric. Then, the attention values are used for the weighted sum over the support set labels to make the final prediction $\hat{y}$:

$$\begin{aligned}\boldsymbol{h}^{\text{assoc},3} &: (\boldsymbol{y}, \boldsymbol{a}) \mapsto \hat{y} \in \mathbb{R} \\ \hat{y} &= \sum_{i=1}^{N} a_i \cdot y_i.\end{aligned}$$

We combine $\boldsymbol{h}^{\text{assoc},1}$, $\boldsymbol{h}^{\text{assoc},2}$ and $\boldsymbol{h}^{\text{assoc},3}$ and name the resulting function $\boldsymbol{h}^{\text{assoc}}$.

A.1.3 Prototypical Networks

We define a suitable embedding function $\boldsymbol{h}_{\boldsymbol{w}}$ with learnable parameters $\boldsymbol{w}$:

$$\boldsymbol{h}^{\text{mol}} \equiv \boldsymbol{h}_{\boldsymbol{w}}.$$

We consider the subsets of the support set $\boldsymbol{Z} = \{(\boldsymbol{x}, y) | \boldsymbol{x} \in \boldsymbol{X}, y \in \boldsymbol{y}\}$:

$$\begin{aligned}\boldsymbol{Z}^+ &:= \{(\boldsymbol{x}, y) \in \boldsymbol{Z} | y = 1\}, \\ \boldsymbol{Z}^- &:= \{(\boldsymbol{x}, y) \in \boldsymbol{Z} | y = -1\}.\end{aligned}$$

and define:

$$\begin{aligned}\boldsymbol{h}^{\text{mem}} &: \boldsymbol{Z} \mapsto (\boldsymbol{r}^+, \boldsymbol{r}^-) \\ \boldsymbol{r}^+ &= \frac{1}{|\boldsymbol{Z}^+|} \cdot \sum_{(\boldsymbol{x},y) \in \boldsymbol{Z}^+} \boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{x}) \\ \boldsymbol{r}^- &= \frac{1}{|\boldsymbol{Z}^-|} \cdot \sum_{(\boldsymbol{x},y) \in \boldsymbol{Z}^-} \boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{x}),\end{aligned}$$

where $\boldsymbol{r}^+$ and $\boldsymbol{r}^-$ are the prototypical representations of the active and inactive molecules in the support set. The core of the memory encoder function $\boldsymbol{h}^{\text{mem}}$ is the embedding function $\boldsymbol{h}_{\boldsymbol{w}}$, which means that $\boldsymbol{h}^{\text{mol}}$ and $\boldsymbol{h}^{\text{mem}}$ use weight sharing. Let $\boldsymbol{d}$ be a distance metric. We define:

$$\begin{aligned}\boldsymbol{h}^{\text{assoc}} &: \left(\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{m}), \boldsymbol{r}^+, \boldsymbol{r}^-\right) \mapsto \hat{y} \in \mathbb{R} \\ \hat{y} &= \frac{\exp(-\boldsymbol{d}(\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{m}), \boldsymbol{r}^+))}{\exp(-\boldsymbol{d}(\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{m}), \boldsymbol{r}^+)) + \exp(-\boldsymbol{d}(\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{m}), \boldsymbol{r}^-))},\end{aligned}$$

where $\hat{y}$ is the prediction for the input molecule $\boldsymbol{m}$ and exp the exponential function.

A.2 Details on the dataset

We use the benchmark dataset FS-Mol (Stanley et al., 2021), which was originally extracted from ChEMBL27 (Mendez et al., 2019). In total, 489,133 measurements, 233,786 compounds and 5,120 tasks are available. Per task, the mean number of data points is 94. The dataset is well balanced, which means that, per task, the mean ratio of active molecules and inactive molecules is 1.

Data split

The FS-Mol benchmark dataset already includes a training, validation and test split, guaranteeing disjoint task sets (Stanley et al., 2021). We use this data split and therefore ensure a fair method comparison with the methods included in FS-Mol.

Hyperparameter	Explored values
Number of hidden layers	1, **2**, 4
Number of units per hidden layer	**1024**, 4096
Output dimension	**512**, 1024
Activation function	ReLU, **SELU**
Learning rate	0.0001, **0.001**, 0.01
Optimizer	**Adam**
Weight decay	**0**, $1 \cdot 10^{-4}$
Mini-batch size	**4096**
Input Dropout	**0.1**
Dropout	**0.5**
Layer-normalization	False, **True**
- Affine	**False**
Similarity function	cosine similarity, **dot product**, MinMax similarity

Table 2: Hyperparameter space considered for model selection. The hyperparameters of the best configuration are marked bold.

Descriptors

Stanley et al. (2021) precomputed extended connectivity fingerprints (ECFP) (Rogers and Hahn, 2010) and key molecular physical descriptors, which were defined by RDKit. We use these descriptors as inputs for our methods. Therefore, no additional, external information about molecules or tasks was added.

A.3 Neural Similarity Search: Details on architectures and hyperparameters

The core of our implementation is a descriptor-based fully-connected Deep Neural Network with scaled exponential units (Klambauer et al., 2017), which is used as the molecule encoder and the memory encoder in a Siamese Network fashion. Then, the representations for the input molecule and the support set molecules are transformed by the Layer Normalization (Ba et al., 2016) operation and associated via the dot product. Finally, the raw prediction value is scaled between zero and one by the sigmoid function $\boldsymbol{\sigma}$. We trained the networks using the Adam optimizer (Kingma and Ba, 2014) to minimize binary cross-entropy loss.

Hyperparameter search

We performed manual hyperparameter search on the validation set. We report the explored hyperparameter space (Table 2). Bold values indicate the selected hyperparameters for the final model.

Association functions

Searching for suitable hyperparameters for the final Neural Similarity Search architecture, we explored different choices for similarity functions within the association function $\boldsymbol{h}^{\text{assoc}}$, which are the cosine similarity, the dot product and a function called MinMax similarity (Ralaivola et al., 2005). In case of dot product as similarity measure and with shared encoder, Neural Similarity Search could also be considered as a Prototypical Network (see above).

The MinMax similarity measure is a generalization of the Tanimoto similarity to continuous values (Ralaivola et al., 2005; Li, 2016) given by:

$$k(\boldsymbol{u}, \boldsymbol{v}) = \frac{\sum_{i=1}^{N} \min(u_i, v_i)}{\sum_{i=1}^{N} \max(u_i, v_i)}.$$

A.4 Additional results

To assess the variability of the method across different support sets and re-runs, we both selected multiple support sets (see Table 3) and we trained the model with five different seeds (see Table 4).

During test time, five different seeds are used to draw different support sets. For this experiment, we randomly chose a trained model for the classic Similarity Search as well as for the Neural Similarity Search variant.

Table 3: Results for different support sets reported for all tasks [ΔAUC-PR]

Method	Mean±Std	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
Similarity Search (this work)	.118 ± .003	.118	.122	.113	.121	.118
Neural Similarity Search (this work)	.222 ± .003	.223	.221	.221	.217	.227

The classic Similarity Search method does not include trainable parameters. Therefore, variability across training re-runs are just reported for the Neural Similarity Search method.

Table 4: Results for different training runs reported for all tasks [ΔAUC-PR]

Method	Mean±Std	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
Neural Similarity Search (this work)	.220 ± .005	.223	.219	.211	.224	.221