# Improving Chemical Autoencoder Latent Space and Molecular De-novo Generation Diversity with Heteroencoders (Automated Smiles V2)

* Chemical autoencoders are attractive models as they combine chemical space navigation with possibilities for de-novo molecule generation in areas of interest.
* By employing SMILES enumeration for either the encoder or decoder, it is found that the decoder has the largest influence on the properties of the latent space.
* Training a sequence-to-sequence heteroencoder based on recurrent neural networks (RNNs) with long short-term memory cells (LSTM) to predict different enumerated SMILES strings from the same canonical SMILES string gives the largest similarity between latent space distance and molecular similarity measured as circular fingerprints similarity.
* Using the output from the bottleneck in QSAR modelling of five molecular datasets shows that heteroencoder derived vectors markedly outperforms autoencoder derived vectors as well as models built using ECFP4 fingerprints, underlining the increased chemical relevance of the latent space.
* Introduction: - autoencoders have emerged as deep learning solutions to turn molecules into latent vector representations as well as decode and sample areas of the latent vector space.
* The encoder compresses and changes the input information into a bottleneck layer and a decoder decoder part recreates the original input from the compressed vector representation.
* In this research various encoder-decoder architectures have been proposed as well as the latent space has been regularized and manipulated using variational autoencoder and adversarial autoencoders. Both convolutional neural networks (CNNs), as well as recurrent neural networks (RNNs) have been used for the encoder part, whereas the decoder part has mostly been based on RNNs with either gated recurrent units (GRU) or long short-term memory cells (LSTM) to enable longer range sequence memory.
* The issue which autoencoder based on SMILES strings is that is the latent space a representation of the molecules or is it a condensed representation of the SMILES strings representing the molecules?
* A simple challenge with different SMILES representations of the same molecule suggest that the latent space is mote related to the SMILES string than to the molecule, which has also previously been noted.
* One way to solve this challenge could be to use special engineered networks and graph-based approaches for molecular generation.
* By translating from one format or representation of the molecule to the other, the encoder-decoder network is forced to identify the latent information behind both representations.
* It is also tested if these changes influence the properties of the decoder when used for *de-novo* design of molecules.
* An optimized and expanded heteroncoder architectures trained on ChEMBL23 datasets are used to extract latent vectors for subsequent use an input to QSAR models of five different molecular datasets.
* Methods: - Datasets **GDB-8** – it is dataset that is downloaded and split randomly into a train and test set using a 0.9 to 0.1 ratio.
* **ChEMBL23** – structures were extracted from the ChEMBL23 database. The maximum available length of the canonical SMILES string allowed for a molecule was 100 characters.
* For training 1.2 million molecules randomly selected and for testing 10 thousand molecules were selected.
* From the training set 400 thousand molecules are used for validation and 300 thousand molecules are used for training procedures.
* **QSAR datasets** - five experimental datasets were used, spanning physico-chemical properties as well as bioactivity.
* For the QSAR modeling, four datasets (IGC50, BDF, MP, LD50) were downloaded from the EPA Toxicity Estimation Software Tool webpage. Molecules for solubility dataset were obtained by resolving CAS numbers from the supporting info and the dataset was randomly split using the same ratio as the other QSAR datasets.
* **1D and 2D Vectorization** – vectorization was done similar to the vectorization used in Chemception networks with the following modifications: A PCA with three principal components was calculated on atomic properties from the Mendeleev python package.
* PCA and scaling were performed with the Scikit-Learn python package. RDKit was used to compute 2D coordinates and extract information about atom type and bond order.
* The normalized PCA scores of the atom types were used to encode the first three layers and the bond order was used to encode the fourth layer.
* The fifth layer was used to encode the RDKit aromaticity perception.
* **Neural Network Modeling for GDB-8 Dataset** – the final internal memory (c) and hidden layer (H) states were concatenated and used as input to a dense layer of 64 neurons with the rectified linear unit activation function (ReLU).
* The Decoder consisted of a single layer of 64 LSTM cells trained with teacher forcing in batch mode.
* A two-layer model was also constructed by increasing the number of LSTM cells to 128 and the number of LSTM layers to two in both the encoder and decoder.
* Four separate dense networks were used to decode the bottleneck layer into the initial C and H states for the two LSTM layers in the decoder.
* After training in batch mode, three models were created from the parts of the full model. A decoder model from the initial input to the output of the bottleneck layer. A model to calculate the initial states of the LSTM cells in the decoder, given the output of the bottleneck. Lastly, a stateful decoder model was constructed by creating a model with the exact same architecture as the decoder in the full model, except the LSTM cells were used in stateful mode and the input vector reduced to a size of one in the sequence dimension.
* A standard inception module was stacked with a reduction inception module three times, giving 7 inception modules in total including the initial one.
* The output was connected to the bottleneck consisting of a dense layer with the ReLU activation function.
* **Similarity metrics** – SMILES strings sequence similarities were calculated as the alignment score reported by the pairwise global alignment algorithm of the Biopython package.
* The match score was set to 1, the mismatch to -1, the gap opening to -0.5 and the gap extension to -0.05. The fingerprint similarity metric was calculated on the basis of circular Morgan fingerprints with a radius of 2.
* **Enumeration Challenge** – the encoder was used to calculate the latent space of the test set, the latent space coordinates of the non-canonical SMILES were calculated with the encoder and transformed and projected onto the visualization of the principal components from the PCA analysis.
* **Error analysis of output** – the percentage of invalid SMILES was quantified as the number of produced SMILES which could not be validated as molecules by RDKit. The number and nature of bonds was compared via a “bond sum formula” by counting the number of the single, double, triple and aromatic bonds.
* **Multinomial sampling of decoder** -**neural network modelling for the ChEMBL dataset** – the encoder consisted of two bidirectional layers of 128 CuDNNLSTM cells in each one-way layer.
* The final C and H states were concatenated and passed as input to a dense layer with 256 neurons using the ReLU activation function.
* The output from the dense layer were decoded by four parallel dense layers with the ReLU activation function, whose outputs were used to set the initial C and H states of the decoder LSTM layers.
* The decoder also consisted of two unidirectional layers of 256 CuDNNLSTM cells each.
* The network was trained using mini-batches of 256 one-hot encoded SMILES strings, using the Adam optimizer with an initial learning rate of 0.005.
* The training was monitored and controlled by three callbacks. One callback monitored the loss of the validation set and lowered the learning rate by a factor two when no improvement had been observed for 2 epochs.
* Another Callback stopped training when no improvement had been observed for 2 epochs.
* Another Callback stopped training when no improvement in the validation set loss had been observed for 5 epochs, and the last Callback saved the model if the validation loss has improved.
* **QSAR Modeling** – QSAR modelling was performed using the machine learning capabilities of the Open Science Data Repository. The hyper parameter search for a neural network was performed using Tree of Parzen Estimators algorithm as implemented in Hyperopt with the search space bounds.
* The performance on each dataset was optimized using 3-fold cross validation on the training set.
* The auto/heteroencoders trained on the ChEMBLE23 molecules were subsequently used to encode the QSAR datasets into vectors using the output from the bottleneck layer.
* The same hyper parameters were used as identified for the ECFP4 based models, with no further attempt to optimize the hyper parameters of the feed forward neural networks using the auto-/heteroencoder extracted inputs.
* **Results – GDB-8 dataset-based models**: - the models trained on enumerated SMILES output have a markedly larger final loss, but all models show a low degree of malformed SMILES when sampling the latent space vectors calculated from the test set.
* **Molecular and sequence similarity**: - using the same reference molecule, similarity metrics were calculated based on the latent space vectors of the test set, Morgan fingerprints and sequence alignment scores, followed by calculation of SMILES alignments coefficients (R2).



* Both the sequence alignment score and the fingerprint-based similarity have correlation with the latent space similarity, which shows that the latent space is at least somehow related to our traditional understanding of similarities between molecules.



* The models with a decoder trained on canonical SMILES show a markedly larger correlation between the latent space and the SMILES sequence similarity metric than between the fingerprint-based similarity and the latent space.
* The fingerprint and sequence similarities correlations to the latent space similarity are more on the same level when the decoder is trained using enumerated SMILES.
* The heteroencoder based on the image embedding of the molecule has the lowest correlations, indicating a markedly different or noisy latent space.
* **Error analysis**: - the use of enumeration in the input and output significantly increases the percentage of the outputs where the decoded molecule is not the same as the encode molecule.
* The bond types and atoms are in principle simple accounting operations independent of the SMILES enumeration, whereas the models struggle more with the scaffold reconstruction and the atom order, which are influenced by the SMILES enumeration.



* **Enumeration Challenge**: - the encoders capabilities to handle different SMILES from the same molecule were tested by projection on a PCA reduction of the latent space.
* Training the encoder with enumerated SMILES strings gives the tightest clustering, showing that the encoder has learned to recognize the same molecule independent on actual serialization of the SMILES string.
* By showing multiple different SMILES strings to the encoder during training, the encoder can produce the latent space coordinate most suitable for recreating the SMILES form of the decoder, irrespective of the SMILES form shown to the encoder.
* The model is doing a more complicated task which could work as regularization leading to better generalization.
* **Sampling using probability distribution**: - below figure show the difference between probability sampling of the can2can and can2enum model.



* Towards the end of the sampling, the decoder gets completely certain with the last 6 characters, probably because there is only one way to finish the molecule with the already sampled characters.
* The model trained on canonical SMILES in both encoder and decoder are very sure about the SMILES it wants to recreate, as only one SMILES form and one molecule is sampled.
* In the contrast, the decoders trained with the enumerated SMILES create different SMILES forms of the correct molecule, but also creates other molecules as well.
* **QSAR modelling using ChEMBL trained heteroencoders**: - training of the models on the ChEMBL datasets, resulted in final losses of approximately 0.001, 0.01, 0.10, 0.11 for the can2can, enum2can, can2enum, enum2enum configurations of the training sets, respectively Reconstruction performance of the different encoder/decoder configurations.
* The three different heteroencoders seem to produce latent vectors which perform very similar to each other in the QSAR modelling, with a tendency for the average performance to rise from enum2can to enum2enum over can2enum.
* The researchers observed that approximately 40% of the neurons of the bottleneck layer for each configuration are never activated.
* **Discussion**: - the representations used for training autoencoders have a marked influence on the properties and organization of the latent space.
* It is more reassuring that the dependence to the SMILES sequences is at a similar level to the fingerprint-based similarity, than the situation where the correlation to the SMILES sequence is much larger than the correlation to the fingerprint metric.
* The model could in principle memorized all graph structure instead of learning the rules behind the graph scaffolds, and then simply assign a specific sequence of atoms to the memorized graph.
* The model trained on enumerated data may have struggled because of low neural network fitting capacity.
* The 2-layer enum2enum model has much lower final loss and also much better reconstruction statistics when reconstructing and sampling the molecules than the single layer enum2enum model.
* With heteroencoders the molecule reconstruction rate becomes a more relevant term to measure than the SMILES validity rate, as the former can diverge a lot, while the SMILES validity error rate is still low.
* It is likely that even more complex architectures with three LSTM layers or a further enlargement of the number of LSTM cells would be needed to lower the molecule reconstruction error further.
* The image to sequence model produces a very low percentage of invalid SMILES and also has a low error rate with respect to molecule reconstruction, but is also decoding to canonical SMILES, which is an easier task then decoding to enumerated SMILES.
* The heteroencoder architecture may be useful for architecture experiments with large unlabeled datasets to find better architectures and suitable deep learning feature extractions for training on 2D embeddings of molecules.



* The failure of the image to SMILES heterencoder to produce significantly better latent representation fits with the observation that the latent space is mostly influenced by the decoding procedure, not the encoding procedure.
* It seems that using enumeration techniques or other formats for the decoder will influence that latent space the most.
* Training autoencoder on enumerated or different data further seems to improve the latent space with respect to its relevance for QSAR modelling.
* It seems that already the encoder independence of the SMILES form for the enum2can leads to a smoother latent space, which increases the relevance for QSAR modelling.
* Future benchmarking on common datasets will likely show the way to the best network architecture and what unlabeled datasets to use for specific tasks.



* The solubility dataset used have previously been carefully modelled with chosen features and topological descriptors, resulting in a R2 of 0.92 and a standard deviation of prediction of the test set of 0.6.
* The QSAR models based on heteroencoder derived latent vectors seem to almost match the performance of highly optimized QSAR models from selected features, and it may rather be the ECFP4 and can2can model derived latent vectors that are medicore for the tested type of QSAR tasks.
* For making sure that the improvements were not due to different optima of the model hyper parameters for the different data, the neural network architectures for the QSAR models were optimized based on the ECFP4 fingerprint input.
* The denser dimensionality could help protect against over fitting and make the choice of hyper parameters less critical for these models.
* The increased relevance of the latent space with respect to bioactivity and physico-chemical properties are likely to increase the relevance and qualify of the *de-novo* generated libraries where the neighborhood of as example lead compounds are sampled on purpose.
* The use of enumeration for training the decoder comes at the cost of greater uncertainty in the decoding, at a marginal improvement to the relevance of the latent space for QSAR modelling when compared to the enum2can model.
* Conclusion: - the study using a fully enumerated train and test set with 8 atoms showed that the latent space representation is sensitive to the chosen formats of the input and out in the training.
* The canonical SMILES for the decoder give a latent space representation which seem closer correlated to the SMILES strings than the molecular graphs.
* The latent space properties were mostly found to be influenced by the choice of training data and representations used as the decoder targets.
* The changed properties of the decoders have broken the dependence on producing canonical SMILES, and may make them more relevant in *de-novo* design approaches in drug discovery, where a balance between similarity and variance is the goal.
* The improved performance when using the latent space vectors from heteroencoders for QSAR modelling, further emphasizes their increased relevance, not just being a more SMILES independent representation of the molecule, but also for a better description of the chemical space relevant for biological as well as physico-chemical properties.