JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

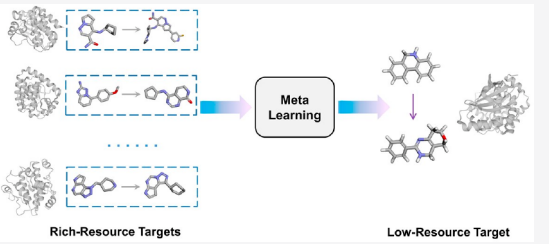
Meta Learning for Low-Resource Molecular Optimization

Jiahao Wang, Shuangjia Zheng, Jianwen Chen, and Yuedong Yang*

Cite This: https://doi.org/10.1021/acs.jcim.0c01416 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The goal of molecular optimization (MO) is to discover molecules that acquire improved pharmaceutical properties over a known starting molecule. Despite many recent successes of new approaches for MO, these methods were typically developed for particular properties with rich annotated training examples. Thus, these approaches are difficult to implement in real scenes where only a small amount of pharmaceutical data is usually available due to the expense and significant effort required for the data collection. Here, we propose a new approach, Meta-MO, for molecular optimization with a handful of training samples based on the well-recognized first-order meta-learning algorithms. By using a set of meta tasks with rich training samples, Meta-MO trains a meta model through the meta-learning optimization and adapts the learned model to new low-resource MO tasks. Meta-MO was shown to consistently outperform several pretraining and multitask training procedures, providing an average improvement in the success rate of 4.3% on a large-scale bioactivity data set with diverse target variations. We also observed that Meta-MO resulted in the best performing models across fine-tuning sets with only dozens of samples. To the best of our knowledge, this is the first study to apply meta learning to MO tasks. More importantly, such a strategy could be further extended to many low-resource scenarios in real-world drug design.



INTRODUCTION

Designing molecules with desirable functionalities lies at the core of the drug discovery process. However, it is practically impossible to discover a new drug through brute-force exploration of the chemical space that contains more than 10^{23} to 10^{26} druglike molecules.[1] A common strategy of rational molecular design is to narrow down the searching space by starting from a known molecule with potential. Thus, the molecular design turns to optimize a candidate molecule by finding novel molecules with improved pharmaceutical activities and reduced side effects to patients, i.e., the molecule optimization (MO).[2] Yet, it remains challenging to form satisfactory optimization for hit compounds since there is usually only a small amount of pharmaceutical data for the source and target molecules.[3]

Conventional computational approaches utilized hand-crafted rules such as matched molecular pair analysis (MMPA)[4] or reaction-based fragment growth[5,6] that often generate compounds with limited novelty and diversity. In addition, these methods often require a substantial amount of data engineering work. With the development of deep learning techniques, various neural approaches[7−10] have been proposed to design novel, natural, and diverse molecules with desired properties through reinforcement learning or Bayesian Optimization (BO). However, these *de novo* design methods tend to generate molecules from scratch, taking no account of source molecules and synthetic feasibility, and hence are usually impractical in the context of MO tasks.

Recently, several studies[11,12] attempted to use a supervised learning paradigm for the transformation of source and target compounds with desired properties, where MO was reformulated into a graph-to-graph translation problem. The original idea is to translate the source compound into the target compound with prespecified desired properties.[13] To train these graph-based methods, they have simulated tens of thousands of molecules (data points) through expert-defined scoring functions from deterministic and rigid rules like calculated logP or druglike index (QED) that can be calculated directly from the compound structure. However, under the large number of training points, these well predefined scoring functions could be well fit even through a basic seq-to-seq model[14] and cannot represent the challenge in real scenes of MO where the scoring function needs to train on only a few potential compounds (points) for a new target protein. Therefore, it is necessary to develop a new MO framework that could be trained from a small number of data points.

The problem of modeling a small number of training points has been addressed fiercely by the few-shot learning community. A remarkable solution is the meta-learning paradigm to obtain a meta model that is efficient at adapting

Received: December 8, 2020

to new tasks.[15−17] Under this paradigm, Vinyals et al. proposed the Matching Networks[18] as the first pioneering approach using metric learning and augmented neural networks to learn a meta model that could be rapidly adapted to limited data. Another related approach is the Model Agnostic Meta-Learning (MAML) algorithm,[19] which has been particularly successful at producing state-of-the-art results on few-shot representation learning benchmarks. The key idea of MAML is to learn good initialization parameters from a set of resource-rich tasks for easy fine-tuning on low-resource scenarios. Particularly, MAML is able to be applied to a wide range of areas because it is model-agnostic and thus introduces no extra neural architectures or parameters. However, MAML needs to calculate second-order derivatives through the optimization process and is computationally expensive when performing a large number of gradient steps. To address this shortcoming, Nichol et al.[20] proposed a quasi MAML framework, Reptile, by computing only first-order gradients through the optimization process, making it more suitable for optimization problems. Though these meta-learning techniques have recently been shown superior in many fields[21] including low-resource molecular property predictions,[3,22] no effort has been given to the task of MO due to the inherent complexity of the generation task.

In this work, we adopted the standard meta-learning paradigm to the MO problem through the Reptile technique. By viewing bioactivity data points in different protein targets as separate MO tasks, our method Meta-MO learns a meta-model from a series of resource-rich targets that are able to generalize for other protein targets. Such a meta-model allows for the optimization of candidate molecules for low-resource protein targets (e.g., with only tens of known actives).

The main contribution of this paper is 2-fold:

- We proposed a meta-learning algorithm Meta-MO based on Reptile for low-resource MO tasks. Since Meta-MO is model-agnostic, it is applicable to many other MO models. To the best of our knowledge, this is the first study of applying meta learning to MO tasks.
- We constructed a real-world bioactive MO data set and utilized the data set for extensive evaluations with various settings. As a result, Meta-MO was shown to significantly outperform other methods in multiple configurations. Further analyses indicated that the superior performance of Meta-MO resulted from much faster and better adaption to new tasks.

METHODS

Data Preprocessing. To assess our molecule optimization (MO) method, we constructed sets of MO pairs from a subset of ChEMBL20[23] using a custom-made scoring function for training and large-scale evaluations. More specifically, we first selected kinase-related targets with 300−5000 unique bioactivity instances. After filtering out the SMILES strings containing disconnected ions or fragments, molecule compounds were then preprocessed using RDKit[24] for salt and isotopes removal, as well as charge neutralization. The final data set contained 103 511 bioactivity compounds across 152 kinases. Note that we used pChEMBL values as the standard activity unit defined as −Log(molar EC50, XC50, IC50, AC50, K_i, K_d, or Potency).

To mimic the MO scenario, we constructed our data set following the idea of the matched molecular pair (MMP) cutting algorithm proposed by Hussain et al.[4] In particular, we constructed data sets by sampling target-based molecular pairs ((X; Y)|Z) where the source molecule X and target molecule Y need to be similar in 2D ($2Dsim(X; Y) \geqslant 0.4$), and Y has significant bioactivity improvement over X (pChEMBL ≥ 1) in the context of protein kinase Z. Here, the 2D molecular similarity was measured by the Tanimoto coefficient over the Morgan fingerprints.[25] The source molecules were restricted to those with low bioactivity (pChEMBL ≤ 6). To avoid potential bias, the bioactive molecules for each protein kinase were randomly split into training, validation, and test by 6:2:2, and the molecules for training were then paired under the similarity and bioactivity improvement constraints to construct the final training molecule pairs set. During the pairing, each molecule was limited to act as a source molecule or a target molecule for at most five times to avoid possible biases caused by frequently appearing molecules. One protein kinase was considered as a task. After removing tasks containing less than 300 training molecular pairs, 64 tasks remained for the experiments.

These 64 tasks were further divided into 57:1:6 for the training, validation, and test task subsets ($\mathcal{T}^{train}$, $\mathcal{T}^{val}$, and $\mathcal{T}^{test}$), respectively. The 57 training tasks in $\mathcal{T}^{train}$ are defined as support tasks for meta training, containing around 33K molecular pairs. The rest of the seven tasks from $\mathcal{T}^{val}$ and $\mathcal{T}^{test}$ contain at least 1000 molecule pairs to ensure that deep QSAR models could be generated to qualify generated molecules. Table 1 shows the data set split of $\mathcal{T}^{test}$ and $\mathcal{T}^{val}$ tasks (see the Supporting Information for $\mathcal{T}^{train}$). Figures S1−S4 show the distributions of molecule weight, synthetic accessibility score, logP, and property of $\mathcal{T}^{test}$ tasks.

Table 1. Target Names and the Numbers of Compounds or Compound Pairs in the Training, Validation, and Test Sets for Seven Tasks from the $\mathcal{T}^{val}$ and $\mathcal{T}^{test}$ Task Subsets[a]

CHEMBL ID	target name	code	train	validation	test
CHEMBL262	glycogen synthase kinase-3 beta	GSK-3B	1130	210	210
CHEMBL267	tyrosine-protein kinase SRC	SRC	1756	250	250
CHEMBL3267	PI3-kinase p110-gamma subunit	P110γ	1078	136	136
CHEMBL3650	fibroblast growth factor receptor 1	FGFR1	1280	164	164
CHEMBL4005	PI3-kinase p110-alpha subunit	P110α	2156	222	223
CHEMBL4282	serine/threonine-protein kinase AKT	AKT	1364	161	161
CHEMBL4722	serine/threonine-protein kinase Aurora-A	AURKA	1060	146	147

[a]Serine/threonine-protein kinase Aurora-A (CHEMBL4722) is selected as the $\mathcal{T}^{val}$ task.

Meta-MO. Meta-MO was designed for a specific MO task having a small number of training samples but having many relevant tasks totally with a substantial number of samples. Here, we first employed the tasks in $\mathcal{T}^{train}$ to train a meta-model and then used the trained meta-model to learn an effective agent for the target task $\mathcal{T}^{test}$. Generally, we aimed to

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

harness the information available in the training tasks to create robust generators for the test system.

In the following section, we will briefly introduce the fundamentals of MAML and Reptile to help readers understand the framework. For more details, refer to the original literature.[19,20]

Meta Learning. Optimization-based meta learning aims at training a meta-model based on one set of similar tasks $\mathcal{T}^{train}$ such that the meta-model can quickly adapt to a new task with limited training samples or iterations. To accomplish this, MAML proposed to sample gradient-based learning rules from the distribution of tasks and fine-tune the rules on new tasks.[19] Despite the inspiring performance reported, MAML requires second-order derivatives through the optimization process, which is computationally expensive when performing a large number of gradient steps. In this study, Reptile, one of the first-order approximation algorithms of MAML that can provide both conveniences in implementation and efficiency in computation, was adopted as the meta-learning algorithm for our Meta-MO framework.

Algorithm 1 Reptile (Training)

1: Initialize model parameters θ
2: For iteration=1, 2, …, M do
3: Sample a support task τ
4: Inner-update: Update θ_τ, k steps of Adam from θ
5: Outer-update: Update $\theta \leftarrow \theta + \epsilon(\theta_\tau - \theta)$
6: End

As shown in Algorithm 1, the meta-model was initialized by random model parameters θ. In each iteration, a task τ was randomly selected from $\mathcal{T}^{train}$ as the support task, and the model was inner-updated through k steps of Adam optimization. Based on the final parameters θ_τ, the meta-model was outer-updated by partially moving toward $\theta\tau$ with ϵ as the learning rate for the outer update. These continue until the preset number of iterations, and the best meta-model was selected based on the validation task set $\mathcal{T}^{val}$.

Figure 1 displays the overall architecture of the Reptile meta-learning framework for molecular optimization. Each support task $\mathcal{T}_i^{train}$, $i \in 0, ..., N-1$ indexes the source molecules train-X and the corresponding target molecules train-Y. During the meta training stage, parameters are inner-updated based on the sampled task at each iteration (line 4 in Algorithm 1) and then outer-updated as the initialized parameters for the next iteration (line 5 in Algorithm 1). During the meta testing on a query task $\mathcal{T}^{test}$, the learned meta-model is fine-tuned through the training pair set among the task. The fine-tuned model is selected and then evaluated through the validation and test pair sets in the query task, respectively.

Neural Networks for Molecular Optimization. Although the meta learning is assumed to be model-independent and insensitive to the selected model, an efficient and effective basic MO model is required to aid the verification during training and testing. Previous graph-based MO models are criticized for fragile training processes and limited chemical diversity. For this reason, we adopted a graph enhanced Transformer (GET) for generating optimized molecules with the input of a source molecule.[26−29]

As shown in Figure 2, GET comprises two main components: a graph neural network (GNN) for molecular graph embedding and a Transformer neural network for mapping the molecular pairs.

For a given source molecule, we first represent it as an attributed molecular graph $G = (V, E)$, where $V = \{v_1,...,v_n\}$ denotes a set of nodes (atoms), and $E = [e_{ij}]_{i,j=1}^{n}$ represents edges (bonds) between atoms i and j. Each atom v_i is represented by a d-dimensional initial feature vector h_i containing the 2D chemical features, and each bond is represented by an f-dimension vector e_{ij}, encoding the bond types including single, double, triple, and aromatic types (see more details in Table S1). The graph encoder generates new representations $\hat{H}_s = \{h'_1,...,h'_N\}$ for atoms of the source molecule through multilayer message passing networks.

Then, we concatenate the learned graph representations $\hat{H}_s$ with SMILES sequence embedding $M_s = (s_1,\cdots, s_m)$ and convert them through a simple linear transformation. The combined multimodal molecular representations are sent to the Transformer encoder to convert into a latent representation $L \in \mathbb{R}^{m\times f}$, where m is the sequence length of molecular SMILES, and f is the hidden state dimension. Given L, the decoder iteratively generates an output SMILES sequence $Y = (y_1,...,y_o)$ until the ending token "⟨/s⟩" is generated.

The Transformer neural networks contain multiple encoder-decoder modules. Each encoder layer consists of a multihead

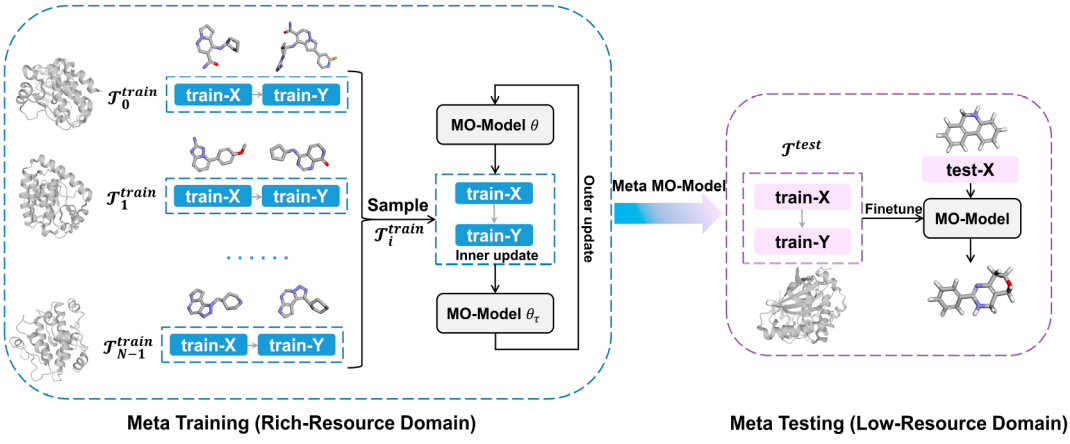


Figure 1. Framework of Reptile meta learning for molecular optimization. The left part aims to train a meta-model by sampling tasks from training tasks (detailed in Algorithm 1). The trained meta-model will be fine-tuned and tested on target tasks (the right part).

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

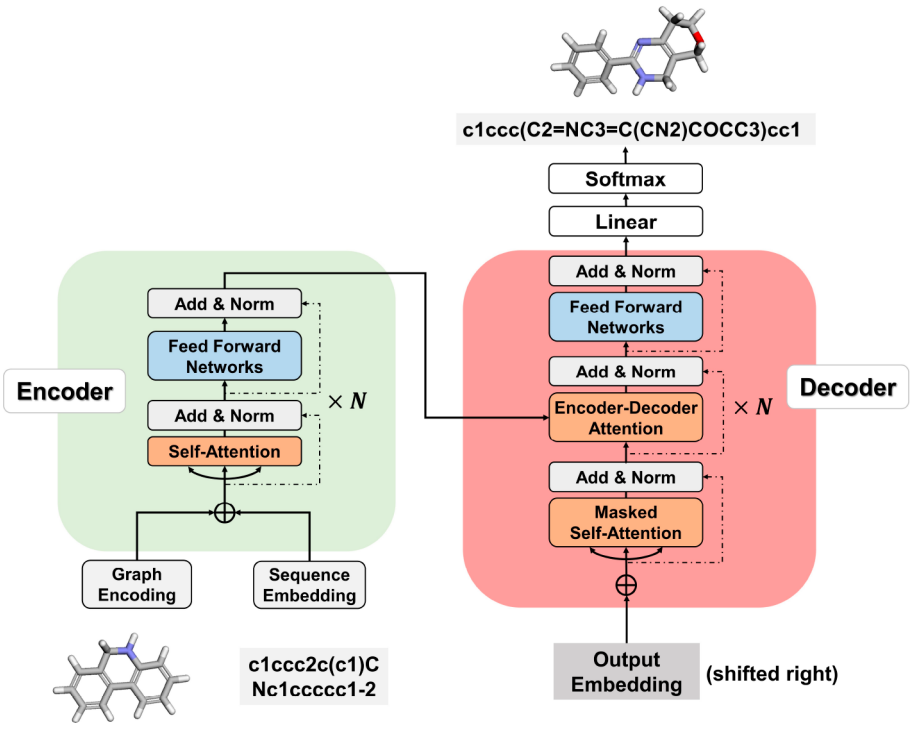


Figure 2. Overall architecture of a graph enhanced Transformer.

self-attention sublayer and a position-wise feed-forward network (FFN) sublayer. Multihead attention has several scaled dot-product attention functions working in parallel, which allows the model to focus on messages from different subspaces at different positions. More formula derivations of the graph neural network and Transformer neural network have been demonstrated in the Supporting Information.

During training, the model minimizes the cross-entropy loss between the target sequence $M_t = (t_1,...,t_k)$ and the output sequence Y.

$$\mathcal{L}(Y, M) = -\sum_{i=1}^{k} y_i \log t_i$$

Additionally, to assess our method, we also employed two popular molecular transformation methods for comparison since there are many successful examples to combine meta learning with LSTM or GCN techniques.[21]

1. **LSTM-Attention**. The seq2seq LSTM-Attention model consists of a bidirectional LSTM encoder and an LSTM decoder,[14] with an attention mechanism adopted to generate target molecules within the context information of source molecules.
2. **Graph2graph (G2G)**. G2G[11] is a Graph-to-Graph model that extends the junction variational autoencoder (VAE) via an attention mechanism and generative adversarial networks (GAN). The model is capable of translating the current molecule to a similar molecule with a predefined desired property (e.g., logP).

Baseline Training Strategies. We included different learning strategies as baselines:

a) **No pretraining**. We remove the pretraining procedure as the "no pretraining" baseline, where model f_θ is randomly initialized and learned only through the training data from each $\mathcal{T}^{test}$ task. The model is used to indicate how much the below models can benefit from the pretraining procedures.

b) **Multi-MO**. Multi-MO follows a multitask pretraining paradigm. We train f_θ using multitask learning (MTL) with $\mathcal{T}^{train}$ tasks data and then fine-tune it based on the training data in the $\mathcal{T}^{test}$ task. It is the major baseline to compare with and to prove the effectiveness of task adaption ability of meta learning.

c) **Multi-MO-Zero-Shot**. Multi-MO-zero-shot corresponds to a zero-shot learning scenario. We train f_θ using multitask learning (MTL) with $\mathcal{T}^{train}$ tasks data and directly test the model on $\mathcal{T}^{test}$ tasks without fine-tuning steps.

d) **Meta-MO-Zero-Shot**. Meta-MO-zero-shot corresponds to a zero-shot learning scenario as Multi-MO-zero-shot does. We train f_θ using meta learning instead of MTL with $\mathcal{T}^{train}$ tasks data and test the model on $\mathcal{T}^{test}$ tasks without fine-tuning steps. Compared with Multi-MO-zero-shot, the results directly indicate how well the learned meta model can perform on a new test task.

Evaluation Metrics. We adopted the following five metrics for evaluation partly following Jin et al.:[11]

- **Similarity**. The similarity (range 0−1) between an input molecule and the output molecule is calculated by Tanimoto similarity over Morgan fingerprints.[25]
- **Success Rate (SR)**. The success rate is the core metric as the percentage of successfully optimized molecules out of all designed molecules in the test set. A design was considered to be successful if the output molecule Y has a 2D similarity ≥ 0.4 to input molecule X and bioactivity improvement (pBioactivity(Y) − pBioactivity(X) ≥ 1.0). Note that the bioactivities of generated compounds are predicted by QSAR models

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(see below) as we do not have experimentally measured bioactivity for generated molecules.

- **Uniqueness**. Uniqueness calculates the percentage of distinct molecules out of all generated molecules. A higher uniqueness score means that the model tends to generate molecules beyond the target molecules in the training data.
- **Diversity**. Diversity is defined as the average difference between the source molecule and the generated valid candidate target molecules. The difference was measured as 1.0 − Tanimoto coefficient.
- **Validity**. Validity measures the percentage of valid SMILES strings out of all generated sequences. A high validity score ensures the generation of enough valid molecules during adaption to a new task and thus partly reflects the task adaption ability of a model.

Experimental Details. *QSAR Model.* One important factor required to assess the performance of MO is whether the generated molecules have improved bioactivity on the desired targets. To enable rapid and accurate profiling of generated molecules, virtual profiling models were trained on all compounds in the test kinase data sets. We evaluated several state-of-the-art messages passing neural networks[30,31] with molecular graphs as the molecular representations and found the Communicative Message Passing Network (CMPNN)[31] obviously outperformed other models with an average R^2 of 0.64 and an RMSE of 0.66 (pChEMBL value) on internal test sets. Thus, the CMPNN model was used as the virtual profiling model in the following studies. The modeling details and results are shown in the Supporting Information. It should be noted that QSAR was not involved in the training of MO methods and only used for assessing the final outputs of different MO methods.

Implementations and Hyperparameters. We implemented Transformer with the same settings as the original model.[26] The input atoms are limited to 13 atoms as listed in Table S1, and the corresponding input dimension of atoms is 21. The initial learning rate of Adam is 0.1 for our MO task. For each source molecule, 10 candidate target molecules are generated using the beam search.[32]

To train the meta-model, we randomly sampled tasks from $\mathcal{T}^{train}$ for 57 × 50 = 2850 times. For each sampled task, we trained the inner model for 10 epochs using the training data from the task. The learned model was used to update the meta-model through the outer parameter update with a learning rate of 0.0005. The meta-models were saved as checkpoints every 57 times of task sampling, and the 500 checkpoint models were separately assessed over $\mathcal{T}^{val}$ to select the best meta-model.

For the assessment on $\mathcal{T}^{val}$, each checkpoint model was fine-tuned for 50 epochs using the training set and then validated by the validation set. The model with the highest success rate during the validation was tested on the test set, and the tested result was used to measure the checkpoint model. Finally, the checkpoint model with the best performance was selected as the meta-model for tested tasks.

For each task $\mathcal{T}^{test}$, the selected meta-model was fine-tuned, validated, and tested on the test task following the same steps as the assessment on $\mathcal{T}^{val}$, and the results on the test set represent the performance of the test task.

For a fair comparison, Multi-MO was performed with a similar procedure. Since the pretraining was made on all 57 tasks, we also pretrained the model for 500 epochs costing approximately the same training time.

RESULTS

Model Performance on the Multikinase Data Set. We have collected three basic models popular for molecular generation. To select appropriate models for our MO task, we have implemented these basic models in the multitask MO scenario to avoid potential bias to the meta-learning method. As shown in Table 2, the graph enhanced Transformer (GET) achieved not only the highest success rates and diversity scores but also the most important metrics for the MO task. By comparison, G2G has higher similarity and validity scores because the graph-based model is able to generate similar junction tree structures over source molecules, and the generated molecule graph is always a valid molecule leading to a validity score of 100%. However, it achieved significantly lower uniqueness and diversity due to the limited vocabulary size, consistent with previous findings.[33] The LSTM-Attention model achieved consistently worse results than the GET model except for the uniqueness score. Therefore, we chose GET as the backbone model for further experiments.

Table 2. Multitask Molecular Optimization Performance

model	similarity	success rate	uniqueness	diversity	validity
LSTM-Attention	0.1487	18.05%	**75.35%**	0.2001	58.44%
G2G	**0.3951**	30.67%	23.24%	0.1742	**100%**
GET	0.2647	**39.43%**	61.80%	**0.2649**	64.28%

Performance Comparisons with Different Training Strategies. Based on the GET backbone model, we compared the effects of different model parameters or pretraining strategies through three different experiments. As shown in Table 3, without fine-tuning on the test tasks, Meta-MO-zero-shot outperformed Multi-MO-zero-shot in five out of six test tasks and improved the average success rate by 2.46%. Interestingly, these two methods even achieved higher success rates on glycogen synthase kinase-3 beta than the "no pretraining" model that was directly trained on the test task. After fine-tuning on the test tasks, Meta-MO and Multi-MO have significantly improved the success rates with 10−25% over the "no pretraining" method across six test tasks. Compared with Multi-MO, Meta-MO could further improve the success rates with 4.37% on average, indicating that the meta-learning paradigm provided better generalization ability than a simple multitask pretraining strategy.

Table 4 detailed the comparisons of Multi-MO and Meta-MO. Meta-MO consistently outperformed Multi-MO on all $\mathcal{T}^{test}$ tasks by success rate, similarity, uniqueness, and validity metrics, indicating the superiority of the meta-learning setting in MO scenario.

To further dive into how meta learning learns at the pretraining stage, we show the learning curves of meta learning and multitask pretraining for the convergence. Figure 3 displays the success rates on the $\mathcal{T}^{val}$ task changed with the pretrained epochs. In the first ∼200 epochs, Meta-MO-zero-shot achieved lower success rates than Multi-MO-zero-shot, likely because the 57 training tasks have not been sufficiently sampled to learn the task distribution for the meta learning. After 200 epochs, meta learning is able to surpass multitask

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 3. Performances of Different Learning Strategies by Success Rates (%) on $\mathcal{T}^{test}$

target	no pretraining	Multi-MO-zero-shot	Meta-MO-zero-shot	Multi-MO	Meta-MO
GSK-3B	14.05 ± 3.37	17.02 ± 1.56	22.38 ± 2.88	28.81 ± 1.04	**37.02** ± 1.40
SRC	24.20 ± 6.96	10.70 ± 1.58	15.00 ± 2.01	44.20 ± 1.18	**49.50** ± 0.52
P110γ	16.73 ± 7.72	8.09 ± 1.80	11.40 ± 1.33	35.48 ± 1.68	**38.42** ± 1.41
FGFR1	37.04 ± 6.57	15.39 ± 0.79	14.93 ± 1.58	54.12 ± 2.67	**58.54** ± 0.75
P110α	50.79 ± 5.45	9.53 ± 0.73	10.65 ± 0.86	66.03 ± 1.50	**70.29** ± 1.11
AKT	31.37 ± 2.96	4.51 ± 1.77	5.59 ± 0.76	41.46 ± 1.11	**42.55** ± 0.70
average	29.03 ± 5.51	10.87 ± 1.37	13.33 ± 1.57	45.02 ± 1.53	**49.39** ± 0.98

Table 4. Detailed Comparison of Multi-MO and Meta-MO on $\mathcal{T}^{test}$ [a]

metric	method	GSK-3B	SRC	P110γ	FGFR1	P110α	AKT	av
success rate (%)	Multi	28.81	44.20	35.48	54.12	66.03	41.46	45.02
	Meta	**37.02**	**49.50**	**38.42**	**58.54**	**70.29**	**42.55**	**49.39**
similarity	Multi	0.21	0.27	0.30	0.33	0.35	0.27	0.29
	Meta	**0.30**	**0.35**	**0.39**	**0.40**	**0.43**	**0.35**	**0.37**
uniqueness (%)	Multi	78.07	65.71	57.92	60.47	57.79	70.09	65.01
	Meta	**82.88**	**66.79**	**60.55**	**63.06**	**59.71**	**74.94**	**67.99**
diversity	Multi	0.27	0.27	0.29	0.30	0.31	0.32	0.29
	Meta	**0.28**	**0.28**	0.29	**0.32**	**0.33**	**0.36**	**0.31**
validity (%)	Multi	60.61	62.80	68.42	72.06	75.43	65.08	67.40
	Meta	**81.95**	**80.46**	**86.25**	**87.04**	**91.01**	**83.14**	**84.98**

[a]The standard deviations are displayed in the Supporting Information.

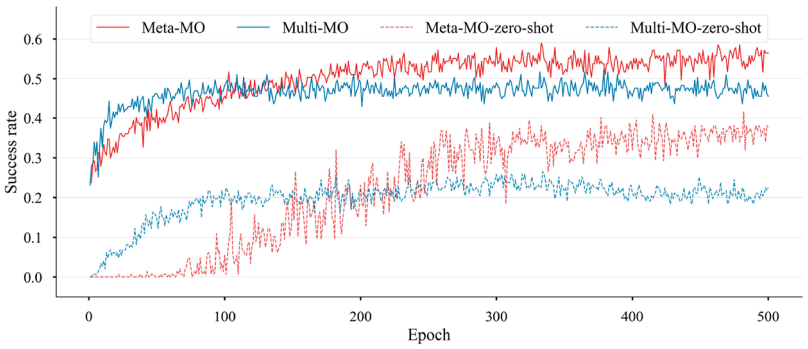


Figure 3. Comparison of learning curves by different strategies on the $\mathcal{T}^{val}$ task.

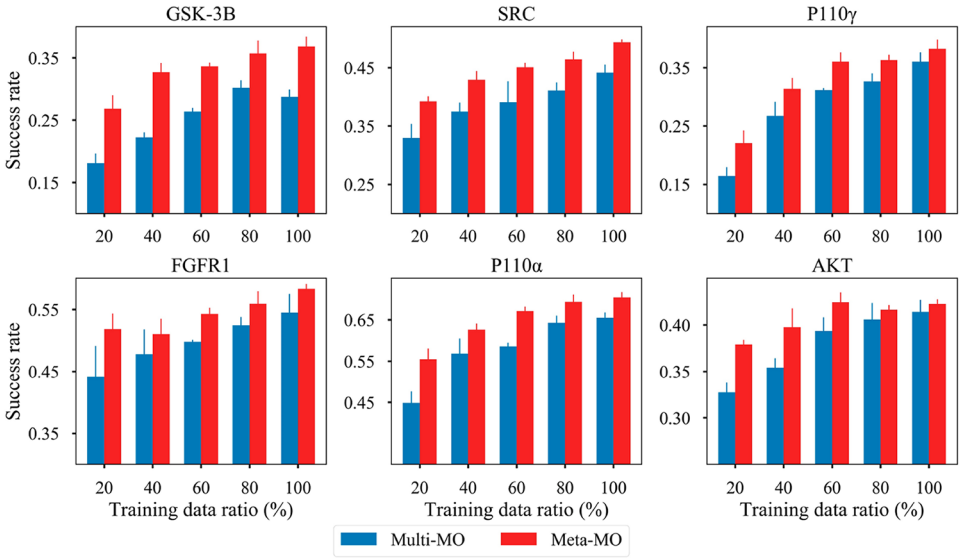


Figure 4. Success rates of two different pretraining strategies with different ratios of training data on $\mathcal{T}^{test}$ tasks.

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

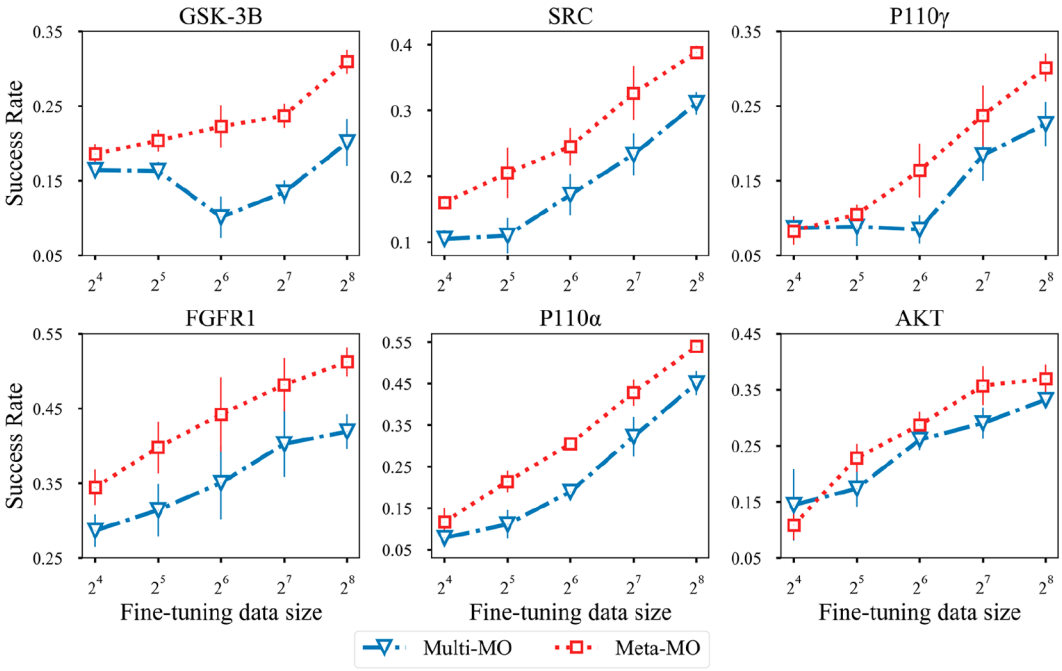


Figure 5. Success rates of two different pretraining strategies after fine-tuning with a different number of training samples ranging from 16 to 256 for $\mathcal{T}^{test}$ tasks.

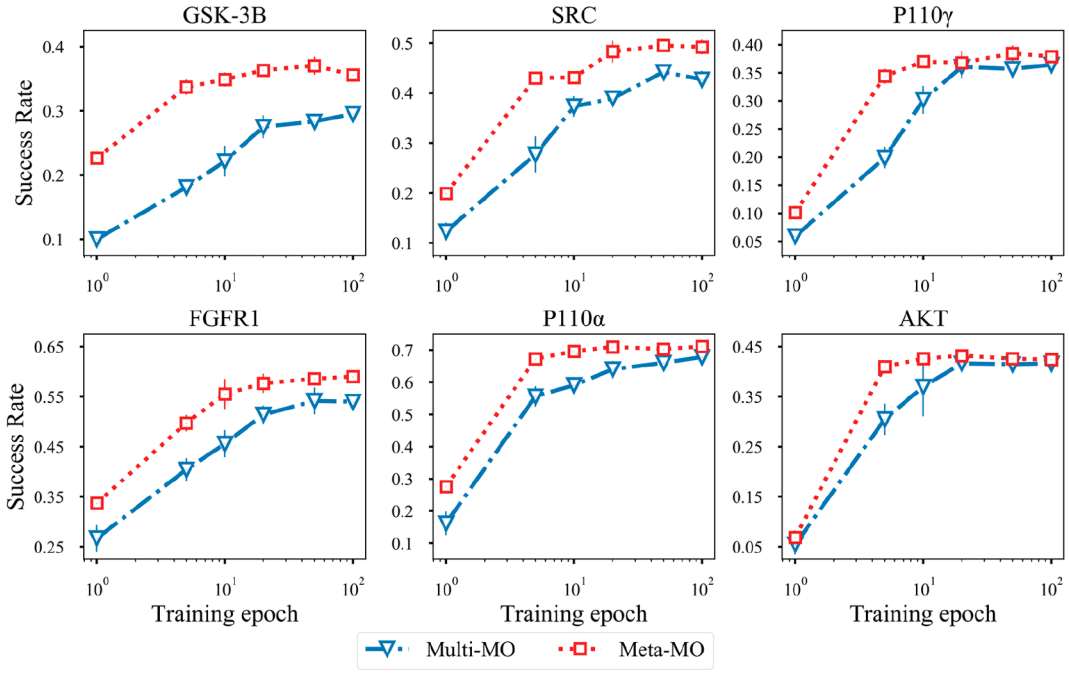


Figure 6. Success rates of two different pretraining strategies with different training epochs on $\mathcal{T}^{test}$ tasks.

pretraining and quickly arrives at a stable performance. By fine-tuning over the query task, there is a similar trend that Meta-MO achieved a lower performance in the early stage, but the surpassing over Multi-MO is earlier at ∼100 epochs. These phenomena explain the design of Reptile to iteratively sample training tasks, for such procedures can help the model to learn the distribution of training tasks so that new tasks can be rapidly and well adapted.

Effect of Fine-Tuning Set Size and Meta Test Training Epochs. In the above evaluation, all the testing sets have sufficient samples (>1000 training pairs, as demonstrated in the Data Preprocessing section). To verify the impact of the training data size, we first reduced the number of training samples on $\mathcal{T}^{test}$ to 20%, 40%, 60%, and 80%, respectively. As shown in Figure 4, the success rates generally increase with more training data, while the meta-learning setting achieves better success rates, especially with limited data (e.g., compare bars of 20% and 100% data ratio for tasks SRC, P110γ, FGFR1, P110α, and AKT).

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

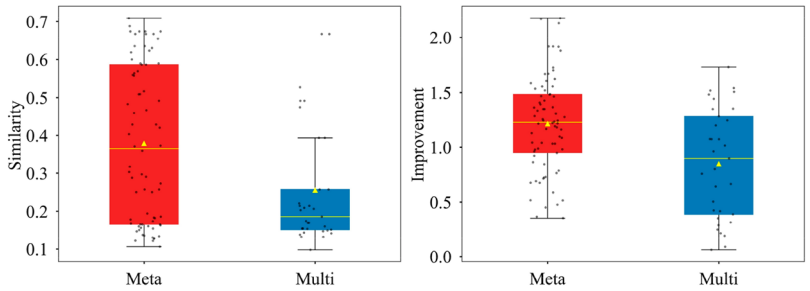


Figure 7. Distributions of 100 valid molecules generated by Meta-MO and Multi-MO over a randomly selected source molecule. Each black point represents a valid molecule.

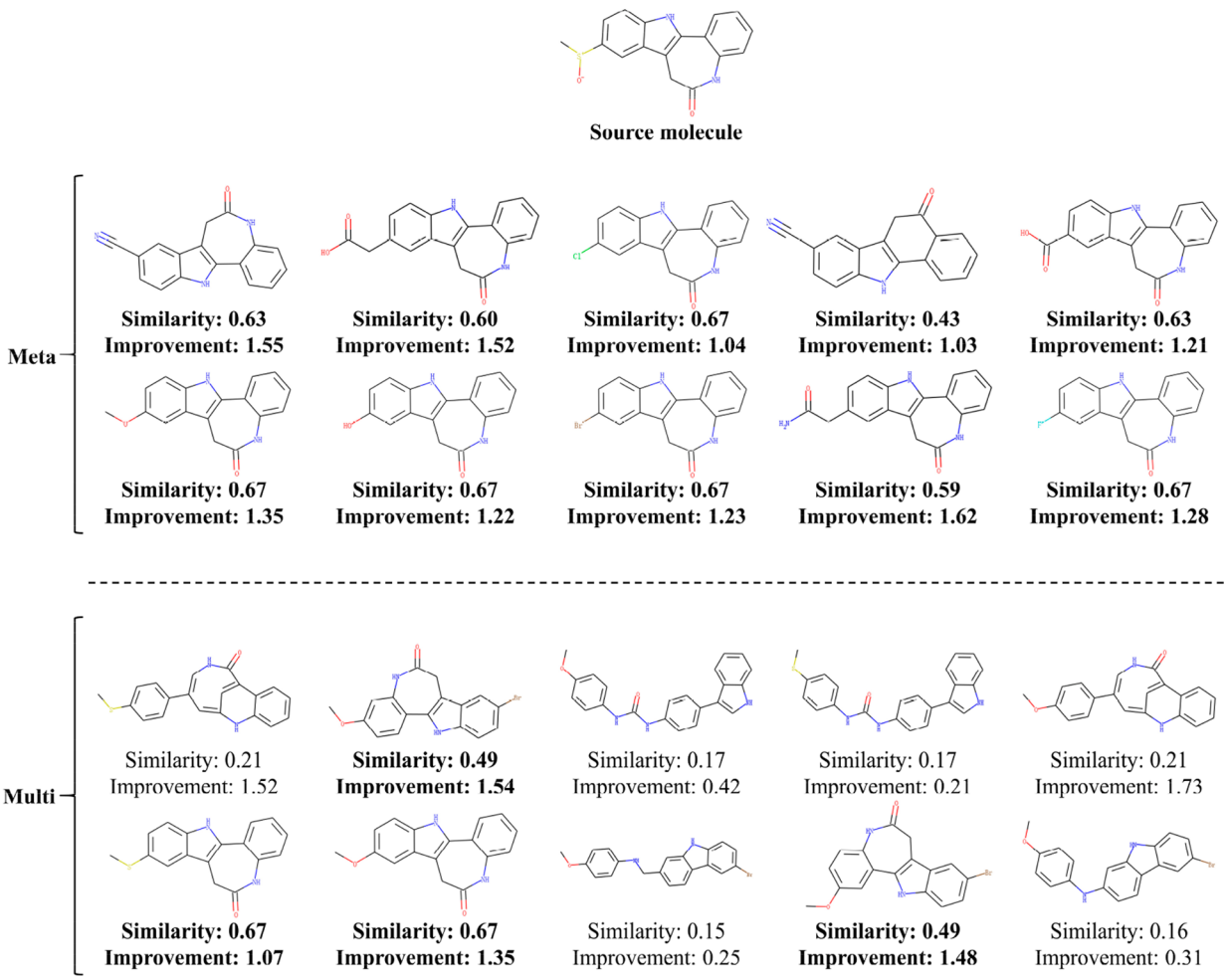


Figure 8. A test source molecule in GSK-3B and the corresponding top 10 target molecules generated by Meta-MO and Multi-MO, respectively. Both models are fine-tuned with only 64 training samples in GSK-3B. Target molecules with bold text descriptions are the molecules satisfying similarity and property improvement constraints.

Figure 5 further displays more challenging experiments with a small number of training samples for each task, ranging from 16 to 256 instances. The results show that meta learning still outperforms multitask pretraining in such an extremely low-resource scenario. We also observed that multi-MO, different from the stable increase by Meta-MO, shows a small increase or drop in small data (16−64) for tasks GSK-3B, SRC, and P110γ. This is likely because Meta-MO has learned a common mode easier for adapting to new tasks. In terms of fine-tuning speed, Meta-MO also shows better results under limited training epochs, as Figure 6 displays. It can be seen as a hint that a meta-learning strategy not only learns better but also faster for task adaptions.

Case Study. To illustrate our methods, we employed glycogen synthase kinase-3 beta (GSK-3B) as an example and generated 100 target molecules through the Meta-MO and Multi-MO models that have been fine-tuned with 64 randomly selected training samples. As shown in Figure 7, Meta-MO has significantly higher similarity and bioactivity improvements than Multi-MO, respectively, with P-values of 0.0028 and 0.0002 by the t test. Figure 8 further shows the top 10 generated compounds for a randomly selected source

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

molecule. Basically, both models are able to generate molecules with required similarity and property improvement. According to the predefined success standard (similarity ≥ 0.4 and bioactivity improvement ≥ 1.0), Meta-MO successfully optimized all molecules with a similarity of 0.43−0.67 and bioactivity improvements of 1.03−1.62. By comparison, Multi-MO successfully optimized only four molecules with a similarity of 0.15−0.67 and bioactivity improvements of 0.21−1.73. On average, molecules generated by Meta-MO are 48.43% and 42.70% higher than those by Multi-MO in 2D similarity and bioactivity improvement.

DISCUSSION AND CONCLUSION

In this work, we propose a meta-learning algorithm Meta-MO based on Reptile for low-resource MO tasks. To the best of our knowledge, this is the first study of applying meta learning to MO tasks. We extensively evaluate Meta-MO on a large bioactivity data set with various low-resource protein targets. Results show that Meta-MO significantly outperforms other optimization methods in various configurations. We further analyze the superior performance of Meta-MO and show that it indeed adapts much faster and better than other models, including multitask learning.

Our model has several weak points. One major problem is the ignoring of target protein information. As molecules interact with the target protein, effective embedding will reduce the limit of the small amount of training samples. The protein information could be embedded through sequence,[34] contact map,[35] or 3D structure.[36] Second, experimental costs limit the discussion about direct comparison of meta learning on recently proposed MO models, which can be covered in future work, and the effectiveness of meta learning on many other directions in low-resource drug discovery is yet to be discovered.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c01416.

Detailed descriptions of graph encoder and Transformer architecture; Table S1, model input representations for atoms; Table S2, R^2, RMSE, and MAE metrics for query task scoring models; Table S3, standard deviations of data in Table 4; and Figures S1−S4, distributions of source molecule weight, synthetic accessibility score, logP score, and bioactivity (PDF)

Data set splits of $\mathcal{T}^{train}$ tasks (XLSX)

AUTHOR INFORMATION

Corresponding Author

Yuedong Yang − *School of Computer Science and Engineering and Key Laboratory of Machine Intelligence and Advanced Computing (MOE), Sun Yat-sen University, Guangzhou 510006, China;* orcid.org/0000-0002-6782-2813; Email: yangyd25@mail.sysu.edu.cn

Authors

Jiahao Wang − *School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China;* orcid.org/0000-0002-4366-5374

Shuangjia Zheng − *School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China; Galixir Technologies (Beijing) Limited, Beijing 100083, China;* orcid.org/0000-0001-9747-4285

Jianwen Chen − *School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China;* orcid.org/0000-0001-7999-2070

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c01416

Author Contributions

J.W. and S.Z. contributed equally. J.W., S.Z., and J.C. contributed the concept and implementation. J.W., S.Z., and Y.Y. wrote the manuscript. All authors contributed to the interpretation of results. All authors reviewed and approved the final manuscript.

Funding

This study has been supported by the National Key R&D Program of China (2020YFB0204803), the National Natural Science Foundation of China (61772566, 62041209, and U1611261), the Guangdong Key Field R&D Plan (2019B020228001 and 2018B010109006), Introducing Innovative and Entrepreneurial Teams (2016ZT06D211), and the Guangzhou S&T Research Plan (202007030010).

Notes

The authors declare no competing financial interest.

The project and data for this article may be accesssed at https://github.com/biomed-AI/Meta-MO.

REFERENCES

(1) Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput.-Aided Mol. Des.* **2013**, *27* (8), 675−679.

(2) Fattori, D.; Squarcia, A.; Bartoli, S. Fragment-based approach to drug lead discovery: overview and advances in various techniques. *Drugs R&D* **2008**, *9* (4), 217−27.

(3) Altae-Tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low data drug discovery with one-shot learning. *ACS Cent. Sci.* **2017**, *3* (4), 283−293.

(4) Hussain, J.; Rea, C. Computationally efficient algorithm to identify matched molecular pairs (MMPs) in large data sets. *J. Chem. Inf. Model.* **2010**, *50* (3), 339−348.

(5) Hartenfeller, M.; Eberle, M.; Meier, P.; Nieto-Oberhuber, C.; Altmann, K. H.; Schneider, G.; Jacoby, E.; Renner, S. A collection of robust organic synthesis reactions for in silico molecule design. *J. Chem. Inf. Model.* **2011**, *51* (12), 3093−8.

(6) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. DOGS: reaction-driven de novo design of bioactive compounds. *PLoS Comput. Biol.* **2012**, *8* (2), No. e1002380.

(7) Ståhl, N.; Falkman, G.; Karlsson, A.; Mathiason, G.; Bostrom, J. Deep reinforcement learning for multiparameter optimization in de novo drug design. *J. Chem. Inf. Model.* **2019**, *59* (7), 3166−3176.

(8) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminf.* **2017**, *9* (1), 48.

(9) Griffiths, R.-R.; Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.* **2020**, *11* (2), 577−586.

(10) Zheng, S.; Yan, X.; Gu, Q.; Yang, Y.; Du, Y.; Lu, Y.; Xu, J. QBMG: quasi-biogenic molecule generator with deep recurrent neural network. *J. Cheminf.* **2019**, *11* (1), 5.

(11) Jin, W.; Yang, K.; Barzilay, R.; Jaakkola, T. Learning Multimodal Graph-to-Graph Translation for Molecule Optimization. In *International Conference on Learning Representations*; 2018.

(12) Fu, T.; Xiao, C.; Sun, J. In *Core: Automatic molecule optimization using copy & refine strategy. Proceedings of the AAAI Conference on Artificial Intelligence* **2020**, *34*, 638−645.
(13) Jin, W.; Barzilay, R.; Jaakkola, T. Junction Tree Variational Autoencoder for Molecular Graph Generation. In *International Conference on Machine Learning*; 2018; pp 2323−2332.
(14) Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations ICLR 2015*; 2015.
(15) Thrun, S.; Pratt, L. *Learning to learn*; Springer Science & Business Media: 2012.
(16) Vilalta, R.; Drissi, Y. A perspective view and survey of meta-learning. *Artificial intelligence review* **2002**, *18* (2), 77−95.
(17) Vanschoren, J. Meta-learning: A survey. 2018, *arXiv preprint arXiv:.03548*. https://arxiv.org/abs/1810.03548 (accessed 2021-03-08).
(18) Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In *Advances in neural information processing systems*; 2016; pp 3630−3638.
(19) Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*; 2017.
(20) Nichol, A.; Achiam, J.; Schulman, J. On first-order meta-learning algorithms. 2018, *arXiv preprint arXiv:.02999*. https://arxiv.org/abs/1803.02999 (accessed 2021-03-08).
(21) Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. 2020, *arXiv preprint arXiv:2004.05439*. https://arxiv.org/abs/2004.05439 (accessed 2021-03-08).
(22) Nguyen, C. Q.; Kreatsoulas, C.; Branson, K. M. Meta-Learning Initializations for Low-Resource Drug Discovery. 2020, *arXiv preprint arXiv:.05996*. https://arxiv.org/abs/2003.05996 (accessed 2021-03-08).
(23) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Felix, E.; Magarinos, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; Gordillo-Maranon, M.; Hunter, F.; Junco, L.; Mugumbate, G.; Rodriguez-Lopez, M.; Atkinson, F.; Bosc, N.; Radoux, C. J.; Segura-Cabrera, A.; Hersey, A.; Leach, A. R. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **2019**, *47* (D1), D930−D940.
(24) Landrum, G. J. R. *Rdkit documentation*; 2013; Vol. *1*, pp 1−79.
(25) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742−754.
(26) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*; 2017; pp 5998−6008.
(27) Zheng, S.; Rao, J.; Zhang, Z.; Xu, J.; Yang, Y. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *J. Chem. Inf. Model.* **2020**, *60* (1), 47−55.
(28) Mao, K.; Zhao, P.; Xu, T.; Rong, Y.; Xiao, X.; Huang, J. Molecular Graph Enhanced Transformer for Retrosynthesis Prediction. 2020, *bioRxiv*. https://www.biorxiv.org/content/10.1101/2020.03.05.979773v1 (accessed 2021-03-08), DOI: 10.1101/2020.03.05.979773.
(29) Yang, Y.; Zheng, S.; Su, S.; Zhao, C.; Xu, J.; Chen, H. SyntaLinker: automatic fragment linking with deep conditional transformer neural networks. *Chem. Sci.* **2020**, *11* (31), 8312−8322.
(30) Song, Y.; Zheng, S.; Niu, Z.; Fu, Z.-H.; Lu, Y.; Yang, Y. In *Communicative Representation Learning on Attributed Molecular Graphs. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,(IJCAI 2020)* **2020**, 2831−2838.
(31) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59* (8), 3370−3388.
(32) Graves, A. Sequence Transduction with Recurrent Neural Networks. 2012, *arXiv preprint arXiv:.3711*. https://arxiv.org/abs/1211.3711 (accessed 2021-03-08).
(33) Arús-Pous, J.; Patronov, A.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. SMILES-based deep generative scaffold decorator for de-novo drug design. *Journal of Cheminformatics* **2020**, *12* (1), 38.
(34) Chen, J.; Zheng, S.; Zhao, H.; Yang, Y. Structure-aware Protein Solubility Prediction From Sequence Through Graph Convolutional Network And Predicted Contact Map. 2020, *bioRxiv*. https://www.biorxiv.org/content/10.1101/2020.06.24.169011v1.full (accessed 2021-03-08), DOI: 10.1101/2020.06.24.169011.
(35) Zheng, S.; Li, Y.; Chen, S.; Xu, J.; Yang, Y. Predicting drug-protein interaction using quasi-visual question answering system. *Nature Machine Intelligence* **2020**, *2* (2), 134−140.
(36) Pu, L.; Govindaraj, R. G.; Lemoine, J. M.; Wu, H.-C.; Brylinski, M. DeepDrug3D: Classification of ligand-binding pockets in proteins with a convolutional neural network. *PLoS computational biology* **2019**, *15* (2), No. e1006718.

https://doi.org/10.1021/acs.jcim.0c01416
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX