RESEARCH

Transformer Neural Network-Based Molecular Optimization Using General Transformations

Jiazhen He[1*], Eva Nittinger[2], Christian Tyrchan[2], Werngard Czechtizky[2], Atanas Patronov[1], Esben Jannik Bjerrum[1] and Ola Engkvist[1,3]

Abstract

Molecular optimization aims to improve the drug profile of a starting molecule. It is a fundamental problem in drug discovery but challenging due to (i) the requirement of simultaneous optimization of multiple properties and (ii) the large chemical space to explore. Recently, deep learning methods have been proposed to solve this task by mimicking the chemist's intuition in terms of matched molecular pairs (MMPs). Although MMPs is a typical and widely used strategy by medicinal chemists, it offers limited capability in terms of exploring the space of solutions. There are more options to modify a starting molecule to achieve desirable properties, *e.g.* one can simultaneously modify the molecule at different places including changing the scaffold. This study trains the same Transformer architecture on different datasets. These datasets consist of a set of molecular pairs which reflect different types of transformations. Beyond MMP transformation, datasets reflecting general transformations are constructed from ChEMBL based on two approaches: Tanimoto similarity (allows for multiple modifications) and scaffold matching (allows for multiple modifications but keep the scaffold constant) respectively. We investigate how the model behavior can be altered by tailoring the dataset while keeping the same model architecture. Our results show that the models trained on differently prepared datasets transform a given starting molecule in a way that it reflects the nature of the dataset used for training the model. These models could complement each other and unlock the capability for the chemists to pursue different options for improving a starting molecule.

Keywords: molecular optimization; matched molecular pairs; transformer; tanimoto similarity; scaffold; ADMET

Introduction

Molecular optimization aims to improve the property profile of a starting molecule. It plays an important role in the drug discovery and development process. However, this problem is challenging due to (i) the requirement of simultaneous optimization of multiple, often conflicting properties, *e.g.* physicochemical properties, ADMET (absorption, distribution, metabolism, elimination and toxicity) properties, safety and potency against its target and (ii) the large chemical space [1] to explore. Traditionally, chemists use their knowledge, experience and intuition [2] to apply chemical transformations to the starting molecule, to design improved molecules that have a balance of multiple properties. However, it heavily relies on chemist's knowledge and is often impacted by individual's biases. This can limit the design process and the opportunities to find improved molecules within a reasonable time scale.

Recently, various deep learning methods have been used and proposed for *de novo* molecular design, *e.g.* recurrent neural networks (RNNs) [3, 4, 5], variational autoencoders (VAEs) [6, 7, 8, 9, 10, 11] and generative adversarial networks (GANs) [12, 13, 14, 15]. To improve the generated molecules towards desirable properties, reinforcement learning [16, 12, 15, 13], adversarial training [17, 18, 19], transfer learning [3] and different optimization techniques [6, 20] have been used. Conditional generative models [8, 11, 21, 22] have also been proposed where the desirable properties are incorporated as condition to directly control the generating process. However, most of them focus on generating molecules from scratch. There are only a few studies on generating molecules with desirable properties from a given starting molecule, which aim to solve the molecular optimization task directly.

[*] Correspondence: jiazhen.he@astrazeneca.com
[1] Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden
Full list of author information is available at the end of the article

Most of them use a set of molecular pairs for training. Jin *et al.* [17, 23, 24] utilized molecular graph representations and viewed the molecular optimization problem as a graph-to-graph translation problem. He *et al.* [25, 26] instead utilized the string-based representation, the simplified molecular-input line-entry system (SMILES) [27] and employed the machine translation models [28, 29] from natural language processing (NLP). They trained machine translation models (Transformer and Seq2Seq) to mimic the chemist's approach of using matched molecular pairs (MMPs) [30, 31] where two molecules differ by a single chemical transformation. It was shown that the Transformer performs better than the Seq2Seq and HierG2G architectures [24].

Application of MMPs is a typical and widely used design strategy by medicinal chemists due to its interpretable and intuitive nature. However, MMPs offers limited capability in terms of exploring the space of solutions. Often more general transformations beyond the nature of MMPs are needed, *e.g.* simultaneous modifications of the starting molecule at different places including the core scaffold. In this study, the same Transformer architecture is trained on different datasets. These datasets consist of a set of molecular pairs, and are prepared to reflect different types of transformations. To capture more general transformations beyond MMPs, two approaches are used to extract molecular pairs from ChEMBL: Tanimoto similarity (allows for multiple modifications) and scaffold matching [32] (allows for multiple modifications but keeps the scaffold constant) respectively. The Transformer model trained on different datasets could unlock the capability for the chemists to pursue different options for improving a starting molecule.

Methods

Transformer Neural Network

Following [25], the SMILES representation of molecule and the Transformer model from NLP is used in our study. The Transformer is trained on a set of molecular pairs together with the property changes between source and target molecules. Three ADMET properties, *logD*, *solubility* and *clearance* which are important properties of a drug are selected to be optimized simultaneously. The property changes are encoded as the property constraint tokens which are included in the input sequence for guidance. Figure 1 shows an example of source and target sequences which are fed into the Transformer model during training. More details can be found in [25].

Given a set of molecular pairs $\{(X, Y, Z)\}$ where X represents source molecule, Y represents target molecule, and Z represents the property change between source molecule X and target molecule Y, the Transformer model will learn a mapping $(X, Z) \in \mathcal{X} \times \mathcal{Z} \to Y \in \mathcal{Y}$ during training where $\mathcal{X} \times \mathcal{Z}$ represents the input space and $\mathcal{Y}$ represents the target space. During testing, given a new $(X, Z) \in \mathcal{X} \times \mathcal{Z}$, the model will be expected to generate a diverse set of target molecules with desirable properties [25].

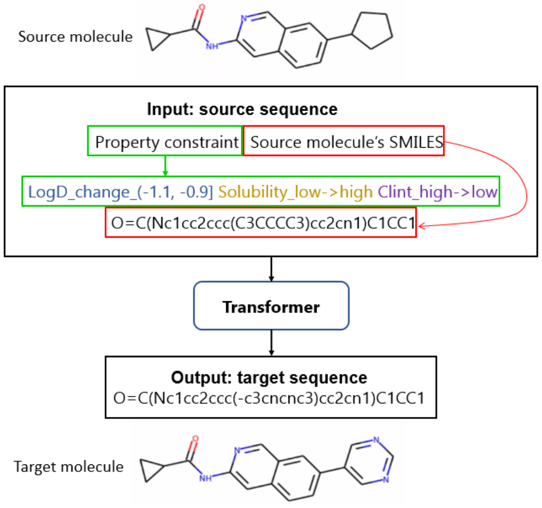

Source molecule
Input: source sequence
Property constraint
Source molecule's SMILES
LogD_change_(-1.1, -0.9] Solubility_low->high Clint_high->low
O=C(Nc1cc2ccc(C3CCCC3)cc2cn1)C1CC1
Transformer
Output: target sequence
O=C(Nc1cc2ccc(-c3cncnc3)cc2cn1)C1CC1
Target molecule

Figure 1 Input and output of the Transformer model (following [25]). The input is the concatenation of property change tokens and the SMILES of the starting molecule. During training, the output is the target molecule with the desirable properties while during inference the output is generated token by token and is expected to satisfy the property constraint in the input.

Data Preparation

The datasets[1] consist of a set of molecular pairs extracted from ChEMBL 28 [33]. In particular, the pairs were extracted from the molecules that are originated from the same publication since the molecules are more likely to be in the same project. Therefore, the molecular pairs are more likely to reflect the chemist's intuition. The molecules, publications and molecular pairs are processed in the following fashion,

Molecule pre-processing

- Standardization using MolVS [2]: Keep uncharged version of the largest fragment; Sanitize; RemoveHs; Disconnect metals; Apply normalization rules; Reionize acids; Keep sterochemistry
- $10 \leq$ Number of heavy atoms ≤ 50
- Number of rings > 0
- AZFilter="CORE" [34] to filter out low-quality compounds

[1] https://doi.org/10.5281/zenodo.5707626

[2] https://molvs.readthedocs.io/en/latest/

- Substructure filters [35] for hit triaging with SeverityScore<10 [3]
- Each molecule's property values are within 3 standard deviations of all molecules' property values (predicted)

Publication pre-processing

- Year $\geq$ 2000
- 10 $\leq$ Number of molecules $\leq$ 60

Molecular pair pre-processing

- Remove duplicated pairs (keep the earliest reported)
- Include reverse pairs

The data statistics can be found in Supplementary Figure S1.

Constructing Molecular Pairs.

To capture different types of transformations, the following criteria are considered for extracting the pairs from different perspectives.

MMP. The matched molecular pairs are two molecules differ by a single transformation, which has been widely used as a strategy by medicinal chemists to support molecular optimization. Here, the MMPs are extracted using mmpdb, an open-source matched molecular pair tool [36]. The ratio between the number of heavy atoms (non-hydrogen atoms) in the R-group and the number of heavy atoms in the entire molecule is not greater than 0.33 [37].

To capture more general transformations (*e.g.* multiple modifications), apart from single transformations, the following criteria are used,

Tanimoto similarity. The Tanimoto similarity is computed based on Morgan Fingerprint with radius=2 (ECFP4) using RDKit. Figure 2 shows the distribution of Tanimoto similarity between all the possible unique pairs originating from the same publication. We extract the molecular pairs based on the following thresholds,

- Similarity ($\geq$0.5) for similar molecules
- Similarity ([0.5,0.7)) for medium similar molecules
- Similarity ($\geq$0.7) for highly similar molecules

Scaffold matching. For the molecules originating from the same publication, if two molecules share the same scaffold then they are extracted as pairs. In particular, the Murcko scaffold from RDKit which removes the side chains and the Murcko scaffold generic which converts all atom types to C and all bonds to single are used. The top 20 frequently occurring scaffold and generic scaffold can be found in Supplementary Figure S2 and Fig. S3.

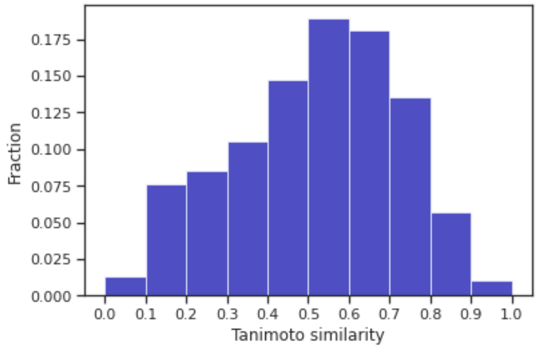


Figure 2 Tanimoto similarity distribution considering all the possible unique pairs with the same publication.

Table 1 Dataset

Datasets	Training (2000-2017)	Validation (2018)	Test (2019-2020)
MMPs	2,287,588	143,978	166,582
Similarity ($\geq$0.5)	6,543,684	418,180	475,070
Similarity ([0.5,0.7))	4,543,472	286,682	327,606
Similarity ($\geq$0.7)	2,000,212	131,498	147,464
Scaffold	2,850,180	171,914	199,786
Scaffold generic	4,127,058	255,580	289,034

Table 1 shows the resulting datasets (all datasets include reverse pairs). The training, validation and test sets are split based on the year of the publications from which the pairs are extracted. The Transformer neural network is trained on each dataset, and is expected to transform the input molecule in a way that it reflects the nature of the dataset used for training the model.

ADMET Property Prediction Model

The input of our Transformer model takes the property changes of molecular pairs into account. Here, we employ property prediction models due to the limited experimental data from ChEMBL. In particular, we build our ADMET property prediction models based on in-house experimental data using message passing neural network [38]. The property prediction models are used for obtaining the properties of molecular pairs therefore the property changes for the dataset and also for evaluating the generated molecules during test. Table 2 shows the train and test size, root-mean-square error (RMSE), normalized RMSE (NRMSE) and R^2 for each property prediction model.

Experimental Settings

For each starting molecule in the test set, 10 unique valid molecules, which are different from the starting molecule, were generated using multinomial sampling.

[3] https://github.com/rdkit/rdkit/tree/master/Contrib/NIBRSubstructureFilters

Table 2 Property prediction model performance

	LogD	Solubility	Clearance
Train size	186,575	197,988	155,652
Train RMSE	0.295	0.489	0.271
Train NRMSE	0.025	0.056	0.053
Train R^2	0.942	0.775	0.76
Test size	20,731	21,999	17,295
Test RMSE	0.395	0.600	0.352
Test NRMSE	0.038	0.076	0.091
Test R^2	0.897	0.659	0.555

Evaluation Metrics

The models are evaluated in two main aspects,

- **Successful property constraints** gives the percentage of generated molecules that fulfill the three desirable properties specified by model input simultaneously. The ADMET property prediction model in Table 2 is used to compute the properties of generated molecules. Following [25], the model error (Test RMSE in Table 2) is considered to determine if a generated molecule satisfies its desirable properties. For *logD*, the generated molecules with $|logD_{generated} - logD_{target}| \leq 0.4$ will be considered as satisfying desirable *logD* constraint. For *solubility*, the threshold for low and high will be a range considering the model error, *i.e.* 1.7±0.6. The generated molecules with *solubility* $\leq$ 2.3 will be considered as low, and those with *solubility* $\geq$ 1.1 will be considered as high. Similarly, for *clearance*, the threshold is 1.3±0.35.
- **Successful structure constraints** gives the percentage of generated molecules that when comparing with their corresponding starting molecules, have the same structure constraints as the pairs in the training set. This differs according to datasets, *e.g.* for the MMPs dataset, this metric gives the percentage of generated molecules that are matched molecular pairs with their starting molecules while for the Similarity ($\geq$0.5) dataset, the structure constraint is that the Tanimoto similarity between the generated molecules and their corresponding starting molecules is between 0.5 and 1.0. This metric evaluates if the model has learned to use the type of transformation reflected in the training set to modify starting molecules.

Baselines

We compare our model Transformer with the following baselines,

- **Transformer-U** is the unconditional Transformer architecture trained on molecular pairs but without any input property constraints.
- **Random** randomly selects 10 molecules (for a direct comparison with our Transformer model where 10 molecules are generated) from the unique set of molecules in the test set that have the same structure constraint as the training set. For example, for the Scaffold dataset, it randomly select 10 molecules that share the same scaffold with the given starting molecule. Since it is computationally expensive to evaluate all the samples (each sample consist of a starting molecule desirable property changes) in the test set, we randomly select 1% of the test set, repeat 5 times with different sampling seeds and report the average results.

Results and Discussion

Data Statistics

Figure 3 shows the overlap of training molecular pairs among different datasets. Almost all the MMPs are in the dataset of pairs with Similarity ($\geq$0.5). The overlap between the MMP dataset and the Similarity ($\geq$0.7) dataset is bigger than the one between the MMP dataset and the Similarity ([0.5,0.7)) dataset. Exemplar molecular pairs only in dataset Similarity ($\geq$0.5) show that the scaffold is changed compared to pairs sharing generic scaffold and are non-MMPs because of multiple modifications and/or big change in R-group. The molecular pairs only in scaffold generic have Tanimoto similarity below 0.5. A tiny proportion of MMPs have Tanimoto similarity below 0.5 and change the scaffold.

Performance Comparison with Baselines

Table 3 compares our Transformer model with the baselines (Transformer-U and Random) in terms of successful property and structure constraints on different datasets. Transformer outperforms Transformer-U and Random in terms of successful property constraints, generating more molecules with desirable properties on all datasets. For the successful structure constraints, Transformer-U is comparable or better than Transformer. Transformer-U has learned to generate “similar” molecules to the given input starting molecules. However, it generates much less molecules with desirable properties compared to Transformer. It is mainly because Transformer-U was trained only on molecular pairs, and does not include the property change of the pairs in the input, while Transformer having the property changes as additional input, allows for more directed output generation. Both Transformer and Transformer-U outperform the Random baseline - finding more molecules that satisfy desirable properties and structure constraint simultaneously.

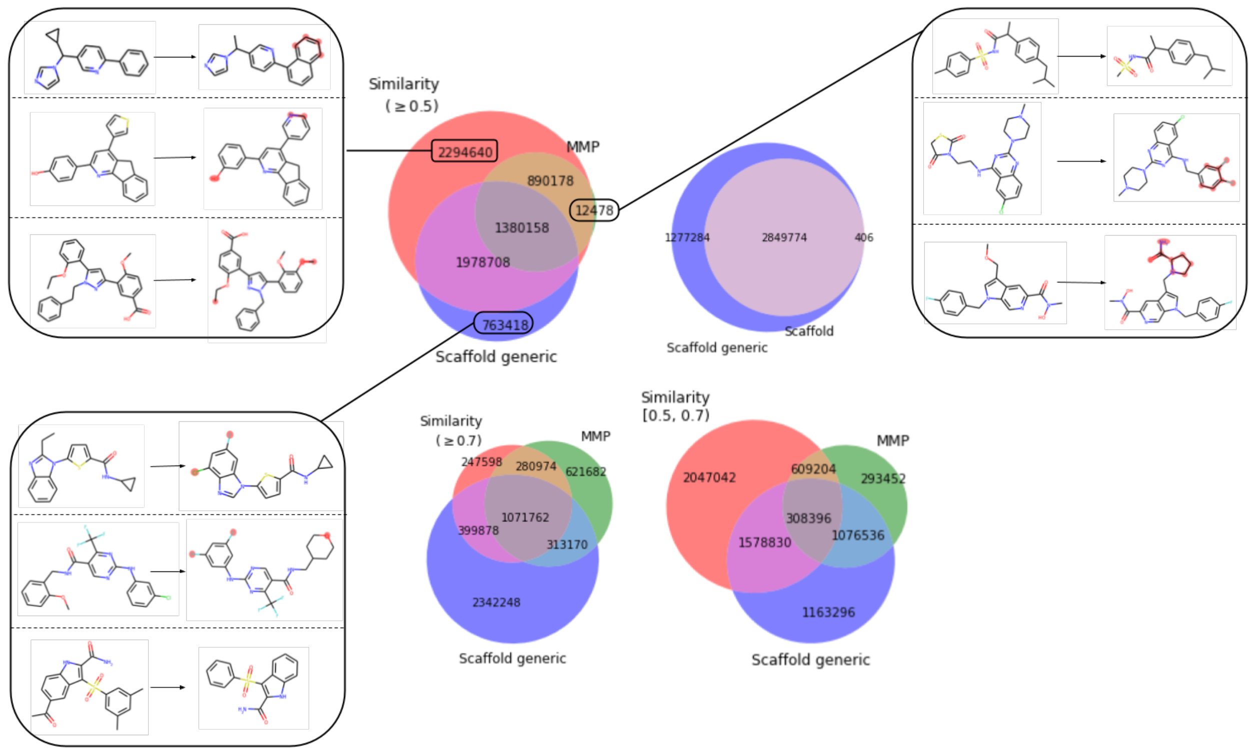


Figure 3 Overlap of training molecular pairs among different datasets. Exemplar molecular pairs are shown for data only in dataset Similarity (≥0.5), scaffold generic and MMP respectively.

Table 3 Performance comparison of Transformer and baselines in terms of successful property constraints, successful structure constraints and both metrics simultaneously. Each model is trained on the corresponding dataset for that row.

Dataset	Model	Successful property constraints (%)	Successful structure constraints (%)	Successful property and structure constraints (%)
MMP	Transformer	**61.90**	91.55	**58.09**
	Transformer-U	33.67	93.25	31.85
	Random	13.44±0.43	100	13.44±0.43
Similarity (≥0.5)	Transformer	**51.83**	82.30	**44.53**
	Transformer-U	29.04	83.63	25.32
	Random	15.17±0.27	100	15.17±0.27
Similarity ([0.5,0.7))	Transformer	**46.75**	68.09	**32.96**
	Transformer-U	26.23	69.13	18.72
	Random	14.57±0.37	100	14.57±0.37
Similarity (≥0.7)	Transformer	**65.09**	82.68	**56.07**
	Transformer-U	39.57	84.83	34.70
	Random	11.48±0.29	100	11.48±0.29
Scaffold	Transformer	**61.53**	95.32	**59.69**
	Transformer-U	37.16	95.69	36.26
	Random	17.22±0.74	100	17.22±0.74
Scaffold generic	Transformer	**55.05**	96.01	**53.66**
	Transformer-U	32.55	96.30	31.69
	Random	16.48±0.41	100	16.48±0.41

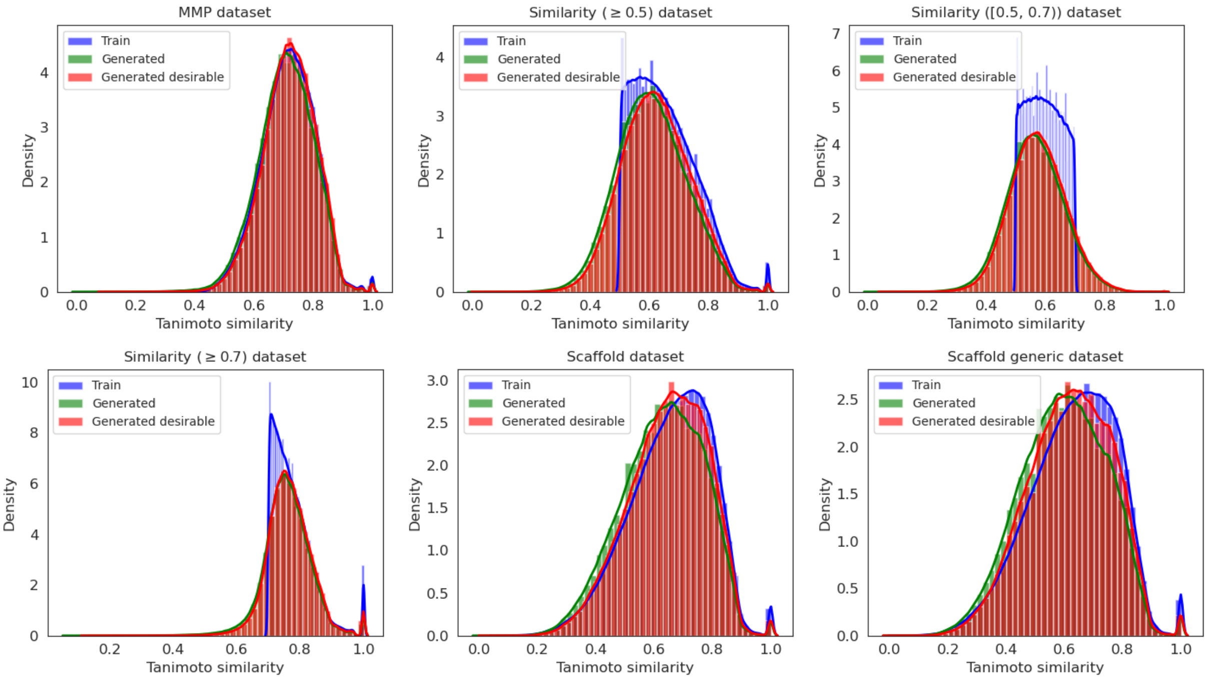


Figure 4 Tanimoto similarity distribution. Blue for the molecular pairs on the training set; green for the pairs between the generated molecules and their starting molecules on the test set; red for the pairs between the generated molecules that fulfil the desirable properties specified in the input and their starting molecules on the test set.

Figure 4 compares the Tanimoto similarity distribution of the molecular pairs on the training set with the one between the generated molecules and their starting molecules on the test set for the Transformer model. It can be seen that the distribution of the generated pairs align well with the pairs on the training set for most datasets. For the datasets based on Tanimoto similarity, the alignment is worse, but the model systematically generates molecules which fulfil the desirable properties even though the Tanimoto similarity between generated molecules and the corresponding starting molecules are outside the constrains of the training set (see the overlap between green and red distribution), indicating the model can extrapolate the learning beyond the structure constraints defined by the training data. Additionally, the overlap between generated and train on scaffold-based dataset is not as good as the one on MMP dataset, but in terms of successful property constraints, scaffold-based datasets are slightly better than MMP dataset, indicating it is relative easy to keep scaffolds than MMPs structure constraint for the Transformer model.

Performance Comparison of Models Trained on Different Types of Molecular Pairs

With the following experiments, we evaluate how the models trained on different types of molecular pairs perform on the same test sets. Figure 5 shows the overlap of the original test sets (Table 1) among MMP, Similarity (≥0.5) and Scaffold generic datasets. Here, five test sets are extracted based on Figure 5 for model comparison, shown in Table 4. Table 5 shows the comparison results.

Firstly, for the restricted intersection dataset, the model trained on MMP dataset performs best in terms of successful property constraints, followed closely by the one trained on Similarity (≥0.7) dataset, while the model trained on Similarity ([0.5, 0.7)) dataset performs worst. This might because the molecular pairs in the restricted intersection dataset have smaller structural changes and desired property changes, and it is easier to achieve small desirable property changes by making small structural changes. It might also be because of the varying performance of the models trained on different types of molecular pairs in the beginning (Table 3). Therefore we also report the difference (numbers in bracket) compared to their performance on their original test sets (Table 3). We can see that most models perform better compared to the performance on their own original test set, indicating this restricted intersection dataset is an relative easy task. The performance change of the models trained on Similarity (≥0.7) and Scaffold are very small, indicating there is not much difference between this restricted

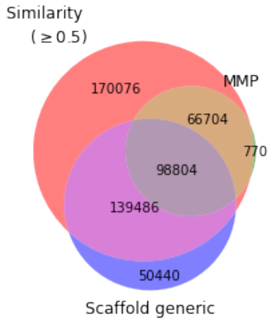


Figure 5 Overlap of molecular pairs among different test sets, MMP, Similarity (≥0.5), Scaffold generic datasets, used for extracting test sets (Table 4) for model comparison.

Table 4 Test sets extracted for model comparison. **Restricted intersection** represents the overlapping among MMP, Similarity (≥0.5) and Scaffold generic test sets; **Merged** represents the union of MMP, Similarity (≥0.5) and Scaffold generic test sets; **MMP only** represents the set of molecular pairs that appear only in MMP test set not the other two; Similarly for Similarity (≥0.5) only and Scaffold generic only.

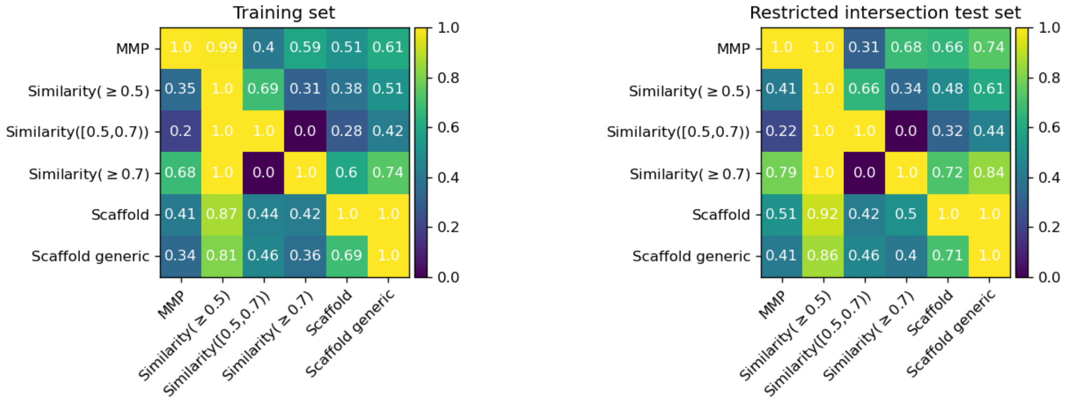
Test set	Size
Restricted intersection	98,804
MMP only	770
Similarity (≥0.5) only	170,076
Scaffold generic only	50,440
Merged	526,584

Table 5 Performance comparison of the Transformer models trained on different types of molecular pairs on different test sets (numbers in bracket represent the absolute increase or decrease compared to the corresponding Transformer model performance on the original test set in Table 3). The extremes (best/worst performance or largest/smallest change) are highlighted in bold.

Test set	Type of molecular pairs where Transformer is trained	Successful property constraints (%)	Successful structure constraints (%)	Successful property and structure constraints (%)
Restricted intersection	MMP	**65.71** (↑ 3.81)	91.68 (↑ 0.13)	**61.82** (↑ 3.73)
	Similarity (≥0.5)	55.55 (↑ 3.72)	84.47 (↑ **2.17**)	48.97 (↑ **4.44**)
	Similarity ([0.5,0.7))	**50.17** (↑ 3.42)	**68.66** (↑ 0.57)	**35.28** (↑ 2.32)
	Similarity (≥0.7)	65.39 (↑ **0.30**)	81.49 (↓ **1.19**)	55.55 (↓ 0.52)
	Scaffold	62.91 (↑ 1.38)	94.42 (↓ 0.90)	60.70 (↓ **1.01**)
	Scaffold generic	59.07 (↑ **4.02**)	**96.14** (↑ 0.13)	57.68 (↑ 4.02)
MMP only	MMP	**50.01** (↓ 11.89)	85.48 (↓ 6.07)	**43.14** (↓ 14.95)
	Similarity (≥0.5)	44.61 (↓ 7.22)	78.03 (↓ 4.27)	35.78 (↓ 8.75)
	Similarity ([0.5,0.7))	**41.88** (↓ **4.87**)	**65.57** (↓ **2.52**)	**27.94** (↓ **5.02**)
	Similarity (≥0.7)	45.48 (↓ **19.61**)	68.53 (↓ **14.15**)	33.78 (↓ **22.29**)
	Scaffold	44.83 (↓ 16.70)	87.47 (↓ 7.85)	40.86 (↓ 18.83)
	Scaffold generic	42.21 (↓ 12.84)	**88.81** (↓ 7.20)	38.60 (↓ 15.06)
Similarity (≥0.5) only	MMP	**56.43** (↓ 5.47)	86.93 (↓ 4.68)	**50.51** (↓ 7.59)
	Similarity (≥0.5)	49.57 (↓ 2.26)	81.49 (↓ 0.81)	42.09 (↓ 2.44)
	Similarity ([0.5,0.7))	**45.75** (↓ **1.00**)	**67.98** (↓ **0.11**)	**32.09** (↓ **0.87**)
	Similarity (≥0.7)	55.08 (↓ **10.01**)	78.87 (↓ 3.81)	45.24 (↓ **10.83**)
	Scaffold	52.94 (↓ 8.59)	88.48 (↓ **6.84**)	48.70 (↓ 7.99)
	Scaffold generic	49.75 (↓ 5.30)	**90.11** (↓ 5.90)	46.06 (↓ 7.60)
Scaffold generic only	MMP	**49.65** (↓ 12.25)	87.77 (↓ 3.78)	44.84 (↓ 13.25)
	Similarity (≥0.5)	46.21 (↓ 5.62)	77.17 (↓ 5.13)	36.94 (↓ 7.59)
	Similarity ([0.5,0.7))	**43.38** (↓ **3.37**)	**64.78** (↓ 3.31)	**28.88** (↓ **4.08**)
	Similarity (≥0.7)	47.53 (↓ **17.56**)	74.83 (↓ **7.85**)	37.33 (↓ **18.74**)
	Scaffold	48.86 (↓ 12.67)	94.85 (↓ 0.47)	47.19 (↓ 9.50)
	Scaffold generic	47.07 (↓ 7.98)	**96.26** (↑ **0.25**)	**45.89** (↓ 7.77)
Merged	MMP	**58.60** (↓ 3.3)	88.75 (↓ 2.80)	**53.51** (↓ 4.58)
	Similarity (≥0.5)	51.29 (↓ 0.54)	81.76 (↓ 0.54)	43.77 (↓ 0.76)
	Similarity ([0.5,0.7))	**47.16** (↑ **0.41**)	**67.88** (↓ **0.21**)	**33.01** (↑ **0.05**)
	Similarity (≥0.7)	57.48 (↓ **7.61**)	79.23 (↓ 3.45)	47.50 (↓ **8.57**)
	Scaffold	55.54 (↓ 5.99)	91.33 (↓ **3.99**)	52.25 (↓ 7.44)
	Scaffold generic	52.43 (↓ 2.62)	**92.95** (↓ 3.06)	49.84 (↓ 3.82)

dataset and their own original test set in terms of difficulty.

Secondly, we check the performance on the test sets, MMP only, Similarity (≥0.5) only and Scaffold generic



(a) Relationship between the training molecular pairs of different datasets

(b) Generated molecules from different models (row) in terms of satisfying different structure constraints (column)

Figure 6 Comparison of heatmaps for training set and test set. The more similar, the better. (a) Relationship between the training molecular pairs of different datasets, *e.g.* the number 0.2 with Similarity ([0.5, 0.7)) as row and MMP as column on the training set represents 20% of the pairs with Similarity ([0.5, 0.7)) are also MMPs. (b) Each row represents the model trained on the corresponding dataset, and each column represents the corresponding structure constraints. The number 0.22 with Similarity ([0.5, 0.7)) as row and MMP as column on the Restricted intersection test set represents that when looking at the generated molecules using the Transformer model trained on Similarity ([0.5, 0.7)) dataset, among all the ones fulfilling the the property constraints and structure constraints (*i.e.* Similarity ([0.5, 0.7))), 22% of them are MMPs. The diagonal on the Restricted intersection is always 1 because we only look at the generated molecules that already fulfil the property constraints and structure constraints.

only. Unlike the performance on the restricted intersection dataset which mostly increases compared to the original test sets, the performance on these three test sets drops significantly, indicating they are more difficult tasks than the original test sets. The performance of models trained on Similarity (≥0.7) and Scaffold drop the most (see bracket) in most cases. This might be because it is difficult to achieve desired properties by keeping high similarity or same scaffold. While for the model that trained on Similarity (≥0.5), it has the worst performance but drops the least in most cases. This might because that the molecular pairs are less restrictive, but also make them more difficult to train. Lastly, the performance on the merged test set lie between the one on the restricted intersection test set and the ones on other test sets. For the models trained on Similarity (≥0.5) and Similarity ([0.5, 0.7)), there is not much difference (see bracket) compared to the performance on their own original test set, while the models trained on Similarity (≥0.7) and Scaffold are more sensitive to the test sets.

Even though the model trained on MMP dataset performs best for most datasets, it is not good enough to generate molecules that fulfill property and structure constraints and is preferable to have more alternatives. Figure 6a shows how the training molecular pairs from different datasets correlate with each other. For example, 40% of MMPs (row) are also pairs with Similarity ([0.5, 0.7)) (column) but only 20% of pairs with Similarity ([0.5, 0.7)) (row) are MMPs (column). Figure 6b shows that for the restricted intersection test set, how the generated molecules from models trained on different datasets satisfy different structure constraints. For example, among the generated molecules (that satisfy the property constraints and structure constraints, *i.e.* Similarity ([0.5, 0.7))) from the model trained on Similarity ([0.5, 0.7)) (row), 22% of them are MMPs when comparing with their corresponding starting molecules. Compared to the heatmap for the training set, the one for Restricted intersection test set basically follow the same pattern (similar patterns are found on other test sets), indicating the models have learned to modify the starting molecules in the way that it reflects the nature of the training set. Overall, it is shown that there is no single model generating molecules that cover the ones from all other models. It could be beneficial to use an ensemble of these models which complement each other to provide different options to transform a starting molecule towards desirable properties.

Performance on Test Sets with Large Property Changes Desired

With the following experiments, we evaluate how the models trained on different types of molecular pairs perform on the test sets where large property changes (*logD* change is above 1; *solubility* and *clearance* change is either low→high or high→low) are desired. The molecular pairs in the original test sets where large property changes are extracted and merged excluding duplicates. Table 6 shows that 4.6% (highest) of the Similarity ([0.5, 0.7)) dataset has large property changes desired while Similarity (≥0.7) dataset has the lowest, 2.3%. It is reasonable because it is less likely to have large property changes while keeping higher structural similarity.

Table 6 Test sets where big property changes (*logD* change is above 1; *solubility* and *clearance* change is either low→high or high→low) are desired. Size indicates the number of data points where big property change are desired; Percentage indicates the fraction of the original test set in Table 1 with data points that have big property changes, *e.g.* 6180/166582≈3.7%.

Test set	Size	Percentage (%)
MMP	6,180	3.7
Similarity (≥0.5)	18,546	3.9
Similarity ([0.5, 0.7))	15,130	4.6
Similarity (≥0.7)	3,416	2.3
Scaffold	6,252	3.1
Scaffold generic	10,514	3.6
Merged	21,652	-

Table 7 shows the results on the merged dataset (the results on other datasets in Table 6 can be found in Supplementary Table S1). All models perform worse compared to their performance on their original test set (Table 3). The reason is that only a small proportion of molecular pairs having large property changes in the training set (Supplementary Figure S4), therefore the models generalize less well on such pairs. Intuitively, it would be expected that the model trained on Similarity ([0.5, 0.7)) dataset would perform best since it has higher percentage of pairs with large property changes for training and have more freedom to modify the starting molecule. However, it is observed that the model trained on MMPs performs best. This might because it is easier to train the Transformer model for MMPs compared to pairs with similarity ([0.5, 0.7)) (already seen in Table 3) due to the smaller extrapolated space. Having that said, the performance of the models trained on different types of molecular pairs differ less on this Merged test set where big property changes are desired compared to previous test sets ((Table 3 and Table 5). When looking at the numbers in bracket, we observed that the performance of model trained on Similarity ([0.5, 0.7)) drop the least, while the one for Similarity (≥0.7) drop the most, followed by Scaffold and MMP.

Example of Diverse Molecules Generated using Models Trained on Different Types of Molecular Pairs

Figure 7 and Figure 8 show an example of the generated molecules that fulfill the desirable properties but modify the starting molecule in different ways depending on the training data used for training the model. In particular, the generated molecules in Figure 7b make a single transformation to the starting molecule while the ones in Figure 8c and Figure 8d allow for multiple modifications but keep the scaffold or generic scaffold constant. The generated molecules in Figure 7c, 7d and 8b allow for multiple modifications and changes in scaffold, but the Tanimoto similarity lies approximately [0.5, 1.0], [0.7, 1.0] and [0.5, 0.7) respectively. Overall, this shows the flexibility of modifying starting molecules to achieve desirable properties in different ways by using the models trained on different types of molecular pairs.

Discussion

Varying performance of models trained on different types of molecular pairs

The Transformer models trained on different datasets show varying performance as shown in Table 3. On the MMP, scaffold and scaffold generic datasets, it is easier to generate molecules in terms of successful structure constrains (MMPs, sharing same scaffold) compared to the datasets based on Tanimoto similarity split. This might be because the pairs in the Tanimoto similarity based datasets have more variations, and the models have more freedom to extrapolate which makes it difficult to keep the same structure constraints. It might also be due to the hard Tanimoto similarity cutoff used for constructing the training set (Figure 4), which is difficult for the generated molecules from the Transformer model to follow on.

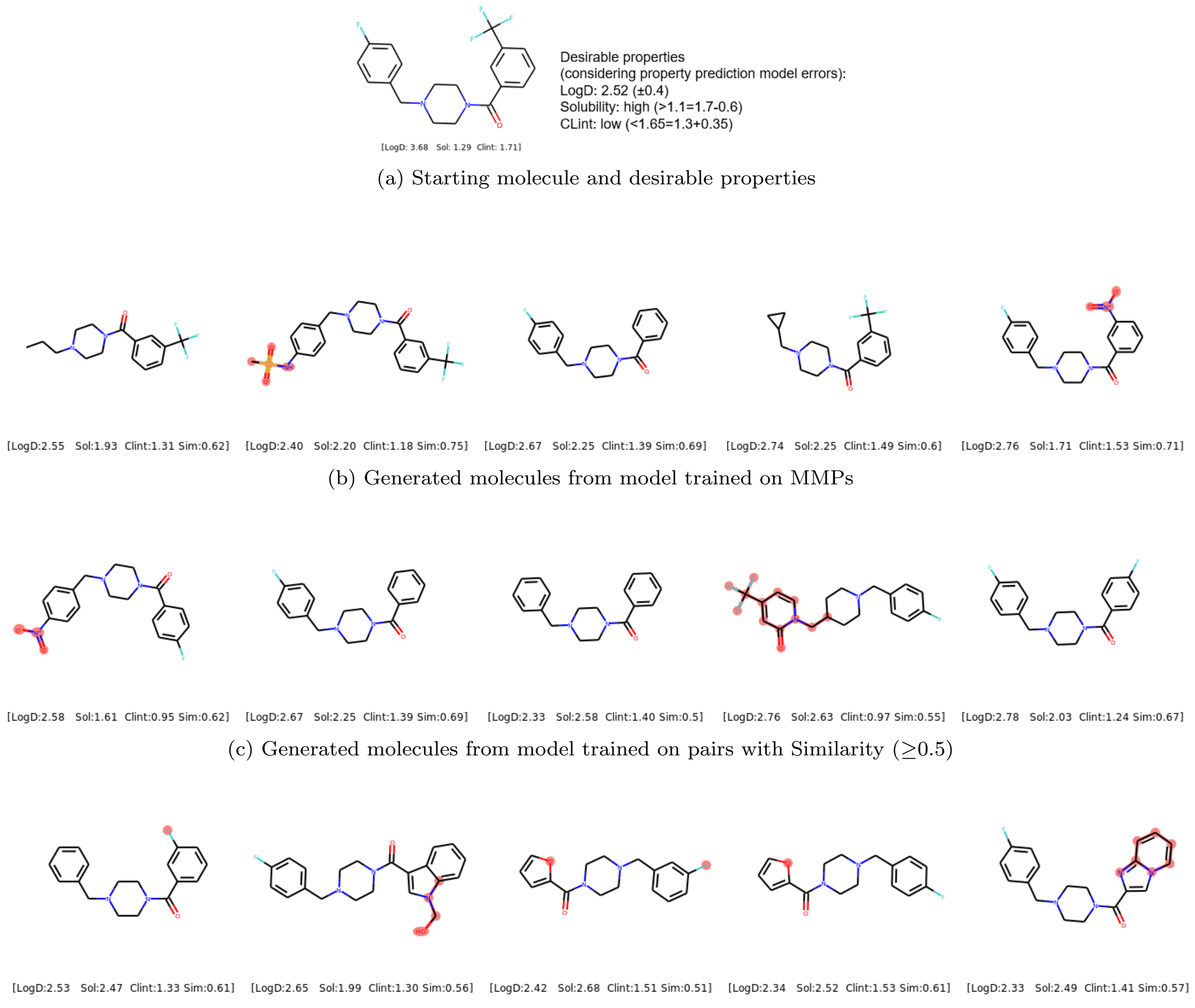
In terms of successful property constrains, Similarity (≥0.7) dataset has the best performance, followed by MMP and scaffold, which are much better than Similarity ([0.5,0.7)), Similarity (≥0.7) and scaffold generic. The reason might be that the extrapolated space is larger which makes it harder to find molecules with desirable properties. It might also be because the molecular pairs are more similar and the property changes are smaller for Similarity (≥0.7), MMP and scaffold dataset (Supplementary Figure S4).

Varying performance in terms of successful structure constraints and successful property constraints

It is observed from Table 3 that the Transformer model's performance in terms of successful structure

Table 7 Performance comparison of Transformer models trained on different types of molecular pairs on the Merged dataset where big property changes are desired (numbers in bracket represent the absolute increase/decrease compared to the corresponding Transformer model performance on the original test set in Table 3). The extremes (best/worst performance or largest/smallest change) are highlighted in bold.

Test set	Type of molecular pairs where Transformer is trained	Successful property constraints (%)	Successful structure constraints (%)	Successful property and structure constraints (%)
Merged	MMP	**40.82** (↓ 21.08)	83.89 (↓ 7.66)	**36.12** (↓ 21.97)
	Similarity (≥0.5)	39.81 (↓ 12.02)	75.00 (↓ 7.30)	30.70 (↓ 13.83)
	Similarity ([0.5,0.7))	38.33 (↓ **8.42**)	**66.64** (↓ **1.45**)	25.94 (↓ **7.02**)
	Similarity (≥0.7)	**36.14** (↓ **28.95**)	68.57 (↓ **14.11**)	**25.58** (↓ **30.49**)
	Scaffold	36.50 (↓ 25.03)	89.17 (↓ 6.15)	33.60 (↓ 23.09)
	Scaffold generic	37.78 (↓ 17.27)	**91.30** (↓ 4.71)	35.26 (↓ 18.40)


[LogD: 3.68 Sol: 1.29 Clint: 1.71]
Desirable properties
(considering property prediction model errors):
LogD: 2.52 (±0.4)
Solubility: high (>1.1=1.7-0.6)
CLint: low (<1.65=1.3+0.35)
[LogD:2.55 Sol:1.93 Clint:1.31 Sim:0.62]
[LogD:2.40 Sol:2.20 Clint:1.18 Sim:0.75]
[LogD:2.67 Sol:2.25 Clint:1.39 Sim:0.69]
[LogD:2.74 Sol:2.25 Clint:1.49 Sim:0.6]
[LogD:2.76 Sol:1.71 Clint:1.53 Sim:0.71]
[LogD:2.58 Sol:1.61 Clint:0.95 Sim:0.62]
[LogD:2.67 Sol:2.25 Clint:1.39 Sim:0.69]
[LogD:2.33 Sol:2.58 Clint:1.40 Sim:0.5]
[LogD:2.76 Sol:2.63 Clint:0.97 Sim:0.55]
[LogD:2.78 Sol:2.03 Clint:1.24 Sim:0.67]
[LogD:2.53 Sol:2.47 Clint:1.33 Sim:0.61]
[LogD:2.65 Sol:1.99 Clint:1.30 Sim:0.56]
[LogD:2.42 Sol:2.68 Clint:1.51 Sim:0.51]
[LogD:2.34 Sol:2.52 Clint:1.53 Sim:0.61]
[LogD:2.33 Sol:2.49 Clint:1.41 Sim:0.57]

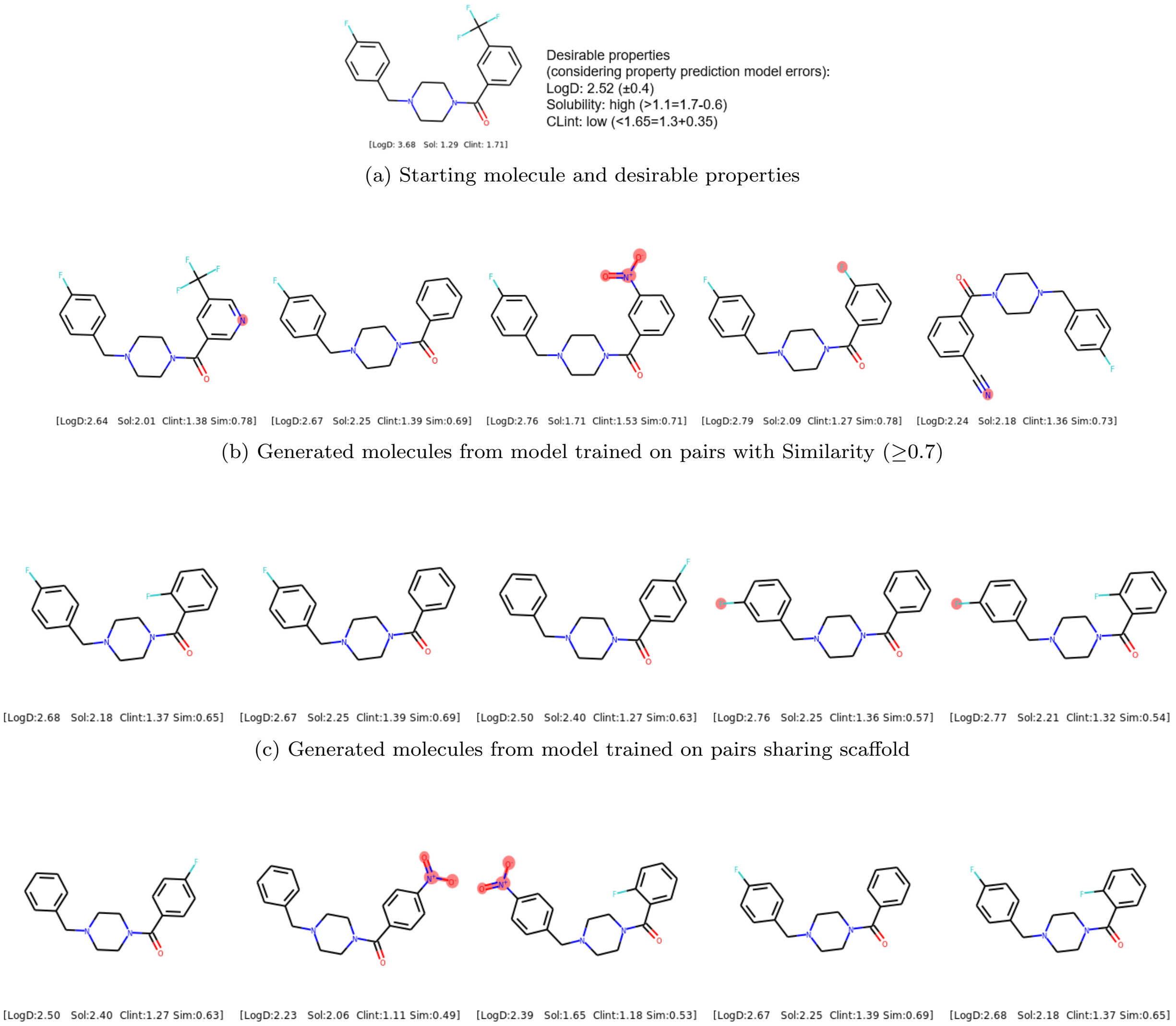
(a) Starting molecule and desirable properties

(b) Generated molecules from model trained on MMPs

(c) Generated molecules from model trained on pairs with Similarity (≥0.5)

(d) Generated molecules from model trained on pairs with Similarity ([0.5,0.7))

Figure 7 Example of diverse molecules with desirable properties generated by models trained on (b) MMPs (c) pairs with Similarity (≥0.5) (d) pairs with Similarity ([0.5, 0.7)). The changes in the generated molecules compared with starting molecule are highlighted in red. Sim represents Tanimoto similarity.


Desirable properties
(considering property prediction model errors):
LogD: 2.52 (±0.4)
Solubility: high (>1.1=1.7-0.6)
CLint: low (<1.65=1.3+0.35)
[LogD: 3.68 Sol: 1.29 Clint: 1.71]
[LogD:2.64 Sol:2.01 Clint:1.38 Sim:0.78]
[LogD:2.67 Sol:2.25 Clint:1.39 Sim:0.69]
[LogD:2.76 Sol:1.71 Clint:1.53 Sim:0.71]
[LogD:2.79 Sol:2.09 Clint:1.27 Sim:0.78]
[LogD:2.24 Sol:2.18 Clint:1.36 Sim:0.73]
[LogD:2.68 Sol:2.18 Clint:1.37 Sim:0.65]
[LogD:2.67 Sol:2.25 Clint:1.39 Sim:0.69]
[LogD:2.50 Sol:2.40 Clint:1.27 Sim:0.63]
[LogD:2.76 Sol:2.25 Clint:1.36 Sim:0.57]
[LogD:2.77 Sol:2.21 Clint:1.32 Sim:0.54]
[LogD:2.50 Sol:2.40 Clint:1.27 Sim:0.63]
[LogD:2.23 Sol:2.06 Clint:1.11 Sim:0.49]
[LogD:2.39 Sol:1.65 Clint:1.18 Sim:0.53]
[LogD:2.67 Sol:2.25 Clint:1.39 Sim:0.69]
[LogD:2.68 Sol:2.18 Clint:1.37 Sim:0.65]

(a) Starting molecule and desirable properties

(b) Generated molecules from model trained on pairs with Similarity (≥0.7)

(c) Generated molecules from model trained on pairs sharing scaffold

(d) Generated molecules from model trained on pairs sharing generic scaffold

Figure 8 Example of diverse molecules with desirable properties generated by models trained on (b) pairs with Similarity (≥0.7) (c) pairs sharing scaffold and (d) pairs sharing generic scaffold. The changes in the generated molecules compared with starting molecule are highlighted in red. Sim represents Tanimoto similarity.

constraints is better than successful property constraints. This might be because it is a relative easy task to keep the same structure constraint as in the training set. While for successful property constraints, it is more restricted due to the requirement of satisfying three properties simultaneously and the *logD* change is encoded at a higher level of granularity (considering the practical use) compared to *solubility* and *clearance* change which only have three possible changes (see [25] for further details). This makes the input space more complicated and bigger, which requires more data to build a good model and makes it harder to generalize well.

Conclusions

We utilized Transformer neural network to mimic more generally chemist's intuition that goes beyond MMPs for molecular optimization. We investigated how the model behavior can be altered by tailoring the dataset while keeping the same model architecture. Different types of dataset (molecular pairs) were extracted from ChEMBL based on MMPs, Tanimoto similarity and scaffold matching which result in six datasets: MMPs, Similarity (≥0.5), Similarity ([0.5, 0.7)), Similarity (≥0.7)), Scaffold and Scaffold generic. These datasets reflect different types of transformations, and the Transformer neural network was trained on each dataset. Our results showed that it is relatively easy to keep the structure constraints for MMP and Scaffold-based datasets compared to Tanimoto similarity-based datasets. Furthermore, the models trained on different types of molecular pairs transform a given starting molecule in a way that it reflects the nature of the dataset used for training the model, *e.g.* the model trained on MMPs modify the starting molecules by a single transformation, the models trained on similarity based molecular pairs allow for multiple modifications but keep the Tanimoto similarity in certain ranges, and the model trained on Scaffold-based molecular pairs allow for multiple modifications but keep the scaffold or generic scaffold constant. These models could complement each other and unlock the capability for the chemists to pursue different options for improving a starting molecule, therefore accelerate the drug discovery process.

Availability of data and materials

All source code and datasets used to produce the reported results can be found at https://github.com/MolecularAI/deep-molecular-optimization/tree/general_transformation and https://doi.org/10.5281/zenodo.5707626.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Jiazhen He performed the research. Christian Tyrchan, Werngard Czechtizky and Ola Engkvist proposed and supervised the project. All authors provided helpful feedback on the datasets used, experiment and results on the project. Jiazhen He wrote the manuscript, and all authors read and approved the final manuscript.

Acknowledgements

Jiazhen He thanks the Molecular AI group at AstraZeneca. Jiazhen He also thanks the postdoc program at AstraZeneca.

Author details

[1]Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden. [2]Medicinal Chemistry, Research and Early Development, Respiratory and Immunology (R&I), BioPharmaceuticals R&D, AstraZeneca, Gothenburg, Sweden. [3]Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden.

References

1. Polishchuk, P.G., Madzhidov, T.I., Varnek, A.: Estimation of the size of drug-like chemical space based on gdb-17 data. Journal of computer-aided molecular design **27**(8), 675–679 (2013)
2. Topliss, J.G.: Utilization of operational schemes for analog synthesis in drug design. Journal of medicinal chemistry **15**(10), 1006–1011 (1972)
3. Segler, M.H., Kogej, T., Tyrchan, C., Waller, M.P.: Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS central science **4**(1), 120–131 (2018)
4. Gupta, A., Müller, A.T., Huisman, B.J., Fuchs, J.A., Schneider, P., Schneider, G.: Generative recurrent networks for de novo drug design. Molecular informatics **37**(1-2), 1700111 (2018)
5. Bjerrum, E.J., Threlfall, R.: Molecular generation with recurrent neural networks (rnns). arXiv preprint arXiv:1705.04612 (2017)
6. Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., Aspuru-Guzik, A.: Automatic chemical design using a data-driven continuous representation of molecules. ACS central science **4**(2), 268–276 (2018)
7. Dai, H., Tian, Y., Dai, B., Skiena, S., Song, L.: Syntax-directed variational autoencoder for molecule generation. In: Proceedings of the International Conference on Learning Representations (2018)
8. Lim, J., Ryu, S., Kim, J.W., Kim, W.Y.: Molecular generative model based on conditional variational autoencoder for de novo molecular design. Journal of cheminformatics **10**(1), 1–9 (2018)
9. Jin, W., Barzilay, R., Jaakkola, T.: Junction tree variational autoencoder for molecular graph generation. In: International Conference on Machine Learning, pp. 2323–2332 (2018)
10. Liu, Q., Allamanis, M., Brockschmidt, M., Gaunt, A.: Constrained graph variational autoencoders for molecule design. In: Advances in Neural Information Processing Systems, pp. 7795–7804 (2018)
11. Simonovsky, M., Komodakis, N.: Graphvae: Towards generation of small graphs using variational autoencoders. In: International Conference on Artificial Neural Networks, pp. 412–422 (2018). Springer
12. Guimaraes, G.L., Sanchez-Lengeling, B., Outeiral, C., Farias, P.L.C., Aspuru-Guzik, A.: Objective-reinforced generative adversarial networks (organ) for sequence generation models. arXiv preprint arXiv:1705.10843 (2017)
13. Putin, E., Asadulaev, A., Ivanenkov, Y., Aladinskiy, V., Sanchez-Lengeling, B., Aspuru-Guzik, A., Zhavoronkov, A.: Reinforced adversarial neural computer for de novo molecular design. Journal of chemical information and modeling **58**(6), 1194–1204 (2018)
14. Putin, E., Asadulaev, A., Vanhaelen, Q., Ivanenkov, Y., Aladinskaya, A.V., Aliper, A., Zhavoronkov, A.: Adversarial threshold neural computer for molecular de novo design. Molecular pharmaceutics **15**(10), 4386–4397 (2018)
15. De Cao, N., Kipf, T.: Molgan: An implicit generative model for small molecular graphs. arXiv preprint arXiv:1805.11973 (2018)
16. Olivecrona, M., Blaschke, T., Engkvist, O., Chen, H.: Molecular de-novo design through deep reinforcement learning. Journal of cheminformatics **9**(1), 48 (2017)
17. Jin, W., Yang, K., Barzilay, R., Jaakkola, T.: Learning multimodal graph-to-graph translation for molecular optimization. arXiv preprint arXiv:1812.01070 (2018)
18. Kadurin, A., Nikolenko, S., Khrabrov, K., Aliper, A., Zhavoronkov, A.: drugan: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. Molecular pharmaceutics **14**(9), 3098–3104 (2017)
19. Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J., Chen, H.: Application of generative autoencoder in de novo molecular design. Molecular informatics **37**(1-2), 1700123 (2018)
20. Winter, R., Montanari, F., Steffen, A., Briem, H., Noé, F., Clevert, D.-A.: Efficient multi-objective molecular optimization in a continuous latent space. Chemical science **10**(34), 8016–8024 (2019)
21. Li, Y., Zhang, L., Liu, Z.: Multi-objective de novo drug design with conditional graph generative model. Journal of cheminformatics **10**(1), 33 (2018)
22. Kotsias, P.-C., Arús-Pous, J., Chen, H., Engkvist, O., Tyrchan, C., Bjerrum, E.J.: Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. Nature Machine Intelligence **2**(5), 254–265 (2020)
23. Jin, W., Barzilay, R., Jaakkola, T.: Hierarchical graph-to-graph translation for molecules. arXiv, 1907 (2019)
24. Jin, W., Barzilay, R., Jaakkola, T.: Hierarchical generation of molecular graphs using structural motifs. arXiv preprint arXiv:2002.03230 (2020)
25. He, J., You, H., Sandström, E., Nittinger, E., Bjerrum, E.J., Tyrchan, C., Czechtizky, W., Engkvist, O.: Molecular optimization by capturing chemist's intuition using deep neural networks. Journal of cheminformatics **13**(1), 1–17 (2021)
26. He, J., Mattsson, F., Forsberg, M., Bjerrum, E.J., Engkvist, O., Tyrchan, C., Czechtizky, W., et al.: Transformer neural network for structure constrained molecular optimization (2021)
27. Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of chemical information and computer sciences **28**(1), 31–36 (1988)
28. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
29. 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, pp. 5998–6008 (2017)

30. Kenny, P.W., Sadowski, J.: Structure modification in chemical databases. Chemoinformatics in drug discovery **23**, 271–285 (2005)
31. Tyrchan, C., Evertsson, E.: Matched molecular pair analysis in short: algorithms, applications and limitations. Computational and structural biotechnology journal **15**, 86–90 (2017)
32. Bemis, G.W., Murcko, M.A.: The properties of known drugs. 1. molecular frameworks. Journal of medicinal chemistry **39**(15), 2887–2893 (1996)
33. Mendez, D., Gaulton, A., Bento, A.P., Chambers, J., De Veij, M., Félix, E., Magariños, M.P., Mosquera, J.F., Mutowo, P., Nowotka, M., *et al.*: Chembl: towards direct deposition of bioassay data. Nucleic acids research **47**(D1), 930–940 (2019)
34. Cumming, J.G., Davis, A.M., Muresan, S., Haeberlein, M., Chen, H.: Chemical predictive modelling to improve compound quality. Nature reviews Drug discovery **12**(12), 948–962 (2013)
35. Schuffenhauer, A., Schneider, N., Hintermann, S., Auld, D., Blank, J., Cotesta, S., Engeloch, C., Fechner, N., Gaul, C., Giovannoni, J., *et al.*: Evolution of novartis' small molecule screening deck design. Journal of Medicinal Chemistry **63**(23), 14425–14447 (2020)
36. Dalke, A., Hert, J., Kramer, C.: mmpdb: An open-source matched molecular pair platform for large multiproperty data sets. Journal of chemical information and modeling **58**(5), 902–910 (2018)
37. Gogishvili, D., Nittinger, E., Margreitter, C., Tyrchan, C.: Nonadditivity in public and inhouse data: implications for drug design. Journal of cheminformatics **13**(1), 1–18 (2021)
38. Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., *et al.*: Analyzing learned molecular representations for property prediction. Journal of chemical information and modeling **59**(8), 3370–3388 (2019)

Abbreviations

MMPs: matched molecular pairs ADMET: absorption, distribution, metabolism, elimination and toxicity RNNs: recurrent neural networks VAEs: variational autoencoders GANs: generative adversarial networks SMILES: Simplified Molecular-Input Line-Entry System NLP: natural language processing Seq2Seq: sequence to sequence HierG2G: hierachical graph encoder-decoder RMSE: root-mean-square error NRMSE: normalized RMSE

Additional Files

Additional file 1 — Supplementary tables and figures