HIGH-CONTENT SIMILARITY-BASED VIRTUAL SCREENING USING A DISTANCE-AWARE TRANSFORMER MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular similarity search is an often-used method in drug discovery, especially in virtual screening studies. While simple one- or two-dimensional similarity metrics can be applied to search databases containing billions of molecules in a reasonable amount of time, this is not the case for complex three dimensional methods. In this work, we trained a transformer model to autoencode tokenized SMILES strings using a custom loss function developed to conserve similarities in latent space. This allows the direct sampling of molecules in the generated latent space based on their Euclidian distance. Reducing the similarity between molecules to their Euclidian distance in latent space allows the model to perform independent of the similarity metric it was trained on, thus enabling high-content screening with time-consuming 3D similarity metrics. We show that the presence of a specific loss function for similarity conservation greatly improved the model's ability to predict highly similar molecules. When applying the model to a database containing 1.5 billion molecules, our model managed to reduce the relevant search space by 5 orders of magnitude. We also show that our model was able to generalize adequately when trained on a relatively small dataset of representative structures. The herein presented method thereby provides new means of substantially reducing the relevant search space in virtual screening approaches, thus highly increasing their throughput. Additionally, the distance awareness of the model causes the efficiency of this method to be independent of the underlying similarity metric.

1 INTRODUCTION

1.1 MOLECULAR SIMILARITY SEARCH

The mean financial burden of researching and developing a new drug has been estimated to exceed 1 billion US dollars (Wouters et al., 2020). Resource, cost, and time efficient methods of finding new drug molecules are therefore imperative for reducing the costs and duration of drug development. Using computer-based methods can help reach this goal.

A well known concept in drug development is that similar molecules exhibit similar properties and activity profiles (Kumar & Zhang, 2018; Muegge & Mukherjee, 2016). This can enable researchers to find novel hits by comparing them with known active substances, which is the main principle behind similarity search in drug development. Similarities between compounds can be determined by different strategies, from simple descriptor-based comparisons over 2D fingerprints to detailed 3D measures such as shape-based or field-based similarities dependent on alignment of the molecules to be compared. To calculate similarities between molecules for large-scale similarity search, typically molecular fingerprints are utilized and computed. These fingerprints encode chemical properties and usually consist of binary vectors. While traditional molecular fingerprints were mainly rule-based (e.g. based on the presence of substructures or atom-pairs (Awale & Reymond, 2014; Capecchi et al., 2020)), data driven fingerprints (e.g. learned by machine learning models) became more prominent in recent years (Zagidullin et al., 2021). Various metrics like the Tanimoto or Dice coefficient, or the Tversky index can be used to compute similarities based on these binary fingerprints (Muegge & Mukherjee, 2016).

There is a large variety of molecular fingerprints, ranging from simple fragment-based 2D methods to complex 3D approaches (Axen et al., 2017; Kumar & Zhang, 2018). 2D based fingerprints can easily be applied to virtual screenings of multi-million compound databases (up to several billion) (Fischer et al., 2020; Cereto-Massagué et al., 2015). While this is possible in a relatively short period of time due to their low complexity, more complicated 3D similarity measures are realistically only feasible to use on smaller datasets of several hundred thousands up to a few million compounds (Fontaine et al., 2007; Chen et al., 2020).

Here, we present a different approach to the problem of high-content similarity screening combining transformer-based autoencoder models, similarity-based latent space shaping, and direct sampling in the reduced latent space representation. In this current proof-of-concept study presented here, we demonstrate the feasibility of the approach using 2D fingerprint similarities. We show that our approach can capture molecular similarities very well in latent space. The performance of the presented model is, however, independent of the used similarity metric. This allows researchers to train a model on highly complex 3D similarity metrics and thus perform high-content screening using metrics that otherwise would not be feasible to apply to a large set of compounds.

1.2 Related Work

One approach of learning chemical properties of molecules is by using so called autoencoders (Honda et al., 2019; Bjerrum & Sattarov, 2018; Hong et al., 2020; Yan et al., 2020). An autoencoder is a model that attempts to encode its input into latent space and decodes it again while minimizing the difference between the input and the decoded output. The latent space can be considered a reduced representation of the underlying structures of the chemicals in the dataset. Herein, we make use of an autoencoder in order to learn similarities of molecules. Honda et al. (2019) previously used a transformer model to generate molecular fingerprints from SMILES strings using a simple reconstruction loss function. Bjerrum & Sattarov (2018) found that mapping enumerated to canonical SMILES improves the conservation of similarities in latent space.

Conserving similarities in latent space is not only of high relevance in drug discovery but also in other fields such as image recognition. Schroff et al. (2015) proposed a loss function called triplet loss (Equation 1) which can be used to map related images to similar regions in latent space while increasing the distance between dissimilar images:

$$L(\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{N}) = max(||f(\boldsymbol{A}) - f(\boldsymbol{P})|| - ||f(\boldsymbol{A}) - f(\boldsymbol{N})|| + m, 0) \tag{1}$$

This loss function relies on the definition of an anchor ($\boldsymbol{A}$), a positive (i.e. similar) sample ($\boldsymbol{P}$), and a negative (i.e. dissimilar) sample ($\boldsymbol{N}$) and is therefore well suited for data with discrete labels. $f(\cdot)$ describes the coordinates of a compound in latent space, $||\cdot||$ the L2-norm, and m the hyperparameter specifying a margin to separate similar from non-similar molecules.

In this work, we follow the approach of Honda et al. (2019) and use a transformer model to autoencode SMILES strings to generate fingerprints suitable for similarity calculations. We then use the generated latent space encodings for similarity search based on Euclidian distances. In order to improve the similarity conservation in latent space, we compare a model based only on a reconstruction loss with models trained on additional loss terms to specifically learn similarities. Since the triplet loss function in Equation 1 requires discrete labels, working with similarities requires the definition of a similarity threshold separating similar molecules from dissimilar ones. Since such a separation is highly ambiguous for diverse sets of molecules, we developed a novel loss function which we call the similarity loss function. The similarity loss function can be used to work with continuous data, rendering it well-suited for working with similarities.

The herein presented models are therefore intended to estimate similarities based on Euclidian distances in latent space, allowing the subsequent use of exhaustive similarity search on a drastically reduced search space. We also show that a model trained on a small dataset is able to generalize to huge compound libraries containing highly diverse structures.

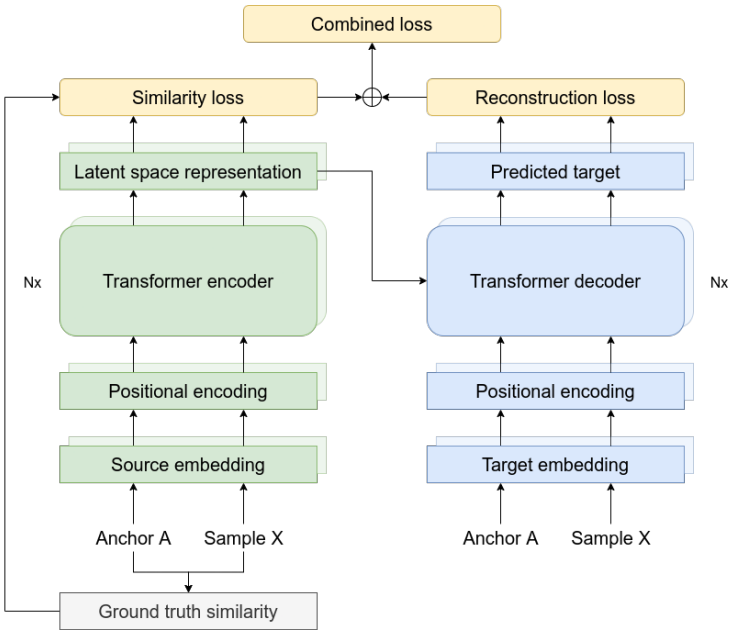

Figure 1: Architecture of the used transformer model. Encoder and decoder layers are constructed following the original publication of the transformer model by Vaswani et al. (2017). To help conserve similarities in latent space, a special loss function denoted as "similarity loss" is added to the reconstruction loss.

2 METHODS

2.1 MODEL ARCHITECTURE

In recent years, transformer-based models witnessed great success in various areas such as natural language processing, speech recognition, object detection, and more (Misra et al., 2021; Shi et al., 2020; Farahani et al., 2020; Hannan et al., 2021; Devlin et al., 2018). In this work, we follow the initial transformer model architecture proposed by Vaswani et al. (2017). Figure 1 shows a representation of the implemented model architecture. To encode simple SMILES representations of molecules, we first tokenized the strings, embedded them and added a positional encoding. An example of a tokenized SMILES string can be found in Figure 8. The positional encoding is done using a set of sine and cosine functions of varying frequencies as indicated in Equation 2 where pos refers to the position of the token in the sequence, d is the size of the embedding, and i is the dimension of the embedding. In this study, we set $d = 256$.

$$\begin{aligned} PE(pos, 2i) &= sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \\ PE(pos, 2i+1) &= cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \end{aligned} \tag{2}$$

The pre-processed data are then passed to a transformer encoder consisting of four layers. Each layer contains a multi-head attention layer. In this model, we used four heads per attention layer. To compute the attention, we follow the original article where attention is defined as shown in Equation 3 where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are matrices containing the queries, keys, and values, respectively, and d_k is the dimensionality of the keys (Vaswani et al., 2017).

$$attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{3}$$

This encoder computes a latent space representation of the input. To obtain a single vector representation for each source molecule, we average over all tokens in the sequence. For the decoder

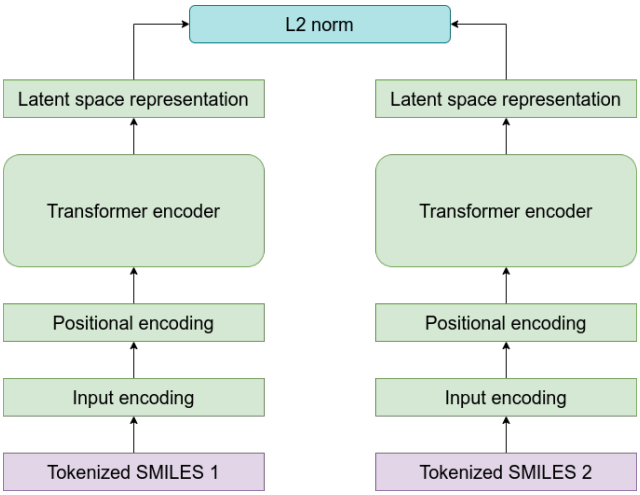

Figure 2: Predicting similarities between two molecules. The L2 norm is used to calculate the distance in latent space based on tokenized SMILES strings.

part, we feed the tokenized target SMILES to an embedding layer and add a positional encoding the same way it was done for the encoder part. Note that since we are working with an autoencoder, the source and target represent the same SMILES string while the target is right shifted. The transformer decoder layers combine the predicted latent space representation of the source with the attention weights and masked target embeddings, and subsequently predict the target sequence.

In a reqular transformer model, this prediction is then used to calculate the reconstruction loss usually in form of a cross entropy loss which is used to train the model. Here, we develop and test novel loss functions to conserve similarities in the produced latent space. When applying the model to predict similarities, the decoder part of the model will not be used. Similarities are calculated based solely on the latent space representation of the query molecules; the L2 norm is used to calculate the distance between two molecules in latent space (Figure 2). In praxis, a perfect correlation between latent space distance and ground truth similarity metric cannot be expected. Therefore, the purpose of this model is to obtain high enrichment in predicted, similar compounds to reduce the relevant search space by a significant degree. This will drastically increase the efficiency of virtual screening.

2.2 Similarity Conservation in Latent Space

When using a transformer model to auto-encode SMILES strings, the used loss function commonly only consists of a reconstruction term, e.g. in form of a cross entropy loss. While this may be sufficient to conserve similarities in latent space for small datasets, the model does not specifically learn relationships between molecules. The triplet loss function introduced in the previous section can be used to separate labelled samples in latent space. Since the herein presented work uses continuous data, a similarity threshold has to be defined with the intention of distinguishing between similar and dissimilar compounds. The determination of such a threshold is ambiguous and may differ between systems and their active molecules.

To better deal with the continuous nature of our data, we developed a novel loss function which we call the similarity loss (Equation 4).

$$L(\boldsymbol{A}, \boldsymbol{X}) = \left| a \cdot \|(1 - sim(\boldsymbol{A}, \boldsymbol{X})\| - \|f(\boldsymbol{A}) - f(\boldsymbol{X})\| \right| \tag{4}$$

The similarity loss depends on an anchor ($\boldsymbol{A}$) sample much like in the triplet loss function. However, it does not have to rely on the determination of positive and negative (i.e. similar and dissimilar) samples. Instead, it compares each anchor in a batch with all other samples ($\boldsymbol{X}$) in the same batch. Since most similarity metrics $sim(\cdot, \cdot)$ range from 0 to 1 (0 being completely different and 1 being identical), $1 - sim(\cdot, \cdot)$ can be used to convert the similarity to a relative distance. The loss function is therefore trying to set the Euclidian distance in latent space equal to the relative distance in data space. In order to spread the embedded samples in latent space, we included a scaling factor a to the

term describing the relative distance in data space. The complete loss function consists of the sum of reconstruction loss (here we use a cross entropy loss) and our similarity loss:

$$L(\boldsymbol{A}, \boldsymbol{X}) = \left| a \cdot \|(1 - sim(\boldsymbol{A}, \boldsymbol{X})\| - \|f(\boldsymbol{A}) - f(\boldsymbol{X})\| \right| - \sum_{I \in \{\boldsymbol{A}, \boldsymbol{X}\}} \sum_{i=1}^{n_I} \sum_{c} t_{i,c} \cdot log(\hat{p}_{i,c}) \quad (5)$$

where $t_{i,c}$ is the label of a token i, $\hat{p}_{i,c}$ is the predicted probability for class c for token i, and n_I is the number of tokens for compound I.

In the following subsections, we compare the performance of the presented loss functions in order to determine their suitability to conserve similarities in latent space.

3 Results and Discussion

3.1 Initial Tests Using a Small Dataset

For a comparison of the three loss functions, the model was trained on a small dataset containing 10,000 compounds (see section A.1.1 for details). The three models were trained using the reconstruction loss of SMILES strings (vanilla transformer), reconstruction plus triplet loss function, and reconstruction plus our newly developed similarity loss function. To compare the performance of the three models, we predicted the distances between a set of 100 randomly chosen reference compounds and all other compounds in the dataset and compared them to the respective ground truth similarities. Based on these calculations, we computed the area under the receiver operating characteristics curve (AUROC) using different similarity thresholds to distinguish similar from dissimilar compounds. To avoid bias from the high number of dissimilar compounds leading to increased AUROC values, we only included compounds with a mimimum similarity of 0.40 to the individual reference compounds in this analysis. As shown in Table 1, the model trained with our similarity

Table 1: AUROC values for the different models trained on a small dataset of 10,000 compounds. While the vanilla transformer model was trained using only a reconstruction loss function, the other two models were trained with an additional loss term to specifically enforce the conservation of ground truth similarities in the latent space.

Similarity threshold	Vanilla transformer	Triplet loss	Similarity loss
0.45	0.68 ± 0.17	0.73 ± 0.17	0.82 ± 0.18
0.50	0.69 ± 0.18	0.75 ± 0.16	0.86 ± 0.17
0.55	0.75 ± 0.18	0.80 ± 0.15	0.92 ± 0.08
0.60	0.76 ± 0.18	0.81 ± 0.15	0.91 ± 0.11
0.65	0.80 ± 0.17	0.85 ± 0.13	0.94 ± 0.09
0.70	0.84 ± 0.18	0.89 ± 0.12	0.96 ± 0.07
0.75	0.87 ± 0.16	0.91 ± 0.12	0.97 ± 0.07
0.80	0.90 ± 0.14	0.94 ± 0.09	0.98 ± 0.07
0.85	0.92 ± 0.14	0.96 ± 0.08	0.98 ± 0.07
0.90	0.94 ± 0.14	0.98 ± 0.05	0.98 ± 0.08
0.95	0.97 ± 0.09	0.99 ± 0.04	1.00 ± 0.01

loss function in addition to the reconstruction loss clearly outperformed the other two models. The AUROC values were above 0.90 for all tested similarity thresholds except the lowest two. For all three methods, we observed an increase in AUROC values with increasing similarity threshold. This is likely due to a negative correlation between the true positive rate and the total number of positives in a dataset.

The vanilla model often failed to distinguish between similar and dissimilar compounds based on the Euclidian distances in latent space. The predicted distances are all very similar which likely caused a blurring in latent space, rendering it difficult to accurately distinguish between similar and dissimilar samples. While the model trained with an additional triplet loss was often able to map similar compounds closer to the reference than dissimilar compounds, it also generated a very dense latent space in which small errors can lead to incorrect predictions. By including our custom

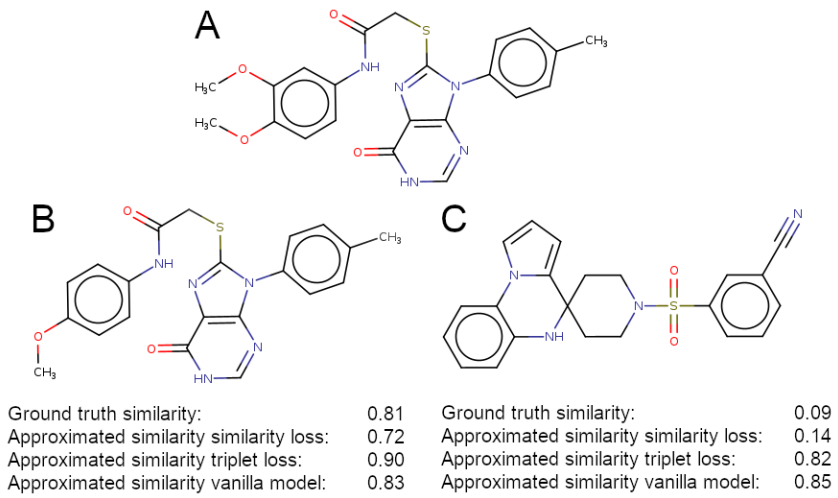

Figure 3: Similarity conservation in latent space. A) 2D structure of a randomly chosen reference compound. B) 2D structure of a molecule similar to the reference. Similarity was defined as having a Tanimoto coefficient above 0.8. The distances to the reference in latent space are shown for the individual models. C) 2D structure of a dissimilar molecule. Dissimilarity was defined as having a Tanimoto coefficient below 0.3. Latent space distances to the reference are shown for the individual models.

similarity loss, the model not only learned to correctly distinguish between similar and dissimilar molecules most of the times, it also spread out the generated latent space much more, making a separation between molecules much clearer.

Figure 3 highlights the differences between the three models on a randomly selected example. Compound B is highly similar to compound A, whereas compound C does not share a high similarity with A. Scaling the latent space distance d_{ij} between two molecules i and j to the range $[0, 1]$ and translating them into similarities s_{ij}^{LS}, allows for a comparison of ground truth and predicted similarities in latent space:

$$s_{ij}^{LS} \approx 1 - \frac{d_{ij}}{d_{max}}, \tag{6}$$

where d_{max} is maximum distance between any two molecules in latent space.

By applying this formula to the compounds in Figure 3, we obtain approximated similarities between A and B of 0.724, 0.899, and 0.825, and between A and C of 0.139, 0.821, and 0.852 using the similarity loss model, the triplet loss model, and the vanilla model, respectively. This shows that the similarity loss model is clearly better at discriminating between similar and dissimilar molecules.

Based on these results, we decided to focus on the model with the additional similarity loss function in the subsequent tests.

3.2 Scale-up Using the ZINC Database

Training of the model was subsequently upscaled using a large dataset of around 500,000 molecules (see section A.1.1 for details on the dataset generation). To test the optimized model, we chose a diverse set of 10 reference compounds and screened the whole downloadable ZINC database (around 1.5 billion SMILES) against each reference compound. The goal of this model was not to achieve a perfect correlation with calculated 2D similarities but to reduce the search space to a manageable size for subsequent exhaustive similarity search. We therefore checked for each reference compound how many of the 10 most similar database entries (determined using an exhaustive search) can be found within the N closest samples according to the model (Figure 4).

The model proved to be effective in reproducing the top 10 most similar compounds within the 15,000 closest samples in latent space for all investigated reference compounds. This corresponds

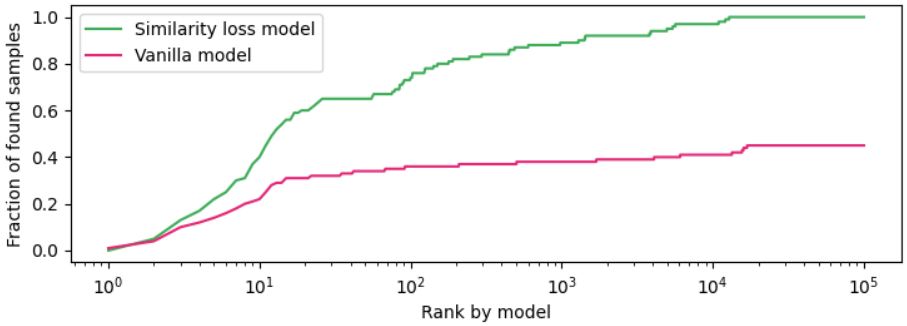

Figure 4: Comparison of reproduction abilities of the models with and without similarity loss function. The lines represent the normalized amount of the 10 most similar compounds within the top N closest samples in latent space for 10 reference compounds.

to a reduction of the search space by 5 orders of magnitude. In comparison, the vanilla model (i.e. without similarity loss function) only managed to identify 45% of all similar compounds within the top 100,000 predictions. To give further insights into the performance differences between the

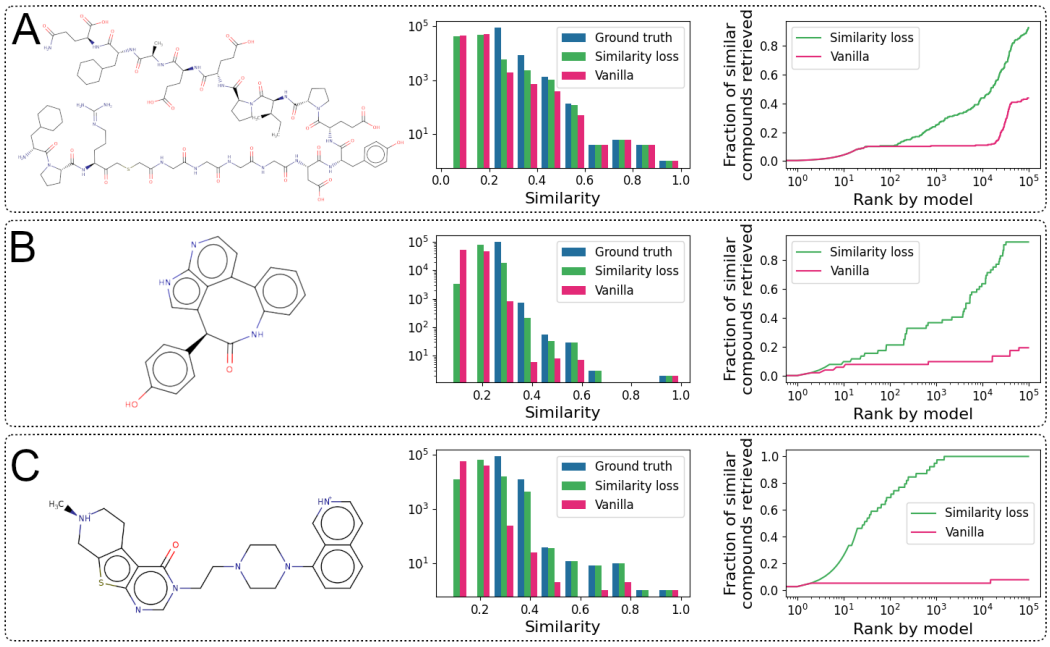

Figure 5: Similarity reproduction abilities. Left: 2D structure of the respective reference compound. Middle: Histogram of similarities (calculated using the exact method) of the 100,000 closest molecules to the reference in latent space ("ranking" task). Right: Reproduction of fairly similar compounds to the reference where a threshold of 0.5 was chosen to distinguish between similar and dissimilar compounds ("hit identification" task). A) analysis of the performance using a very large reference compound. B) performance with a smaller, cyclized reference compound. C) performance using a more linear compound with heterocycles.

vanilla model and the model trained with the similarity loss, we selected three structurally different compounds from the 10 reference molecules. The first reference (**reference1**) is a large peptide with a molecular weight of more than 2000 g/mol (PubChem CID 44335764). The second (**reference2**) is a highly cyclized compound (PubChem CID 44605611) and the third (**reference3**) is a potent 5HT1B receptor antagonist (PubChem CID 44405730).

The first "ranking" analysis (Figures 5, middle column) shows the models' potential to correctly identify and rank the 100,000 most similar compounds from the ZINC database. The right column in Figure 5 analyses the models' performance in identifying similar compounds to the reference (at a similarity threshold of 0.5). This analysis we name "hit identification" in the subsequent paragraphs. In general, the vanilla transformer was capable to identify similar compounds to large reference molecules such as **reference1**, but had significant difficulties for small substances, e.g. **reference3** (Figures 5).

In detail, the analysis showed that both the vanilla and similarity loss model performed very well for **reference1** (Figure 5A), with the similarity loss model being slightly better at reproducing the similarity distribution of the exact metric. In the "hit identification" task, with approximately the first 100 predictions, both models performed similarly. For the compounds ranked lower in predicted similarity to the reference, the similarity loss model started to clearly outperform the vanilla model. Within 100,000 top-ranked compounds, the similarity loss model was able to reproduce around 90% of the similar compounds whereas the vanilla model only managed to find around 40%.

For **reference2** (Figure 5B) and **reference3** (Figure 5C), the similarity loss model clearly outperformed the vanilla transformer model in both "ranking" and "hit identification" tasks. For **reference2**, the similarity loss model and vanilla model were able to identify 90% and 18% of the similar compounds, respectively. The largest difference was seen for **reference3**, where the similarity loss could identify all similar compounds within the top 2000 predictions while the vanilla model could only find around 7% of the similar compounds within the first 100,000 predictions. The relatively good performance of the vanilla model for **reference1** is likely due to the relatively low number of very large molecules in the data set, placing those molecules in a well-separated location in latent space. The model trained on the similarity loss however performed well in all three cases, proving the advantage of the additional loss term.

3.2.1 Exclusion of Scaling Factor in Loss Function

To study the importance of the scaling factor in the similarity loss function (Equation 4), we trained an additional model with a scaling factor of 1, thus disabling its effect. Using the same analyses as previously discussed revealed a drop in accuracy compared to using larger scaling factors, although it still performs better than the vanilla model (Figure 9). These findings have likely to do with the fact that a well structured latent space that is not too densely packed may be important for a good reproduction performance.

Finding a good value for the scaling factor is not trivial and this hyperparameter has to be tuned during training. In our tests, we found a value of 20 to work well for the initial analyses with a smaller dataset. However, when moving to a larger set, we found that decreasing the scaling factor to 10 further improves the performance of the model.

4 Conclusion

In this work, we developed models for similarity-based high-content screening with the aim to translate pairwise similarities in data space to Euclidian distances in latent space. This will facilitate efficient similarity searches independent of similarity metrics. We could show that the use of a loss function specifically designed to conserve molecular similarities in latent space greatly improved the accuracy of the model. By training a transformer autoencoder using a novel similarity loss function, it was possible to obtain a model that could be successfully used for similarity search against a database of more than 1 billion compounds. We demonstrated that our model was able to generalize from a comparatively small dataset, making it possible to learn highly complex similarity metrics that could otherwise not be applied to large datasets. While the presented model did not obtain a perfect correlation to the underlying ground truth similarity metric, it can be used to substantially reduce the available search space by five orders of magnitude. Such a drastic reduction of search space allows for subsequent use of exhaustive classical screening methods.

Here, we provide a proof of concept showing the possibility of generating a model for similarity search that is unaware of the underlying similarity metric, thereby uncoupling its efficiency from the chosen method.

REFERENCES

Mahendra Awale and Jean-Louis Reymond. Atom Pair 2D-Fingerprints Perceive 3D-Molecular Shape and Pharmacophores for Very Fast Virtual Screening of ZINC and GDB-17. *Journal of Chemical Information and Modeling*, 54(7):1892–1907, jul 2014. ISSN 1549-9596. doi: 10.1021/ci500232g. URL https://pubs.acs.org/doi/full/10.1021/ci500232g.

Seth D. Axen, Xi-Ping Huang, Elena L. Cáceres, Leo Gendelev, Bryan L. Roth, and Michael J. Keiser. A Simple Representation of Three-Dimensional Molecular Structure. *Journal of Medicinal Chemistry*, 60(17):7393–7409, sep 2017. ISSN 0022-2623. doi: 10.1021/acs.jmedchem.7b00696. URL https://pubs.acs.org/doi/full/10.1021/acs.jmedchem.7b00696.

Esben Bjerrum and Boris Sattarov. Improving Chemical Autoencoder Latent Space and Molecular De Novo Generation Diversity with Heteroencoders. *Biomolecules*, 8(4):131, oct 2018. ISSN 2218-273X. doi: 10.3390/biom8040131. URL https://www.mdpi.com/2218-273X/8/4/131.

Alice Capecchi, Daniel Probst, and Jean-Louis Reymond. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. *Journal of Cheminformatics*, 12(1):43, dec 2020. ISSN 1758-2946. doi: 10.1186/s13321-020-00445-4. URL https://jcheminf.biomedcentral.com/articles/10.1186/s13321-020-00445-4.

Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71(C):58–63, jan 2015. ISSN 10462023. doi: 10.1016/j.ymeth.2014.08.005. URL https://linkinghub.elsevier.com/retrieve/pii/S1046202314002631.

Ya Chen, Neann Mathai, and Johannes Kirchmair. Scope of 3D Shape-Based Approaches in Predicting the Macromolecular Targets of Structurally Complex Small Molecules Including Natural Products and Macrocyclic Ligands. *Journal of Chemical Information and Modeling*, 60(6):2858–2875, jun 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c00161. URL https://pubs.acs.org/doi/full/10.1021/acs.jcim.0c00161.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4171–4186, oct 2018. doi: 10.48550/arxiv.1810.04805. URL https://arxiv.org/abs/1810.04805v2.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. ParsBERT: Transformer-based Model for Persian Language Understanding. *Neural Processing Letters*, 53(6):3831–3847, may 2020. ISSN 1573773X. doi: 10.1007/s11063-021-10528-4. URL http://dx.doi.org/10.1007/s11063-021-10528-4.

André Fischer, Manuel Sellner, Santhosh Neranjan, Martin Smieško, and Markus A. Lill. Potential Inhibitors for Novel Coronavirus Protease Identified by Virtual Screening of 606 Million Compounds. *International Journal of Molecular Sciences*, 21(10):3626, may 2020. ISSN 1422-0067. doi: 10.3390/ijms21103626. URL https://www.mdpi.com/1422-0067/21/10/3626.

Fabien Fontaine, Evan Bolton, Yulia Borodina, and Stephen H. Bryant. Fast 3D shape screening of large chemical databases through alignment-recycling. *Chemistry Central Journal*, 1(1):12, dec 2007. ISSN 1752-153X. doi: 10.1186/1752-153X-1-12. URL https://bmcchem.biomedcentral.com/articles/10.1186/1752-153X-1-12.

Mohammad A. Hannan, Dickson N. T. How, M. S. Hossain Lipu, Muhamad Mansor, Pin Jern Ker, Zhao Y. Dong, Khairul S. M. Sahari, Sieh K. Tiong, Kashem. M. Muttaqi, T. M. Indra Mahlia, and Frede Blaabjerg. Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model. *Scientific Reports*, 11(1):19541, dec 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-98915-8. URL https://www.nature.com/articles/s41598-021-98915-8.

Shion Honda, Shoi Shi, and Hiroki R. Ueda. SMILES Transformer: Pre-trained Molecular Fingerprint for Low Data Drug Discovery. *ArXiv*, nov 2019. URL http://arxiv.org/abs/1911.04738.

Seung Hwan Hong, Seongok Ryu, Jaechang Lim, and Woo Youn Kim. Molecular Generative Model Based on an Adversarially Regularized Autoencoder. *Journal of Chemical Information and Modeling*, 60(1):29–36, jan 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.9b00694. URL https://pubs.acs.org/doi/10.1021/acs.jcim.9b00694.

Ashutosh Kumar and Kam Y. J. Zhang. Advances in the Development of Shape Similarity Methods and Their Application in Drug Discovery. *Frontiers in Chemistry*, 6(JUL):315, jul 2018. ISSN 2296-2646. doi: 10.3389/fchem.2018.00315. URL https://www.frontiersin.org/article/10.3389/fchem.2018.00315/full.

Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2886–2897, sep 2021. doi: 10.1109/ICCV48922.2021.00290. URL https://ieeexplore.ieee.org/document/9711345/.

Ingo Muegge and Prasenjit Mukherjee. An overview of molecular fingerprint similarity search in virtual screening. *Expert Opinion on Drug Discovery*, 11(2):137–148, feb 2016. ISSN 1746-0441. doi: 10.1517/17460441.2016.1117070. URL http://www.tandfonline.com/doi/full/10.1517/17460441.2016.1117070.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 07-12-June, pp. 815–823. IEEE, jun 2015. ISBN 978-1-4673-6964-0. doi: 10.1109/CVPR.2015.7298682. URL http://ieeexplore.ieee.org/document/7298682/.

Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer. Emformer: Efficient Memory Transformer Based Acoustic Model For Low Latency Streaming Speech Recognition. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021-June:6783–6787, oct 2020. ISSN 15206149. doi: 10.1109/ICASSP39728.2021.9414560. URL https://ieeexplore.ieee.org/document/9414560/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pp. 5999–6009. Neural information processing systems foundation, jun 2017. URL https://arxiv.org/abs/1706.03762v5.

Olivier J. Wouters, Martin McKee, and Jeroen Luyten. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA*, 323(9):844, mar 2020. ISSN 0098-7484. doi: 10.1001/jama.2020.1166. URL https://jamanetwork.com/journals/jama/fullarticle/2762311.

Chaochao Yan, Sheng Wang, Jinyu Yang, Tingyang Xu, and Junzhou Huang. Re-balancing Variational Autoencoder Loss for Molecule Sequence Generation. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, volume 20, pp. 1–7, New York, NY, USA, sep 2020. ACM. ISBN 9781450379649. doi: 10.1145/3388440.3412458. URL https://doi.org/10.1145/3388440.3412458.

Bulat Zagidullin, Ziyan Wang, Yuanfang Guan, Esa Pitkänen, and Jing Tang. Comparative analysis of molecular fingerprints in prediction of drug combination effects. *Briefings in Bioinformatics*, 22(6):1–15, nov 2021. ISSN 1467-5463. doi: 10.1093/bib/bbab291. URL https://academic.oup.com/bib/article/doi/10.1093/bib/bbab291/6353238.

A Appendix A: Method Details

The following section will describe the detailed neural network architecture, its hyperparameters, and the datasets used to train and test the model.

A.1 SMILES Transformer

Our model uses a transformer architecture as described in the publication by Vaswani et al. (2017). It was implemented in PyTorch using their integration of the Transformer module. The vocabulary was generated using tokenized SMILES strings that were used as input and encoded into 256 dimensional latent space. Our model consisted of 4 encoder and decoder layers with attention layers containing 4 heads. All models were trained using an Adam optimizer with a learning rate of 10^{-4} and 128 samples per batch. Since it was not possible to further increase the batch size due to memory limitations, we accumulated the gradients over 4 batches.

In order to determine the ground truth similarities, we calculated the Tanimoto coefficients based on 1024 bit Morgan fingerprints implemented in RDKit with a radius of 2. To conserve similarities in latent space, it is imperative that during training, each batch contains at least one similar compound to each sample (and for the triplet loss also at least one dissimilar compound). For the model trained on the similarity loss, we first randomly assigned compounds to a batch. To guarantee that similar compounds exist for each of those reference compounds, the algorithm randomly selected 3 of the 100 most similar compounds to the reference which were added to the batch. For the model with the triplet loss, we randomly selected 64 anchors per batch and for each chose a random compound with a Tanimoto similarity to the anchor of at least 0.6. It was assumed, that due to the intrinsic diversity of the dataset, for each anchor in a batch, there will always be a negative sample present. We defined negative samples as any compound with a similarity of less than 0.4 to the anchor.

The scaling factor a required by the similarity loss function (Equation 4) was set to 20.0 in the initial tests on a small dataset and was later decreased to 10.0 for the scaled up training. The margin m for the triplet loss function (Equation 1) was set to 1.0 for the comparison of the loss functions. These values were determined based on the retrospective analysis of the performance of each trained model.

A.1.1 Datasets

During an initial test phase, we used a randomly selected subset of 10,000 SMILES extracted from the natural compounds dataset obtained from the ZINC database. In the upscaling experiments, we randomly selected 0.03% of the compounds in each tranche downloaded from the ZINC database, leading to a dataset consisting of approx. 500,000 compounds. For testing the optimized model, the whole ZINC database was used which consisted of around 1,458,000,000 compounds at the time of testing.

For reproducibility, all used SMILES strings were converted to their canonical form using openbabel prior to training and testing.

A.2 Similarity Search

Once obtained, the distance aware SMILES embeddings were used to efficiently calculate distances (i.e. similarities) in embedding space. Facebook's faiss was utilized for this task using a FlatL2 index to calculate Euclidian distances in latent space. Faiss allows the construction and search of several types of indexes with various degrees of approximation.

B Appendix B: Investigation of Molecular Weights

To further investigate the reproduction abilities of the model with and without similarity loss, we analyzed the distribution of molecular weights of the 100,000 molecules predicted to be closest to the reference (Figure 6).

The data show that both models are well able to reproduce the molecular weight distribution of the 100,000 most similar compounds to **reference1** while the vanilla model slightly outperforms the

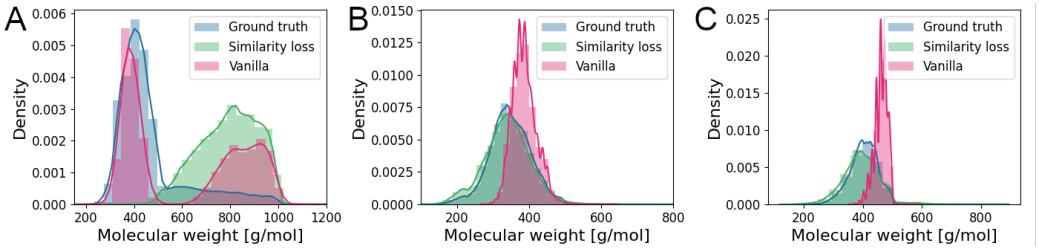

Figure 6: Reproduction of molecular weights. The histograms show the distribution of molecular weights of the 100,000 most similar compounds to **reference1** (A), **reference2** (B), and **reference3** (C) calculated using either the exact similarity metric, the model with similarity loss, or the vanilla transformer model.

model with similarity loss. This effect is the most pronounced at the lower end of the scale where the vanilla model is able to reproduce more of the low molecular weight compounds than the similarity loss model. More detailed analysis of this phenomenon revealed that these low molecular weight compounds are all highly dissimilar to the reference compound. When only including compounds with a similarity to **reference1** of 0.3 or more, these compounds disappeared and the similarity loss model showed a better overlap with the exact method (Figure 7). This sampling of very dissimilar molecules may be due to the fact that the vanilla transformer model generated a much denser latent space, leading to a generally lower distance between the very high molecular weight compounds and the molecules with lower molecular weight. While this benefits the vanilla model for **reference1**, it decreases its performance for **reference2** and **reference3** (Figure 6 B & C). In these examples, the model with similarity loss is generally better able to reproduce the distribution of molecular weights from the underlying (exact) similarity metric. Here, the vanilla transformer model is likely suffering because there are a lot of molecules in the screened data set that have a similar molecular weight to the two reference compounds. This causes the model to over sample these compounds in the densely packed latent space. In these cases, the sparser latent space generated by the similarity loss may prevent such an over sampling.

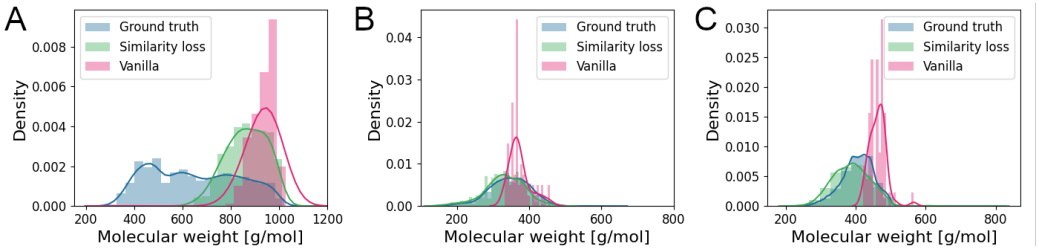

Figure 7: Reproduction of molecular weights. The histograms show the distribution of molecular weights of the 100,000 most similar compounds to **reference1** (A), **reference2** (B), and **reference3** (C) calculated using either the exact similarity metric, the model with similarity loss, or the vanilla transformer model. Only compounds with a similarity of at least 0.3 are considered.

C APPENDIX C: ADDITIONAL FIGURES

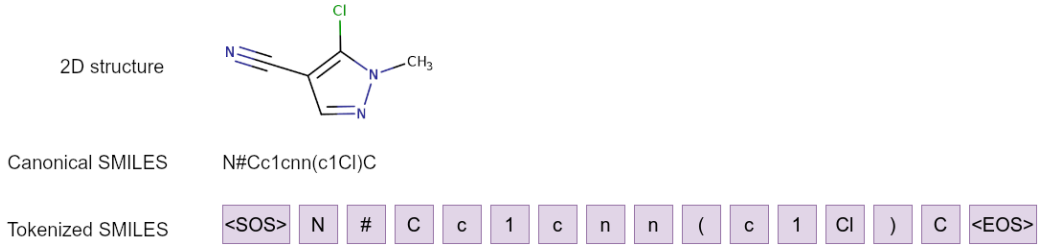

Figure 8: Example of SMILES tokenization. The 2D structure of a molecule, its SMILES representation, and the tokenized SMILES are shown. "<SOS>" and "<EOS>" represent labels specifying the start and the end of the sequence, respectively.

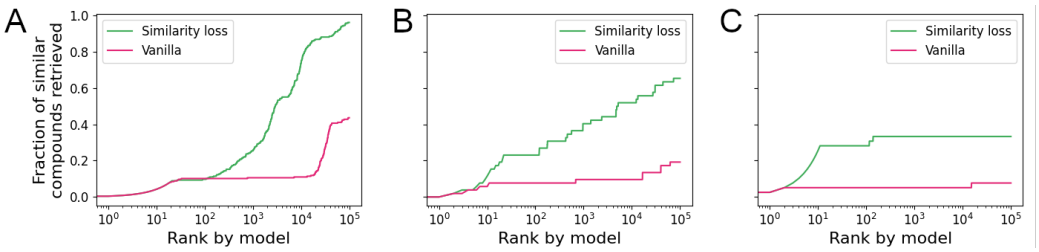

Figure 9: Performance of the model trained with the similarity loss scaling factor set to 1 for the "hit identification" task. The data for **reference1** (A), **reference2** (B), and **reference3** (C) are shown.