# Can We Quickly Learn to "Translate" Bioactive Molecules with Transformer Models?

Emma P. Tysinger[ab], Brajesh K. Rai[a], Anton V. Sinitskiy[a*]

[a] *Machine Learning and Computational Sciences, Pfizer Worldwide Research and Development, Cambridge, MA 02139, USA*

[b] *Permanent address: Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 02139, USA*

* Corresponding author, email: anton.sinitskiy@pfizer.com

## Abstract

Meaningful exploration of the chemical space of druglike molecules in drug design is a highly challenging task due to a combinatorial explosion of possible modifications of molecules. In this work, we address this problem with transformer models, a type of machine learning (ML) model, with recent demonstrated success in applications to machine translation and other tasks. By training transformer models on pairs of similar bioactive molecules from the public ChEMBL dataset, we enable them to learn medicinal-chemistry-meaningful, context-dependent transformations of molecules, including those absent from the training set. Most generated molecules are highly plausible and follow similar distributions of simple properties (molecular weight, polarity, hydrogen bond donor and acceptor numbers) as the training dataset. By retrospective analysis of the performance of transformer models on ChEMBL subsets of ligands binding to COX2, DRD2, or HERG protein targets, we demonstrate that the models can generate structures identical or highly similar to highly active ligands, despite the models having not seen any ligands active against the corresponding protein target during training. Thus, our work demonstrates that transformer models, originally developed to translate texts from one natural language to another, can be easily and quickly extended to "translations" from known molecules active against a given protein target to novel molecules active against the same target, and thereby contribute to hit expansion in drug design.

## Introduction

A critical component of small molecule drug discovery is *hit expansion*, the process of designing, synthesizing and testing candidate compounds active against a certain protein target, when some active compounds against this target ("hits") are already known. A successful drug discovery campaign should generate a diverse set of active compounds to de-risk the absorption, distribution, metabolism, elimination, toxicity (ADMET) challenges. Hit expansion has been a difficult task for several reasons, including a combinatorial explosion of the number of chemically-meaningful modifications of bioactive molecules, a need to ensure synthetic accessibility of new compounds, limiting possible modifications to minor changes in R-groups, and complex relationships between the structure of molecules and their multiple properties relevant for drug optimization.

Machine learning (ML) has been actively used for drug design (for recent reviews, see Refs. 1). However, applications of ML specifically to hit expansion have limited demonstrated success.[2, 3-9] A question arises whether this practice could be improved with ML, and if so, what kind of ML models could better serve this purpose.

In this work, we use *transformer models* for this end. Transformer models were first proposed in 2017[10] for translation between natural languages (Fig. 1a), and have gained a lot of popularity due to successful applications to various tasks, including machine translation[10, 11], predictions of chemical reaction outcomes,[12] and protein structure prediction.[13] In this paradigm, we consider hit expansion as a process of "translation" of known molecules active against a certain target into novel molecules that should be active (preferably, more active) against the same target (Fig. 1b). For this translation, we write input and output molecules in the form of SELFIES,[14] a representation that – unlike a more popular SMILES representation – automatically ensures chemical validity of nearly 100% of random strings, thereby helping an ML model to focus on learning chemically informative trends and rules in the training set, without a need to learn grammar rules of composing chemically valid structures first.

To better simulate the original field of transformer models, we train them on SELFIES of *pairs* of similar bioactive molecules. Such datasets of pairs can be generated from a large corpus of bioactive molecules (specifically, ChEMBL[15] in this work). A transformer model trained on such pairs of molecules should learn chemically-meaningful rules for modifications in a molecule that have a high chance of resulting in another biologically active molecule. Then, known structures of active compounds can be inputted into a trained model to generate new molecules, and therefore, new ideas for hit expansion.

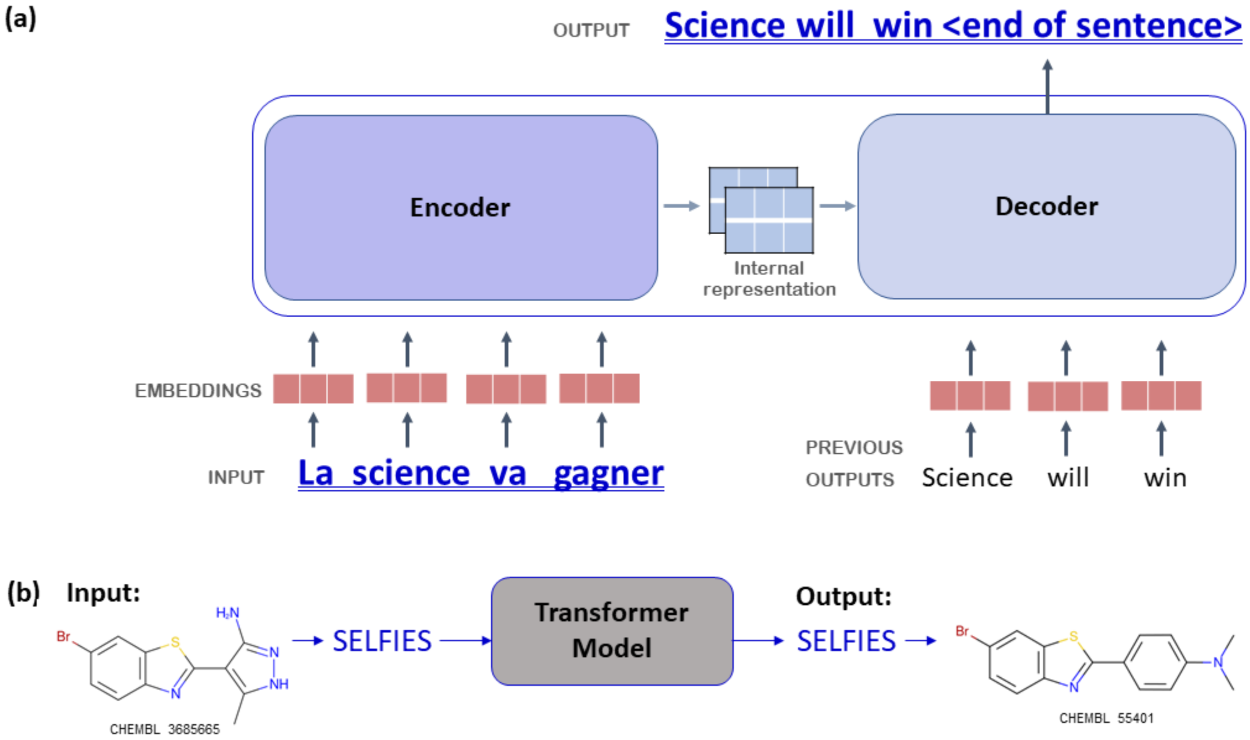

**Fig. 1**. (a) Transformer models were originally developed for and have demonstrated much success in machine translation between natural languages. (b) In this work, we apply the same approach to generating new molecular structures for the purposes of drug design.

The conceptual similarity of this approach to the initial use of transformers allowed us to hope that the goal could be achieved in a relatively quick and straightforward way. The availability of software building blocks and reference transformer implementations from other researchers made it possible to implement and benchmark our implementation during the course of a summer internship of the first author (E.P.T.).

## Results

### *1. Transformer models trained on pairs of bioactive molecules can generate new molecular transformations for hit expansion*

In this work, we show that transformer models are not only able to generate new molecular structures absent from the training dataset, but, in doing so, also go beyond the standard matched molecular pairs (MMP)-based approach.[3, 4, 16] To demonstrate this, we first generated all MMP transformation rules (in the form of SMIRKS) for our training subset of ChEMBL[15] molecules; expectedly,[4, 16] the most ubiquitous of them were additions or replacements of single atoms (H, F, Cl, etc.) or simple groups (methyl, methoxy, ethyl, etc.) (Fig. S1). Then, we applied the same procedure to generate all reasonable SMIRKS between the molecules outputted by the transformer ML model and the input molecules taken from the validation subset (see Methods) We identified 1086 SMIRKS for ***non-evident, medicinal-chemist-level molecular transformations*** that the transformer model invented, in the sense that they were absent from the set of MMP transformations for the training subset (Fig. 2). In most cases, these novel transformations may be interpreted in terms of broad categories familiar to the field, such as rigidification of a molecule by ring formation (Fig 2a), R-group substitution (Fig. 2c), or heterocycle replacement, while preserving the key pharmacophore pattern (Fig. 2b and 2d).

Generating new molecules with a transformer model is not simply applying all possible SMIRKS (known or new) to each input molecule. By construction of the ML model, the transformation rules are context-specific, and the number of outputted molecules is limited. Thus, transformer modeling prevents a combinatorial explosion of all possible ways of applying known molecular transformations to a given molecule.

Despite the structural novelty of the generated molecules, their easily computable molecular properties (molecular weight, octanol-water partition coefficient, and numbers of hydrogen bond donors and acceptors) follow similar distributions as for the training molecules or all molecules in the ChEMBL dataset (Fig. S2). This invariance of distributions is desirable, because with drug-like molecules as the input, the output of the model also stays drug-like (at least for the shown properties). This invariance is also expected, because in the training set, the assignments of which molecule is considered as an input, and which one as the output, were made at random, and therefore, did not create a bias towards higher or lower values of the listed molecular properties.

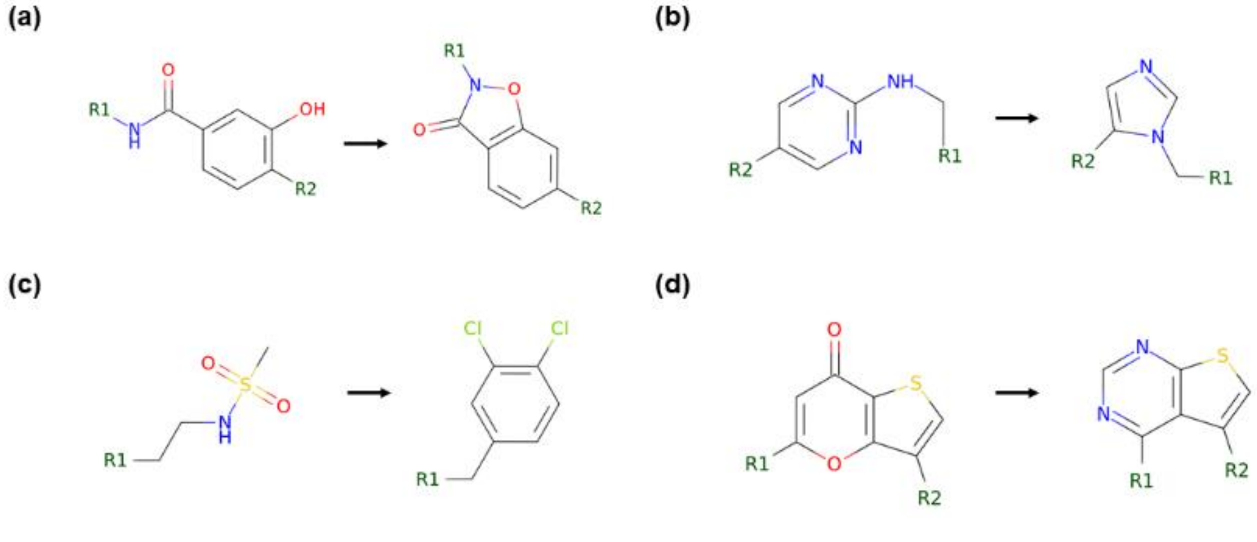

**Fig. 2.** Transformer-generated transformations go beyond matched molecular pair (MMP)-based transformations in exploring the chemical space: Transformer models trained on the ChEMBL dataset in this work have generated 1086 non-trivial molecular transformations absent from the training set. Several transformations are depicted, illustrating various classes of possible changes. (a) New ring(s) may be formed (or broken), typically with the use of atoms and functional groups present in the input molecule, making the structure more (or less) rigid. (b) Certain fragments of a molecule may be simplified, oftentimes approximately preserving original pharmacophore patterns. (c) R-groups attached to a constant core may be changed. (d) Heteroatoms in heterocyclic cores may be rearranged.

### *2. Perplexity, an information-theory-based score, changes in agreement with chemical scores of the performance of a model*

The results shown in Fig. 2 and S2 refer to a transformer model trained for 12 epochs on the ChEMBL data filtered with the cutoff level (Fig. S1, *red lines*) of 50. We focus now on the choice of these parameters and the corresponding sensitivity of the results. To decide on when to stop training, we monitored not only common information-theory-based scores, but also chemical scores of the transformer model output. As a representative of the former, we use perplexity score[17] in this work (see Methods, "Model Evaluation"); for the latter, we computed the total number of successfully generated molecules in the output, the number of scaffold change transformations, the number of R-group change transformations, the number of unique scaffolds, and the number of new scaffolds in the output (see Methods).

The dynamics of the validation perplexity and its difference from the training perplexity (Fig. 3a) suggests that training should be stopped after 10 to 12 epochs. To guide the eye, we also plotted the difference between the perplexities on the validation and training sets for the current epoch (Fig. 3b, "Perplexity Validation – Train"), and the difference of the perplexity on the validation set at the current and previous epochs of training (Fig. 3b, "Delta Perplexity Validation"). Evidently, the validation perplexity stabilizes after 10 to 12 epochs. Note, however, that the early stopping criterion is not applicable here, because the validation perplexity continues decreasing very slowly even around epoch 32, and one would not stop training with this criterion even at such a late stage.

Notably, at the same stage of training (epochs 10-12), we observed a qualitative change in the pattern of chemical scores of the model. While the total number of successfully generated molecules stayed nearly the same at various epochs (Fig. 3c), the type of the molecular transformation undertaken by the model changed. Before epochs 10-12, the scaffold of an output molecule was typically different from the scaffold of the corresponding input molecule ("scaffold change count", Fig. 3c), while after epochs 10-12, increasingly, generated molecules had the same scaffolds as the corresponding input molecules, and the molecular transformations were limited to changes in the scaffold decorations ("R-group change count", Fig. 3d). Also, at epochs 10-12 the maximal diversity of generated molecules is reached, as measured by the number of unique or new (relative to the input set) scaffolds of the generated molecules (Fig. 3e). At subsequent epochs, the number of new generated scaffolds and the fraction of transformations that affect scaffolds decreases, while the count of simpler transformations ("R-group change") increases, which we interpret as overtraining the ML model. These observations may also suggest that appropriately trained transformer models, at least with the parameters used in this work, are more appropriate for scaffold-modifying molecular transformations than for changes affecting only side groups.

We explored sensitivity of these conclusions to how the data filtering was performed, namely, to the filter cutoff values (see Methods). The results reported above refer to the filter cutoff of 50. Lowering the cutoff to 20 (Fig. S3, "filter20") or increasing it to 500 (Fig. S3, "filter500") keeps the behavior of the scores unchanged: perplexities on the training and validation sets decrease fast during first epochs, and then reach plateaus; the total number of generated molecules stays constant and nearly equal to the number of inputted molecules (perhaps, after some fluctuations at initial training epochs); the number of generated molecules with changed scaffolds (relative to the corresponding input molecule) rapidly drops, while that of the molecules with side group changes rapidly increases, both at epochs where perplexity stabilizes; finally, at the same epochs, the numbers of unique and new (relative to all inputted molecules) scaffolds reach maxima. However, further significant decreases in the filter cutoff value deteriorate the performance of ML models: As illustrated for the cutoff value of 5 (Fig. S3, "filter5"), the number of new or unique scaffolds in the set of generated molecules is much smaller than for cutoffs of 20, 50 or 500, and they reach plateaus, not maxima, during training. Therefore, the filter cutoff of 50 used in this work is appropriate (though not unique) for the transformer model to learn medicinally-chemically-relevant transformations of compounds.

### *3. Target-specific held-out models can generate ideas for some of the most active ligands*

To emulate an application of transformer models to real-life drug design projects, we carried out retrospective analysis of its performance for several protein targets with data on potency available in ChEMBL, namely, COX2, DRD2 and HERG proteins (note that the HERG protein is usually considered in drug design as an antitarget due to serious cardiac side effects,[18] but in this work we consider it as a target to test the proposed method on a large public dataset of molecules binding to it). For each target, we excluded its ligands from the training dataset, and trained a new transformer model. Hence, for each of these three protein targets, we trained a separate ML model, ignorant to molecules active against the given target.

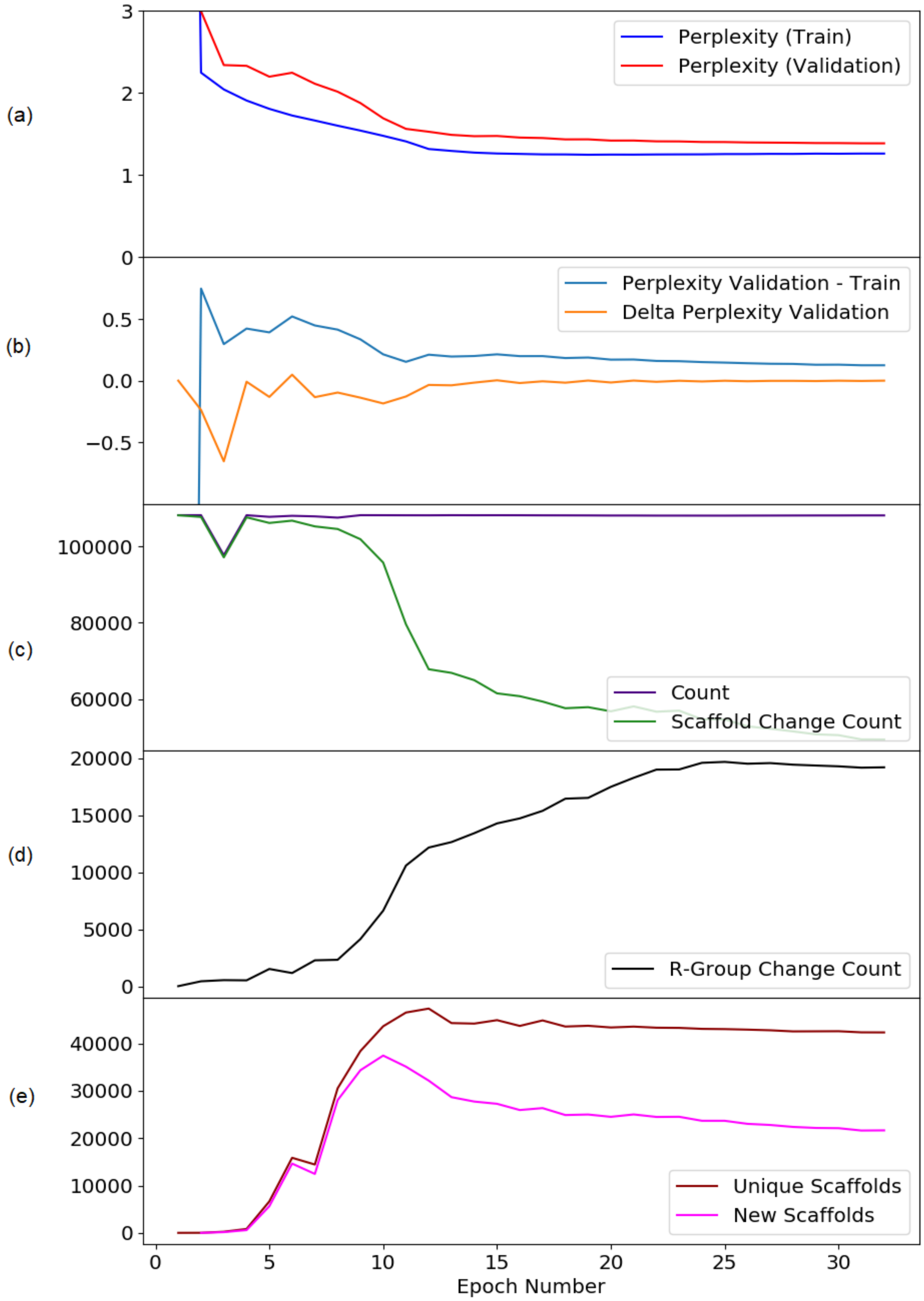

**Fig. 3.** Chemical scores align with perplexity, a common information-theory-based score, during training of the generative chemistry ML model. (a) The dynamics of the validation and training perplexity suggests that training should be stopped around epochs 10-12. (b) The difference between the perplexities on the validation and train sets, and between the validation perplexity at the current and previous training steps are also plotted to guide an eye. (c-e) At the same stages of training, the maximal structural diversity of generated molecules is reached, while at subsequent epochs the diversity decreases, presumably indicating that the ML model gets overtrained. The chemical scores of generated molecules used in this work are: (c) a total number of generated molecules, the number of molecules with new scaffolds, (d) the number of molecules with changes in side groups, (e) the number of unique and new scaffolds in the generated molecules.

Next, for each target protein, we split the set of its ligands into two subsets, based on the ligand activity (measured by pIC50, pEC50 or pKi values, depending on the target). The subset of 95% molecules used as input to a transformer model (further called ‘input subset’) was formed by weakly and moderately active ligands, while 5% most active ligands formed a ‘test subset’ used for scoring the output molecules. With this 95%:5% split, we compare generated structures with the most active molecules; note that in practical scenarios all 100% known active molecules, not only 95%, can be used as the input.

Note that target-specific information was contained only in the input to ML models, and not used for training ML models themselves. In practical applications, one would not have to retrain an ML model to make predictions for a new protein target (though in this work we had to retrain a model from scratch for each target to make sure that the training set does not include molecules active against the given target for the purpose of the method validation).

Surprisingly, these target-naïve ML models managed to generate not just reasonable molecular structures, but molecular structures some of which are highly active ligands against each of the three protein targets, or at least provide ideas for such ligands (Fig. 4). Some of the generated molecules coincided with the known highly active ligands, while some other generated molecules have minor differences from known actives (e.g., they may differ by -Cl, -OH, -$CH_3$ groups, etc.). This ability to generate, among others, highly active ligands from moderately active ones, or ideas for such highly active ligands, was not an ability that these models were explicitly trained for, because the pairs of molecules in the training sets were not ordered by bioactivity. Generated molecules could in principle be less active or more active as input molecules, but the results reported in this section demonstrate that a sufficiently large fraction of output molecules is highly active.

## Discussion

First of all, we compare our approach to *MMP analysis*,[3, 4, 16, 19] a popular algorithm for lead optimization. This method is based on extraction of transformation rules from (in the terminology of ML) a training set of molecules, and application of these rules to known actives. Transformations captured in this way are generic and generally applicable, the method is based on the additivity principle, which is one of the main assumptions in medicinal chemistry, and it yields multiple suggested molecules. On the other hand, combinatorial explosion of generated structures limits its applicability, and the additivity principle often violates in practice. By contrast, the method used in this work generates only prioritized structures, evading a combinatorial explosion by taking into account a wider molecular context of input molecules, not limited by the MMP chemical environment radius, and, being highly non-linear by construction, does not imply the additivity.

As for *transformer models* in generative chemistry, they have demonstrated noticeable success, but for purposes different than in this work. For example, transformers were used to generate structures of molecules from their text description in English[20] or even from the primary amino acid sequence of a protein they should bind to.[21] Fragment replacement with optimization of logP, PSA and other easily computable properties was reported.[7] New molecules with desired values of logP, TPSA, QED (and SAS in the second case) were generated from a seed molecule[6] or a scaffold.[9] A combination of a Restricted Boltzmann Machine (RBM) with two transformer models was built, with the hope to replace an RBM with a Quantum Boltzmann Machine in the future, and thereby enable quantum computing in drug design.[5] A latent space representation of organic molecules was constructed with a transformer model and contrastive learning approach.[22] A transformer model, pre-trained on ChEMBL to generate valid SMILES, was trained on molecules active against a given protein target by transfer learning and reinforcement learning, and used to generate novel molecules.[8] Note that in all these papers generated molecules were validated only based on their easily computable properties, such as logP and QED (typically, by comparisons of distributions of these properties with those for the input or reference sets); the most advanced comparison, as far as we know, used docking scores and similarity to known bioactive molecules.[8] Also, the cited papers address, in essence, the problem of *de novo* compound design. For example, in Ref. 8, the model was transfer-learned on 4702 known molecules active against the protein target, much more than what we usually have for hit expansion, then 50k molecules were generated by the model and prioritized docking. In this work, we follow the best practices in the field, validating generated molecules by their easily computed properties and a comparison to known active molecules. We apply transformer models for the first time, as far as we know, to the task of hit expansion.

Our model is built in such a way that it learns about biological activity of molecules implicitly, from pairs of molecules used for training. It was known that transformer models can learn general rules of composition of relevant organic molecules without explicitly using information on their stability;[7, 12, 20] here, we extend such an approach to generating biologically active molecules. This approach is not the only possible solution. As examples of alternatives, we can mention Ref [23], where a separate LSTM model was trained to prioritize SMILES coming from a generative model, and Ref [7], where a transformer model is trained to optimize logP and other easily computable properties (but not biological activity) of generated molecules. Note also that transfer learning or fine tuning of a generative chemical model on molecules active against the given target[8] is not necessary, as we show in this work.

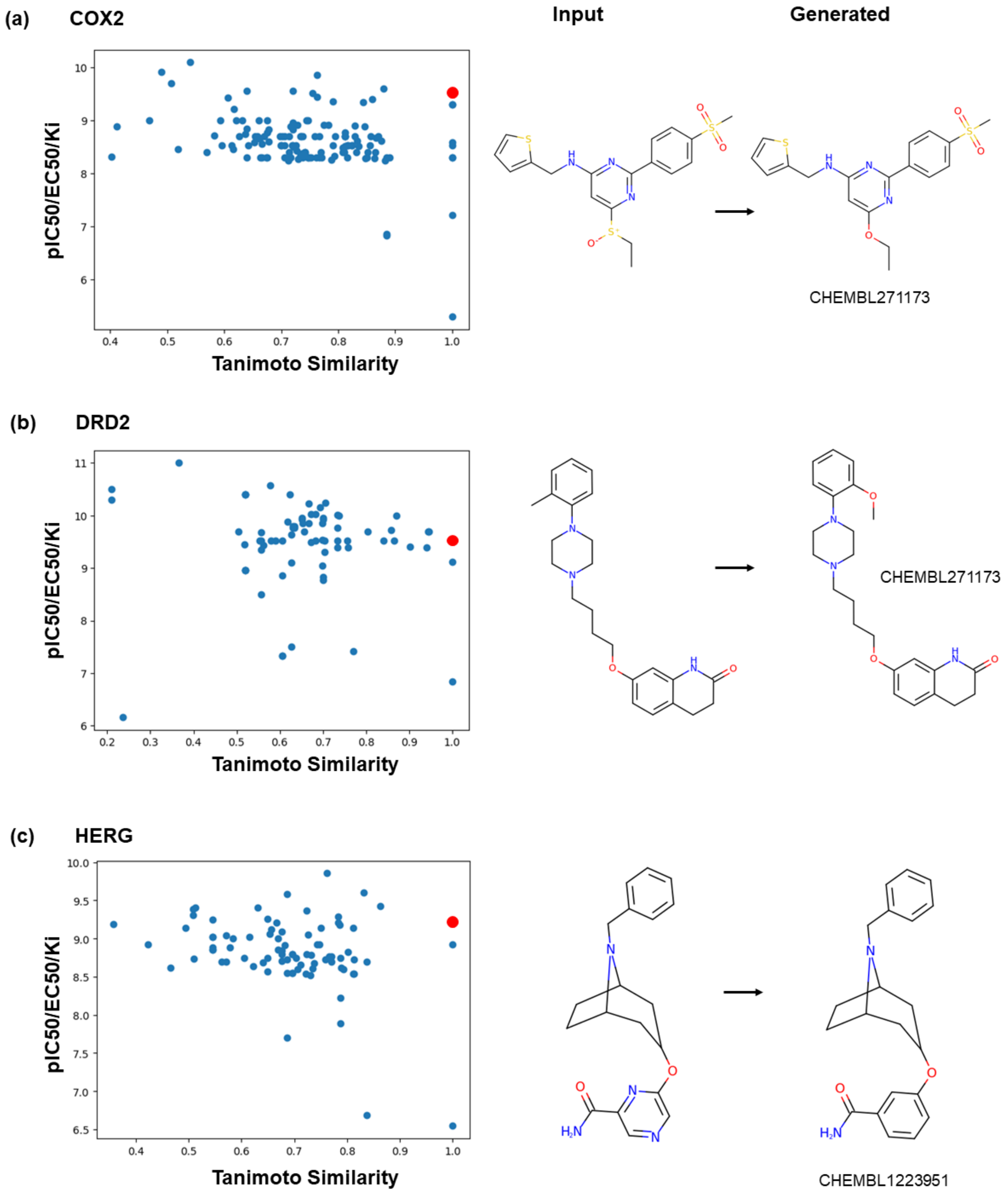

**Fig. 4.** Even without fine-tuning on project-specific data, transformer models can generate structural ideas for some of the most active ligands against a given target, with the use of less potent molecules as the input. In the scatter plots, each dot corresponds to a generated molecule, its Tanimoto similarity to the structurally closest molecule from the training set is shown on the x axis, and the experimental potency (pIC50 or pEC50 or pKi) of this closest molecule shown on the y axis. Most potent molecules exactly generated by the transformer models are shown by red cycles on the plots, and the corresponding chemical structures are shown on the right. Protein targets: (a) COX2, (b) DRD2, (c) HERG.

We could speculate that one of the reasons for success was the use of SELFIES to represent molecules in the input to and output from a transformer model, because nearly any SELFIES corresponds to a meaningful chemical structure. SMILES encoding is still more popular,[5, 6, 8, 9, 12, 17, 21, 23, 24] but a random SMILES string is highly unlikely to represent a valid molecule, and special efforts have to be undertaken to train the model property. Interestingly, papers using SMILES representation in generative chemistry always have a separate section devoted to the demonstration of the validity of generated molecules. By contrast, SELFIES-based generative models do not have to learn formal grammar rules first. Only a small fraction of SELFIES generated by ML models in this work were chemically insensible, and the reasons for it were the following: generated SELFIES were empty strings; generated molecules contained non-flat multi-cycle aromatic systems; or generated molecules contained fragments (e.g., -O-O-F) hardly acceptable in drug molecules. However, such issues rarely occurred (less than 1% of all cases) and were all rejected by simple filters.

This work demonstrates that training a transformer model proceeds from learning to creatively generate more scaffolds of bioactive molecules to – at the stage of overtraining – generating new decorations to known scaffolds. The optimal state of a model is reached at the border between these two regimes, ensuring a maximal chemical diversity of the generated structures. We also show that perplexity can serve as a score of the quality of de novo generated molecules, in agreement with previous work,[17] which differs from our work in the following aspects. In that paper, perplexity was used to score SMILES (not SELFIES) of molecules generated with an LSTM (not transformer) model, and Tanimoto similarities on Morgan and topological pharmacophore fingerprints between generated and true bioactive molecules were used as the chemical scores. This conclusion is of practical importance, because chemical scores typically do not provide a differentiable loss function for ML training, and can realistically be computed only for validation of a trained model.

This work has certain limitations and leaves some important questions unanswered. We did not perform an exhaustive hyperparameter search and explored the effect of only two parameters on the quality of generated molecules: the number of training epochs, and the cutoff value $N$ for filtering the training set. Another limitation is that only one output molecule was generated per input molecule, which could be particularly restrictive when only few initial hits are available, as is often the case early in the hit expansion process. Increasing the number of generated molecules could further improve the practical applicability of the method, though causing a need for additional prioritization, a complication that we tried to evade in this work. Also, note that the training sets were formed by pairs of molecules without ordering by bioactivity. Moreover, two molecules in a pair could be active against different targets. As a result, models reported here cannot be expected to bias generation of molecules towards higher bioactivity. They only perform medicinally-chemically-meaningful steps in the chemical space, and the goal of improving bioactivity (or optimizing ADMET properties, synthesizability, etc., under given restrictions on bioactivity) can be solved, for example, by postprocessing (triaging/filtering) the output of such generative models, or including one into a larger reinforcement-learning-type ML model. Note, however, that despite these limitations, the most active molecules against COX2, DRD2, and HERG have been successfully recovered or approached to in this work. Another direction of performance improvement of the model would be to run fine-tuning on data referring to the same or similar protein targets, possibly ordering the molecular pairs in the training set by their bioactivity against these targets. Finally, an ability to specify target or binding site information as part of training and inference could further improve the performance of such generative models.

We conclude that transformer models can be a powerful tool for hit expansion in drug design. They can go beyond MMP in generating reasonable context-dependent modification of bioactive molecules and demonstrate the ability to predict the most active known molecules, exactly or up to minor modification, in retrospective tests on public datasets. A common informatic-theory-based score of the quality of predictions, namely perplexity, correlates with chemical scores, and can be used instead of the latter, that are more difficult to formalize and, if formalized, are often non-differentiable (such as the number of novel molecules, scaffolds, R-group replacements, etc.). Finally, we demonstrate that the entrance barrier for this method of hit expansion is low, as proven by its successful implementation during a summer internship of the first author of this paper, an undergraduate student without prior professional experience in drug design (E.P.T.).

## Methods

***Data Preparation*** – For training of the transformer model, molecules from ChEMBL[15] (version 28, dated back to January 15, 2021) were used. ChEMBL is a public dataset of biologically active molecules. We selected molecules with the following criteria: molecule type was "small molecule", and molecule record had an IC50, EC50 or Ki value. All disconnected structures (defined as '.' in the SMILES structural representation; typically, these are salts) were excluded, and all stereochemistry was removed. This selection resulted in 940 640 molecules.

SMILES representations of the molecules were inputted into MMPDB (Matched Molecular Pair Database Generation) software[25] to pair structurally similar molecules. For each pair, MMPDB outputs the two SMILES, the constant group, and the

SMIRK, defined as the transformation rule between the two molecules. From the above-mentioned 940 640 molecules, ~57 million pairs of similar molecules were generated.

Next, we filtered the set of molecular pairs in the following way: (a) all molecular pairs with SMIRKs occurring less than $N_1$ times were excluded, and (b) for each remaining SMIRK, only $N_2$ randomly chosen pairs of molecules were kept (Fig. S1, *red lines*). The first condition eliminated very rare SMIRKs to suppress the noise in the MMPDB output. As a visual inspection of a random subset of rare SMIRKs showed, they correspond to transformations that seem unnatural from the viewpoint of Medicinal Chemistry (e.g., keep a small part of a molecule unchanged, even as small as a methyl group, and replace a much bigger part of a molecule). The second condition was introduced to eliminate an extreme bias of the original set of molecular pairs towards the simplest transformations, such as a replacement of a hydrogen atom by a methyl group or a halogen atom, and thereby to ensure diversity of molecular transformations in the dataset (Fig. S1). In general, $N_1$ and $N_2$ may be different, but in this work, we considered only filters with the same values of $N_1$ and $N_2$. For brevity, we use the term "Filter *N*" in this paper to refer to the above-described dataset filtering procedure with the cutoffs of $N = N_1 = N_2$.

Then, a timesplit of a filtered set of molecular pairs into training, validation and testing subsets was performed. The timestamp for each molecule was defined as the year of the first publication that mentioned this molecule (as reported in the ChEMBL). A training set was formed by pairs in which both molecules had timestamps before a train year threshold. A test set was defined as at least one molecule in each pair was experimentally published after a test year threshold. A validation set was comprised of all remaining pairs. We used the train and test cutoff years of 2013 and 2015, respectively, which ensures an approximate 2:1:1 ratio of the sizes of the training, validation and test subsets, respectively. In each subset, SMILES representations of molecules were converted to SELFIES.[14]

***Model*** – We adapted the transformer model from OpenNMT, an open-source neural machine translation project using PyTorch, with its default settings (unless explicitly stated) for our generative chemistry project. We chose this software because it is actively maintained and has a high number of users in various fields of application. The batch size of 128 was used (decreasing the batch size to 16 did not improve the performance of trained models but slowed down training; data not shown). All our programs for preprocessing data, training a model, using a model to make predictions, and postprocessing results were written in Python and ran with SLURM scripts in a High-Performance Computing cluster at Pfizer. Models were trained on one GPU of a 4-GPU (NVidia V100) node.

***Model Evaluation*** – For evaluation, the validation set of molecules was used in two different ways. First, it served as a set of pairs of molecules to compute ***information-theory-based scores*** of ML models. In this work, we report and discuss perplexity of ML models,[17] as shown in Figs. 3a and S3, as a representative score from this group; no new conclusions were found from other information-theory-based scores (data not shown). Perplexity *ppl* is defined, as usual, as: $ppl = \exp(L/N)$, where $L$ is the loss function minimized to train the model, and $N$ is the number of "words" (output molecules in our case). We used the default choice of the loss function in OpenNMT.

Second, all molecules from the validation set were inputted into the model using OpenNMT *translate* module, and multiple ***chemical scoring*** metrics were computed for the molecules generated as an output, as shown in Figs. 3b and S3. The following scoring metrics, aimed to capture the diversity and reasonableness of the molecules generated, were implemented and evaluated on the generated molecules set:

- the total number of successfully generated molecules,
- the number of scaffold change transformations,
- the number of R-group change transformations,
- the number of unique scaffolds,
- the number of new scaffolds.

To calculate these scores, we determined scaffolds of each generated molecule and each input validation molecule with Chem.Scaffolds.MurckoScaffold function from RDKit. Then, the type of transformation for each generated molecule was found by comparing its scaffold to the scaffold of the respective input validation molecule. If the two scaffolds were the same (but the molecules were different), then the transformation was classified as "R-group change". If the two scaffolds were different, then the transformation was classified as "scaffold change". The cases of each transformation type were counted over the whole set of generated molecules. (Note that the sum of the number of scaffold change transformations and the number of R-group change transformations may be slightly less than the total number of successfully generated molecules because of the rare cases when input and output molecules were identical.) The number of unique scaffolds refer to the total number of unique scaffolds in the set of generated molecules. The number of new scaffolds refer to those unique scaffolds that were absent from the training or input validation sets.

***Target-held-out Models*** – Like in the subsection "Data Preparation", SMILES representations of all the molecules were inputted into MMPDB, structurally similar molecules were paired, and filtering was performed to exclude rare SMIRKs and randomly sample remaining SMIRKs. After that, and unlike the previously described data preparation, all pairs with at least one molecule active against a given target (COX2, DRD2 or HERG) were removed. For each target, a transformer model was trained with the default values of hyperparameters. Each of the resulting three trained models was “target-agnostic” in the sense that during its training it has not seen any ligands active against the corresponding protein target.

For each of those three protein targets, the set of ligands active against a given target was split by their activity levels (as measured by EC50, IC50 or Ki values reported in ChEMBL) into two subsets, one with the bottom 95% least active molecules, and the other with the top 5% most active molecules. Molecules from the first group (weakly and moderately active ligands) were inputted into the corresponding target-agnostic transformer model. Then, the Tanimoto similarity score (on ECFP4, Morgan fingerprints with the radius of 2, computed with AllChem.GetMorganFingerprint from RDKit) was calculated between each generated molecule and each of the top 5% most active ligands from ChEMBL for the same protein target. The Tanimoto similarity score is a standard metric of a structural similarity of two organic molecules. Then, for each of those most active molecules from ChEMBL (that is, *experimentally* known to be active), a closest *generated* molecule was identified (as the molecule with the highest Tanimoto similarity). The scatter plots for the corresponding Tanimoto similarities vs. the corresponding experimental potencies were drawn for each of the protein targets (Fig. 4).

## Acknowledgements

The authors would like to thank Vishnu Sresht for his detailed feedback on the results and the manuscript, Ray Unwalla for useful discussion, Jacquelyn Klug-McLeod, Dmitri Bichko, Helena Qi and Nicholas Labello for help with the data and computational resources, and Enoch Huang for the general support.

## References

(1) Schneider, G. Automating drug discovery. *Nat Rev Drug Discov* **2018**, *17* (2), 97-113. DOI: 10.1038/nrd.2017.232 Schneider, G.; Clark, D. E. Automated De Novo Drug Design: Are We Nearly There Yet? *Angew Chem Int Ed Engl* **2019**, *58* (32), 10792-10803. DOI: 10.1002/anie.201814681 Brown, N.; Ertl, P.; Lewis, R.; Luksch, T.; Reker, D.; Schneider, N. Artificial intelligence in chemistry and drug design. *J Comput Aided Mol Des* **2020**, *34* (7), 709-715. DOI: 10.1007/s10822-020-00317-x de Souza Neto, L. R.; Moreira-Filho, J. T.; Neves, B. J.; Maidana, R.; Guimaraes, A. C. R.; Furnham, N.; Andrade, C. H.; Silva, F. P., Jr. In silico Strategies to Support Fragment-to-Lead Optimization in Drug Discovery. *Front Chem* **2020**, *8*, 93. DOI: 10.3389/fchem.2020.00093 Kim, H.; Kim, E.; Lee, I.; Bae, B.; Park, M.; Nam, H. Artificial Intelligence in Drug Discovery: A Comprehensive Review of Data-driven and Machine Learning Approaches. *Biotechnol Bioprocess Eng* **2020**, *25* (6), 895-930. DOI: 10.1007/s12257-020-0049-y Vanhaelen, Q.; Lin, Y. C.; Zhavoronkov, A. The Advent of Generative Chemistry. *ACS Med Chem Lett* **2020**, *11* (8), 1496-1505. DOI: 10.1021/acsmedchemlett.0c00088 Bender, A.; Cortes-Ciriano, I. Artificial intelligence in drug discovery: what is realistic, what are illusions? Part 1: Ways to make an impact, and why we are not there yet. *Drug Discov Today* **2021**, *26* (2), 511-524. DOI: 10.1016/j.drudis.2020.12.009 Bian, Y.; Xie, X. Q. Generative chemistry: drug discovery with deep learning generative models. *J Mol Model* **2021**, *27* (3), 71. DOI: 10.1007/s00894-021-04674-8 Hudson, I. L. Data Integration Using Advances in Machine Learning in Drug Discovery and Molecular Biology. *Methods Mol Biol* **2021**, *2190*, 167-184. DOI: 10.1007/978-1-0716-0826-5_7 Jimenez-Luna, J.; Grisoni, F.; Weskamp, N.; Schneider, G. Artificial intelligence in drug discovery: recent advances and future perspectives. *Expert Opin Drug Discov* **2021**, *16* (9), 949-959. DOI: 10.1080/17460441.2021.1909567 Liu, X.; AP, I. J.; van Westen, G. J. P. Computational Approaches for De Novo Drug Design: Past, Present, and Future. *Methods Mol Biol* **2021**, *2190*, 139-165. DOI: 10.1007/978-1-0716-0826-5_6 Meyers, J.; Fabian, B.; Brown, N. De novo molecular design and generative models. *Drug Discov Today* **2021**, *26* (11), 2707-2715. DOI: 10.1016/j.drudis.2021.05.019 Mouchlis, V. D.; Afantitis, A.; Serra, A.; Fratello, M.; Papadiamantis, A. G.; Aidinis, V.; Lynch, I.; Greco, D.; Melagraki, G. Advances in de Novo Drug Design: From Conventional to Machine Learning Methods. *Int J Mol Sci* **2021**, *22* (4). DOI: 10.3390/ijms22041676 Paul, D.; Sanap, G.; Shenoy, S.; Kalyane, D.; Kalia, K.; Tekade, R. K. Artificial intelligence in drug discovery and development. *Drug Discov Today* **2021**, *26* (1), 80-93. DOI: 10.1016/j.drudis.2020.10.010 Tong, X.; Liu, X.; Tan, X.; Li, X.; Jiang, J.; Xiong, Z.; Xu, T.; Jiang, H.; Qiao, N.; Zheng, M. Generative Models for De Novo Drug Design. *J Med Chem* **2021**, *64* (19), 14011-14027. DOI: 10.1021/acs.jmedchem.1c00927 Walters, W. P.; Barzilay, R. Critical assessment of AI in drug discovery. *Expert Opin Drug Discov* **2021**, *16* (9), 937-947. DOI: 10.1080/17460441.2021.1915982 Du, Y.; Fu, T.; Sun, J.; Liu, S. MolGenSurvey: A Systematic Survey in Machine Learning Models for Molecule Design. *arXiv preprint arXiv:2203.14500* **2022**. Heifetz, A. *Artificial Intelligence in Drug Design*; Springer, 2022. Lee, J. W.; Maria-Solano, M. A.; Vu, T. N. L.; Yoon, S.; Choi, S. Big data and artificial intelligence (AI) methodologies for computer-aided drug design (CADD). *Biochem Soc Trans* **2022**, *50* (1), 241-252. DOI: 10.1042/BST20211240 Mak, K. K.; Balijepalli, M. K.; Pichika, M. R. Success stories of AI in drug discovery - where do things stand? *Expert Opin Drug Discov* **2022**, *17* (1), 79-92. DOI: 10.1080/17460441.2022.1985108

(2) Damani, F.; Sresht, V.; Ra, S. Black box recursive translations for molecular optimization. *arXiv preprint arXiv:1912.10156* **2019**. Ghanakota, P.; Bos, P. H.; Konze, K. D.; Staker, J.; Marques, G.; Marshall, K.; Leswing, K.; Abel, R.; Bhat, S. Combining Cloud-Based Free-Energy Calculations, Synthetically Aware Enumerations, and Goal-Directed Generative Machine Learning for Rapid Large-Scale Chemical Exploration and Optimization. *J Chem Inf Model* **2020**, *60* (9), 4311-4325. DOI: 10.1021/acs.jcim.0c00120 Imrie, F.; Hadfield, T. E.; Bradley, A. R.; Deane, C. M. Deep generative design with 3D pharmacophoric constraints. *Chem Sci* **2021**, *12* (43), 14577-14589. DOI: 10.1039/d1sc02436a .

(3) Polishchuk, P. CReM: chemically reasonable mutations framework for structure generation. *J Cheminform* **2020**, *12* (1), 28. DOI: 10.1186/s13321-020-00431-w .

(4) Awale, M.; Hert, J.; Guasch, L.; Riniker, S.; Kramer, C. The Playbooks of Medicinal Chemistry Design Moves. *J Chem Inf Model* **2021**, *61* (2), 729-742. DOI: 10.1021/acs.jcim.0c01143

(5) Gircha, A.; Boev, A. S.; Avchaciov, K.; Fedichev, P.; Fedorov, A. K. Training a discrete variational autoencoder for generative chemistry and drug design on a quantum annealer. *arXiv preprint arXiv:2108.11644* **2021**.

(6) Kim, H.; Na, J.; Lee, W. B. Generative Chemical Transformer: Neural Machine Learning of Molecular Geometric Structures from Chemical Language via Attention. *J Chem Inf Model* **2021**, *61* (12), 5804-5814. DOI: 10.1021/acs.jcim.1c01289

(7) Rothchild, D.; Tamkin, A.; Yu, J.; Misra, U.; Gonzalez, J. C5t5: Controllable generation of organic molecules with transformers. *arXiv preprint arXiv:2108.10307* **2021**.

(8) Yang, L.; Yang, G.; Bing, Z.; Tian, Y.; Niu, Y.; Huang, L.; Yang, L. Transformer-Based Generative Model Accelerating the Development of Novel BRAF Inhibitors. *ACS Omega* **2021**, *6* (49), 33864-33873. DOI: 10.1021/acsomega.1c05145 .

(9) Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *J Chem Inf Model* **2022**, *62* (9), 2064-2076. DOI: 10.1021/acs.jcim.1c00600

(10) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.

(11) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**.

(12) Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. A. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Cent Sci* **2019**, *5* (9), 1572-1583. DOI: 10.1021/acscentsci.9b00576 .

(13) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Zidek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583-589. DOI: 10.1038/s41586-021-03819-2

(14) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology* **2020**, *1* (4), 045024.

(15) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Felix, E.; Magarinos, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res* **2019**, *47* (D1), D930-D940. DOI: 10.1093/nar/gky1075

(16) Pal, S.; Pogany, P.; Lumley, J. A. Molecule Ideation Using Matched Molecular Pairs. *Methods Mol Biol* **2022**, *2390*, 503-521. DOI: 10.1007/978-1-0716-1787-8_23

(17) Moret, M.; Grisoni, F.; Katzberger, P.; Schneider, G. Perplexity-Based Molecule Ranking and Bias Estimation of Chemical Language Models. *J Chem Inf Model* **2022**, *62* (5), 1199-1206. DOI: 10.1021/acs.jcim.2c00079

(18) Raschi, E.; Vasina, V.; Poluzzi, E.; De Ponti, F. The hERG K+ channel: target and antitarget strategies in drug development. *Pharmacol Res* **2008**, *57* (3), 181-195. DOI: 10.1016/j.phrs.2008.01.009

(19) Kramer, C.; Fuchs, J. E.; Whitebread, S.; Gedeck, P.; Liedl, K. R. Matched molecular pair analysis: significance and the impact of experimental uncertainty. *J Med Chem* **2014**, *57* (9), 3786-3802. DOI: 10.1021/jm500317a Tyrchan, C.; Evertsson, E. Matched Molecular Pair Analysis in Short: Algorithms, Applications and Limitations. *Comput Struct Biotechnol J* **2017**, *15*, 86-90. DOI: 10.1016/j.csbj.2016.12.003 .

(20) Edwards, C.; Lai, T.; Ros, K.; Honke, G.; Ji, H. Translation between Molecules and Natural Language. *arXiv preprint arXiv:2204.11817* **2022**.

(21) Grechishnikova, D. Transformer neural network for protein-specific de novo drug generation as a machine translation problem. *Sci Rep* **2021**, *11* (1), 321. DOI: 10.1038/s41598-020-79682-4

(22) Shrivastava, A. D.; Kell, D. B. FragNet, a Contrastive Learning-Based Transformer Model for Clustering, Interpreting, Visualizing, and Navigating Chemical Space. *Molecules* **2021**, *26* (7). DOI: 10.3390/molecules26072065

(23) Moret, M.; Grisoni, F.; Brunner, C.; Schneider, G. Leveraging molecular structure and bioactivity with chemical language models for drug design. **2021**.

(24) Dollar, O.; Joshi, N.; Beck, D. A. C.; Pfaendtner, J. Attention-based generative models for de novo molecular design. *Chem Sci* **2021**, *12* (24), 8362-8372. DOI: 10.1039/d1sc01050f Li, X.; Fourches, D. SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning. *J Chem Inf Model* **2021**, *61* (4), 1560-1569. DOI: 10.1021/acs.jcim.0c01127 Skinnider, M. A.; Stacey, R. G.; Wishart, D. S.; Foster, L. J. Deep generative models enable navigation in sparsely populated chemical space. **2021**. Grisoni, F.; Schneider, G. De Novo Molecular Design with Chemical Language Models. *Methods Mol Biol* **2022**, *2390*, 207-232. DOI: 10.1007/978-1-0716-1787-8_9

(25) Dalke, A.; Hert, J.; Kramer, C. mmpdb: An Open-Source Matched Molecular Pair Platform for Large Multiproperty Data Sets. *J Chem Inf Model* **2018**, *58* (5), 902-910. DOI: 10.1021/acs.jcim.8b00173

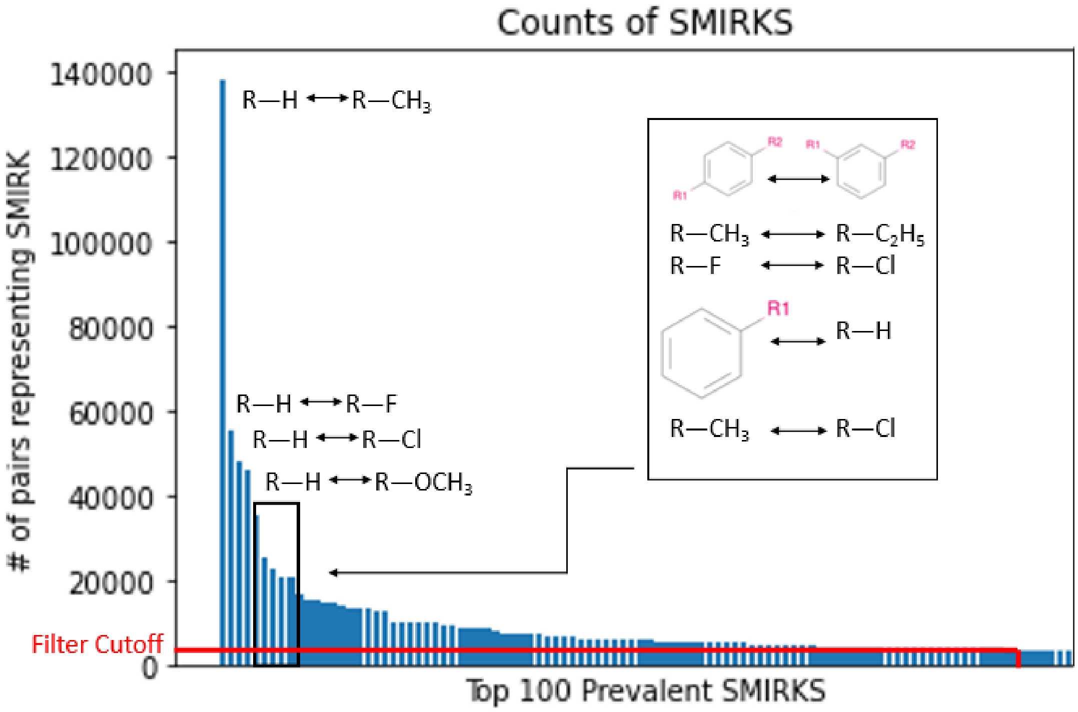

**Fig. S1.** Molecular transformations (SMIRKS) for the pairs of close ChEMBL molecules have an extremely uneven distribution, with replacements or additions of single atoms or simple groups prevailing in numbers. Top 10 most prevalent transformations in entire ChEMBL dataset are shown. Filtering approach, intended to fix the bias towards simple transformations, is illustrated by red lines.

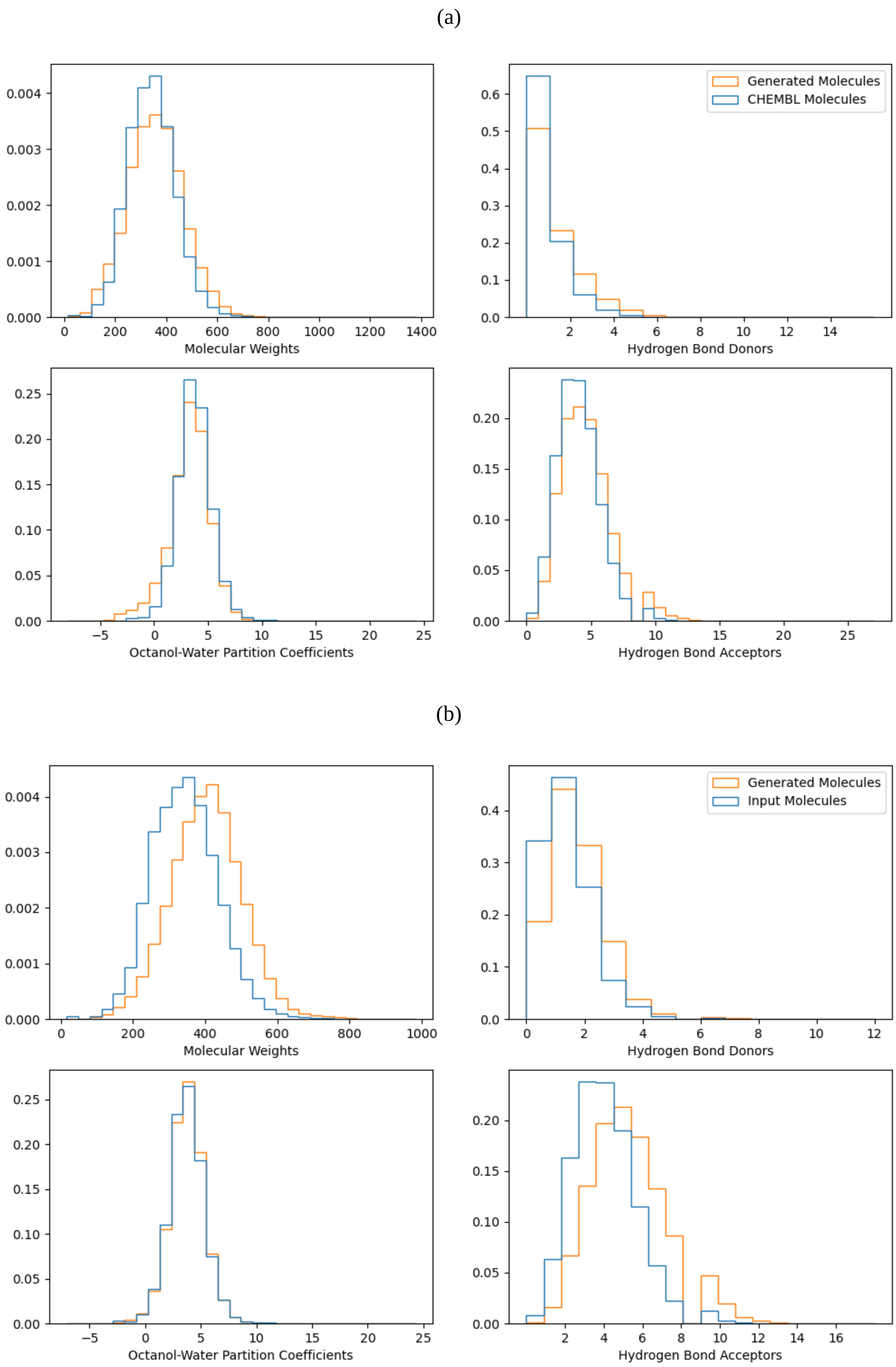

**Fig. S2.** Molecules generated with the transformer model have distributions of molecular weight, octanol-water partition coefficient, and numbers of hydrogen bond donors and acceptors that are similar to the corresponding distributions for (a) all ChEMBL molecules and (b) molecules used as the input to the transformer model.

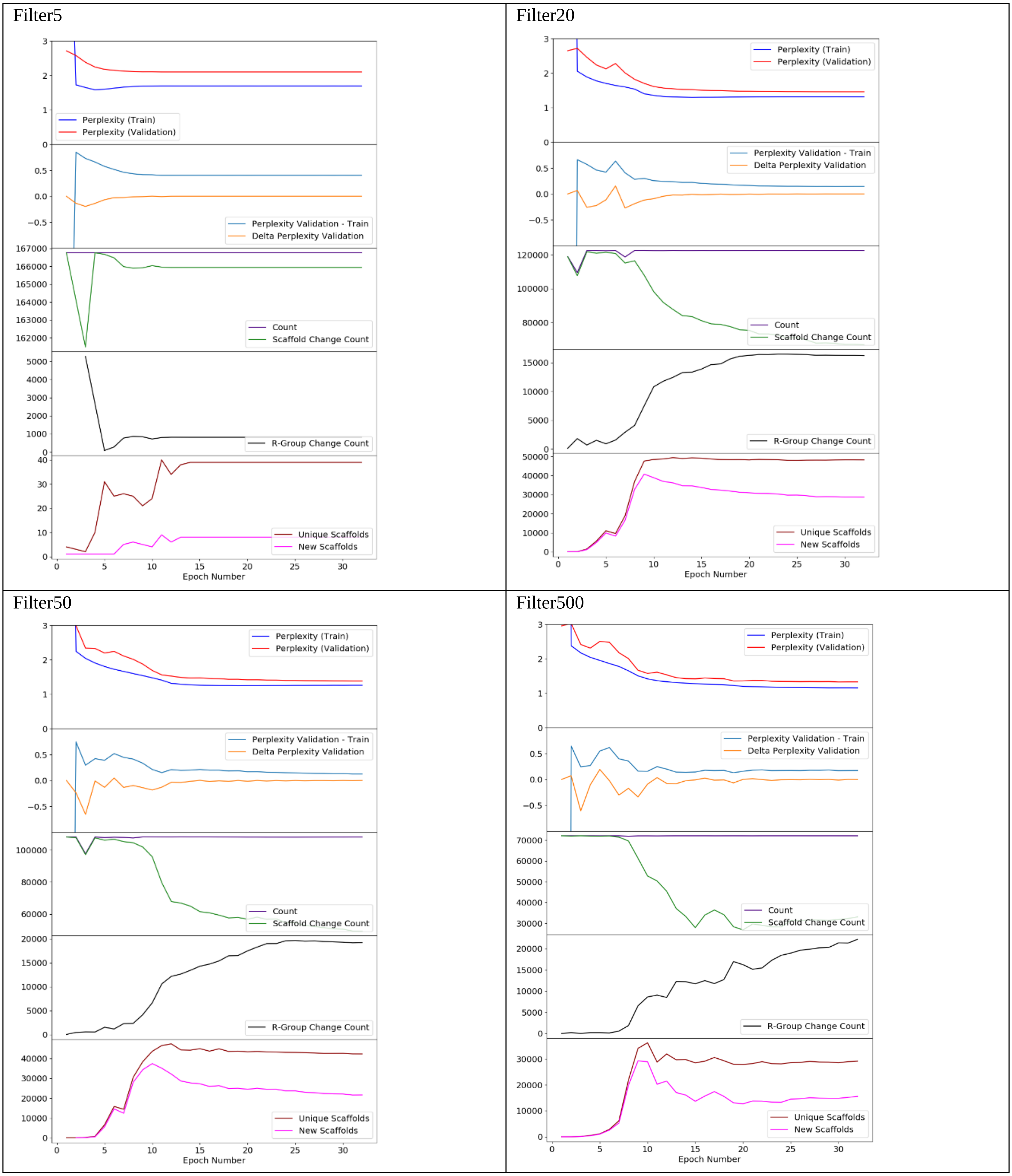

**Fig. S3.** In a wide range of values, the filter cutoff does not much affect the dynamics of information-theory-based and chemical scores of transformer models during training. The cutoff value of 50 ("*filter50*") was used to train the model presented in Results (Subsection 1) and Figs. 2, 3 and S2. Plots for the cutoff values of 20 and 500 demonstrate similar behavior, and only lowering the cutoff value to 5 causes deterioration of the model performance (see more detailed discussion in the main text). The notations are the same as in Fig. 3.