# SMILES Sandwich Transformer: De Novo Drug Design Using Molecular Fingerprints

**Noah Schweitzer, Tianwei Li , Mangalam Sahai , Chenyu Li, Shreyas Piplani** *
Carnegie Mellon University
Pittsburgh, PA 15213
{nschweit, tianwei3, msahai, chenyul2, spiplani}@cs.cmu.edu

## Abstract

Machine learning is showing promise in de novo drug development virtual screening. Molecular fingerprints are computed using deep learning methods, mapping molecules to a sparse discrete space, then using the multilayer perceptron classifiers and regressors to predict molecular properties pertaining to drug applications. However, these algorithms perform poorly for shallow prediction models or small datasets. SMILES Transformer attempts to improve upon this deficit by learning molecular fingerprints through unsupervised pre-training of the sequence-to-sequence language model using a huge corpus of SMILES, a text representation system for molecules. To improve upon this baseline model, we hypothesize that the specific rearrangement of the transformer sublayers, called sandwich transformers, could potentially enhance the encoding of the molecules. We trained and tested the same MLP for three unique classification tasks on 3 respective drug target datasets on the SMILES transformer versus six different models with unique rearrangements of the transformer sublayers: coefficients-1,2,3 for encoder and decoder layers, respectively.[2] The sandwich transformer decoder models performed better than encoder models in our experiments, the sandwich transformer decoder models made our results display a modest improvement in two out of three of the datasets, proving the effectiveness of the sandwich transformer.

## 1 Introduction

The discovery of novel molecules for pharmaceuticals and materials has the potential to significantly advance society and technology, perhaps treating uncommon ailments and paving the road for personalized precision medicine [17]. However, an exhaustive examination of the vast universe of possible chemicals is computationally intractable; the number of pharmacologically sensitive molecules has been estimated to be $10^{23}$ to $10^{80}$ [23]. The search for sensitive molecules is restricted by known structures and desired properties like solubility or toxicity.

The term "de novo drug design" refers to the process of creating unique chemical entities that satisfy a set of requirements via the use of computational growth methods [21]. Early de novo design algorithms used structure based approaches to grow ligands to sterically and electronically fit the binding pocket of the target of interest[4, 8]. The advent of Deep Learning and Machine Learning has accelerated the process of *De Novo* drug design,[1, 29, 13, 15, 22, 26] which is a common technique to create new chemical compounds from the parts of molecules while helping to figure out the right molecular structures that fit into the target binding cavity with good affinity. New chemotypes and compound alterations for lead structure optimization are helpful in new drug discovery designs aided by computer-aided molecular designs.[9]

*Mentor for this project

[2]Code can be accessed: https://github.com/witnik/dlFinalProject

Preprint. Under review.

Deep Learning plays a significant role in predicting molecular properties(dependent on structure) without *in vivo* studies. Essentially there are many models which can be used for generating new molecular: CharRNNs[28], Variational Autoencoder[18], Adversarial Autoencoder (AAE)[14], Junction Tree VAE (JTN-VAE)[12], Latent Vector-Based Generative Adversarial Network (LatentGAN)[25], N-gram Model[19, 23],Graph Convolutional Decoder [5]. Honda et.al proposed a Transformer model that outperformed RNNS, Graph Converlutional Decoder, and ECFP generation [10].Facebook AI researchers Press et.al improved Transformer models performance by reordering their sublayers and they call the new model Sandwich Transformer[24]. They experimented with Sandwich Transformer on machine translation tasks and showed better performance than vanilla Transformer, however, they did not now the math behind Sandwich Transformer at the time of writing the paper. Press et.al also hoped future researchers experiment with Sandwich Transformer on classification tasks.[24]

In this research, we propose a Sandwich Transformer on SMILES approach for extracting molecular fingerprints and subsequently classifying molecules with specified desired features using molecular fingerprints. We aim to use this approach to generate novel lead compounds with desirable pharmacological and physicochemical properties and improve the potential to reduce the time and cost of drug design. The technique tries to enhance the SMILES transformer model [10] trained on the ChEMBL database [7] to provide a more accurate latent space for molecule compounds. The approach is applied to a variety of molecular de novo design problems. An analysis was conducted to demonstrate how reordering sublayers in a transformer model influences the mechanistic behavior of the generative model.

## 2 Related Work

### 2.1 Molecular Representations

#### 2.1.1 String Representations

Representing molecular structures as the string has been adopted for generative models due to the abundance of sequence modeling tools such as recurrent neural networks, attention mechanisms, and dilated convolutions. Simplified Molecular Input Line Entry System(SMILES) is the most widely used string representation for generative machine learning models.[23] SMILES algorithm traverses a spanning tree of a molecular graph in depth-first order and stores atom and edge tokens. SMILES also uses tokens for branching and edges not covered with a spanning tree.

DeepSMILES was introduced as an extension to SMILES that seeks to reduce invalid sequences by altering syntax for branches and ring closures.SMILES syntax into a network architecture to increase the fraction of valid molecules. SELFIES define a new syntax based on Chomsky type-2 grammar augmented with self-referencing functions. International Chemical Identifier(InChI) is a more verbose string representation that explicitly specifies a chemical formula, atoms' charges, hydrogens, and isotopes.

One molecule can have multiple spanning trees and different SMILES strings from a molecule. Canonical SMILES give a single 'canonical' form for any particular molecule. The canonical SMILES are generated from each primary molfile in the ChEMBL and MoleculeNet datasets.

### 2.2 Models Available

#### 2.2.1 Neural Models

- **Character-Level Recurrent Neural Network(CharRNN):** We train the model by maximizing log-likelihood of the training data represented as SMILES strings.[28]
- **Variational Autoencoder(VAE):** It consists of 2 neural networks- an encoder and decoder- that infer a mapping onto the latent space. VAE parameters are optimized to encode and decode data by minimizing reconstruction loss and regularization term in a form of Kullback-Leibler divergence.[18]
- **Adversarial Autoencoder(AAE):** Replaces Kullback-Leibler divergence from VAE with an adversarial objective. An auxiliary discriminator network is trained to distinguish samples from a prior distribution and model's latent codes. The encoder then adapts to its latent codes to minimize discriminators predictive accuracy. The training process oscillates between training the encoder-decoder pair and discriminator.[14]

- **Junction Tree VAE(JTN-VAE):** This generates molecules in two phases by exploiting valid subgraphs as components. In first phase, it generates a tree-structured object(a junction tree) whose role is to represent the scaffold of subgraph components and their coarse relative arrangements. The components are valid chemical substructures automatically extracted from the training set. In the second phase, the subgraphs (nodes of a tree) are assembled together into a coherent molecular graph.[12]
- **Latent Vector Based Generative Adversarial Network(LatentGAN):** This combines an autoencoder and a generative adversarial network. LatentGAN pretrains an autoencoder to map SMILES structures onto latent vectors. A GAN is then then trained to produce latent vectors for the pre-trained decoder.[25]

#### 2.2.2 Non-neural Model

There are also several non-neural model aiming to solve this problem but we will not be elaborating on these here as they are not closely related to our paper. Examples including: N-gram generative model, Hidden Markov Models, and Combinatorial Generator.

### 2.3 Transformer-XL

It's an enhanced form of vanilla transformers. It can learn dependency that is 450% longer than vanilla transformers. It can achieve better performance on both shorter and longer molecules. It eliminates the error of context fragmentation and absolute positional encoding, by utilizing the concept of segment-level recurrence mechanism and relative positional encoding scheme.[6]

### 2.4 ECFP Fingerprints

As the outputs of the transformer are contextualized word-level representations , ST outputs a sequence of symbol-level(atom-level) representations (fingerprints). We concatenated the 4 vectors to get the fingerprints: mean and max pooled output of the last layer , the first output of the last and the penultimate layer. Now we have a 1024-dimensional fingerprint for each molecule from ST. This fingerprint is designed to have the same dimensionality with the baseline we use for , the extended-connectivity fingerprint(ECFP). [27]

## 3 Methodology

### 3.1 Model Implementation

#### 3.1.1 SMILES-Transformer

SMILES Transformer learns molecular fingerprints through unsupervised pretraining of the sequence-to-sequence language model using a huge corpus of SMILES[10]. It is trained using Cross-Entropy Loss, and simply is an encoder-decoder model which reads SMILES vocabulary. In addition, it has a latent-space extractor which extracts molecular fingerprints(using the same dimension as ECFP [27]), and it then passes through an MLP to predict molecular properties.(see figure1a).

We utilized the ChEMBL_24 Dataset- with 1820035 Chemical ids- to created a vocabulary file with 45 symbols in pickle format that is one-hot encoded and ready to be processed through the transformer model. Then we input the molecules from ChEMBL dataset to our transformer using one-hot encoding. To alleviate bias for the canonical representation of SMILES, we randomly transformed them every time they were used by the SMILES enumerator. [3] $nn.Embedding$ class in Pytorch was used to create a lookup table that maps the input sequence of symbol representations $X_1, X_2, ...X_n$ to a sequence of continuous representations $z = (z_1, z_2, ...z_n)$. Given this $z$, the decoder generates an output of sequence $y_1, y_2, ..., y_m$ of symbols one element at a time. Also, this transformer is auto-regressive, which is it inputs the generated output back into the decoder.

After embedding, we passed the dataset through a positional encoding function. The position representations from $vocab24.pkl$ were passed as Long tensors into the embedding layer, where we multiplied those weights by $sqrt(d_{model})$. The two embedding layers and pre-Softmax linear transformation shard the same weight matrix. Following equations were used to get the positional encoding of each SMILES input and add them to word embedding.

$$PE_{pos,2i} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{pos,2i+1} = cos(pos/10000^{2i/d_{model}})$$

This output of the encoding layer is then fed into the the multi-head attention layer. The multi-head attention layer essentially is the Concatenation of heads multiplied by weights. Each head is essentially a matrix multiplication of values and the softmax of queries and keys of each hidden state. $QK^T$ divided by$sqrt(d_k)$ to reduce the large values which could be obtained by the dot product.

$$Attention(Q, K, T) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

.

The multi-head attentions are as follows:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

.

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$.

Furthermore, the input is passed through a feedforward network, which is given by equation $FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$.

This output is normalized and broken down into two sub-layers of keys and queries, which act as input to the Multi-Head attention layer of the decoder. For the decoder part, the output of the previous time step is fed as input to the decoder, which is embedded and to which positional encoding is added. Furthermore, its fed into masked Multi-Head Attention, and its query is fed to the multi-head attention block.

Each given self attention block, we denote $Q_i = W_q h_i$,$K_i = W_i h_i$,$V_i = W_v h_i$.

The decoder inserts a third sub-layers, which performs multi-head attention over the output of the encoder stack. Similar to encoder we apply residual connections around each of the sub-layers, followed by layer normalization. Finally, we pass the output of the attention layers through a Linear Layer, which is further passed through a Softmax to get the required Probability distribution for the word.

$$O_i = \sum_{j=0}^{i-1} W_{ij} V_j$$

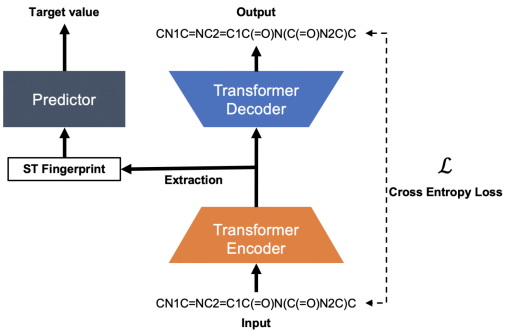
**Loss Function Used: Cross Entropy Loss**
Here cross function entropy loss is defined as the sum of the product of truth label(correct position of symbol(atom) in vocab list) and log(probability of that particular symbol).

$$L_{CE} = -\sum_{i=1}^{n} t_i log(p_i)$$

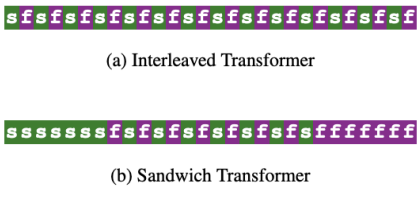
In order to perform classification tasks, we extracted features from the encoder layer of our model. The SMILES-Transformer generates a succession of symbol-level (atom-level) representations which could be pooled together to create molecule-level representations (fingerprints): the mean and maximum pooled output of the final and penultimate layers of the encoding layers, which will provide 1024-dimensional fingerprints for each molecule from SMILES- Transformer.

### 3.1.2 Reordering Sub-layers in Transformer Model

The vanilla transformer contains interleaving sublayers in both encoder and decoder. Each sublayer contains interleaved self-attention layers and feedforward layers( see figure1b ). Given the structure of the transformer, it is possible to rearrange the sublayers in transformer models, which could potentially produce better or worse results compared to the original transformer model. The sublayers are permuted randomly to compose new transformers and then trained on WikiText-103. The models of permuted sublayers use perplexity as the metric to compare with the vanilla transformer model while all hyperparameters are held constant [24]. While some of the models perform worse than the original transformer model, a significant number of them actually perform better(See figure-2b). Furthermore, by randomly generating 20 more unbalanced transformers (different numbers of self-attention layers and feedforward layers), it is observed that a balanced structure(the same number




(a) SMILES Transformer[10]



(a) Interleaved Transformer

(b) Sandwich Transformer

(b) Transformer structure: green symbol s represent self-attention layer and purple symbol f represents feedforward layer[24]

Figure 1: Architecture

Table 1: Average Attention Distance between each model pair on WikiText-103 validation dataset. Initialized with different random seeds.

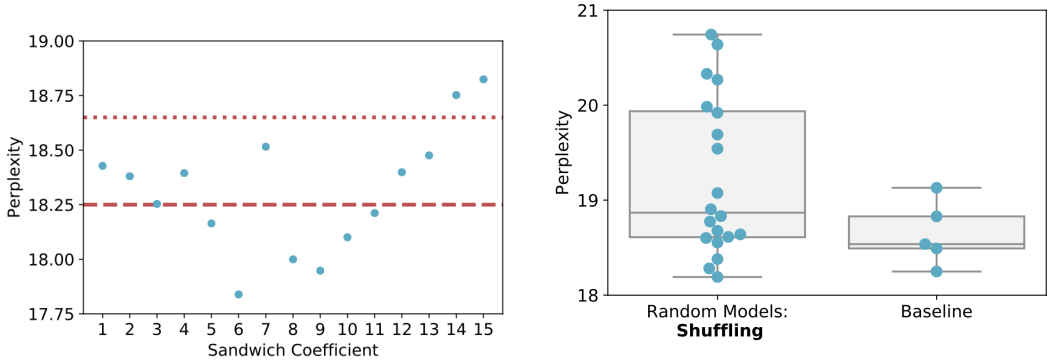
| Model Pair | Average Attention Distance |
|---|---|
| Baseline-Baseline | $1.081 \cdot 10^{-3}$ |
| Sandwich-Sandwich | $1.067 \cdot 10^{-3}$ |
| Baseline-Sandwich | $1.289 \cdot 10^{-3} \pm 0.049 \cdot 10^{-03}$ |

of self-attention layers and feedforward layers) tends to have better performance. Moreover, it is observed that the model has lower perplexity when more self-attention layers are in the bottom half of the model and more feedforward layers are in the top half of the model[24].

This analysis motivates a new design of the SMILES transformer using the concept of a sandwich transformer. The sandwich transformer contains self-attention layers, interleaved layers, and then feedforward layers (see figure 1b). The sandwich transformer model contains a coefficient $k$ which specifies the number of self-attention layers and feedforward layers placed at the bottom and top of the model. For example, bottom representation on figure-2a has coefficient value 7. The Sandwich Transformer is then trained on WikiText-103 and compared to the 16-layer vanilla transformer model. Figure 3 displays the results comparing sandwich transformers with different coefficients to the vanilla transformer model. The dotted line is the average baseline model's perplexity trained with different random seeds, and the dashed line represents the best baseline model [24]. As seen in the figure-2a, several of the sandwich transformer models outperform the vanilla transformer.

As further proof of the sandwich model's effectiveness, the model is then compared to a transformer-based translation model[31] consisting of encoder and decoder layers. The encoder and decoder layers are then separately substituted by the sandwich structure. For the decoder, as it contains self-attention, cross-attention, and feedforward layers, each self-attention layer and cross-attention layer are being considered as a single unit for reordering purposes[24] . The results ( see figure 2a )show where several proposed structures outperform the baseline model.

The authors did not have an explanation for why sublayer reordering improves performance on language modeling at the time this paper was presented at Proceedings of the $58^{th}$ Annual Meeting of the Association for Computational Linguistics. To better understand sublayer reordering's effect on the model, the authors train two baseline, vanilla transformer models and two sandwich models with different seeds on the WikiText-103 dataset. During inference, the attention values that each token's head assigns to all other tokens on the validation set are recorded. The average attention distance was then computed (see Table-1). Given a token and a self-attention sublayer, the Hungarian algorithm[16] was utilized to find a matching of heads in the first model to heads in the second model such that the earth mover's distance (EMD) is minimized. As seen in Table1, the average attention distance between models of the same architecture are significantly lower. The results therefore indicate that reordering the sublayers has a significant effect on the attention function that the model layer learns in each head. Thus, we hypothesize that incorporating the sandwich layering into our SMILES transformer will improve our metric scores.




(a) Transformer's sandwich coefficient (k) and validation perplexity. The dashed line represents the best baseline model.[24]

(b) The perplexities on the WikiText-103 development set of 20 randomly generated models with 16 self-attention and 16 feedforward sublayers and of the 5 standard transformers[24]

Figure 2: Sandwich Transformer Performance

### 3.2 Datasets

#### 3.2.1 ChEMBL

ChEMBL is a freely accessible database that contains information on the binding, functional, and ADMET properties of many drug-like bioactive chemicals.[7, 11]The data are manually extracted from the source published literature and then vetted and standardized to ensure their quality and usability across a broad spectrum of chemical biology and drug development research questions.[7, 20, 2] A large proportion of the data in ChEMBL is extracted manually from full-text articles published in seven Medicinal Chemistry journals: MedChemComm, Journal of Medicinal Chemistry, ACS Medicinal Chemistry Letters, European Journal of Medicinal Chemistry, Bioorganic & Medicinal Chemistry, Bioorganic & Medicinal Chemistry Letters, and Journal of Natural Products.[7]ChEMBL data were used to test the interaction of a drug with numerous potential proteins were mapped to various targets. Some data models have been established inside ChEMBL to distinguish between targets (the entities with which a drug interacts to produce its effects) and the molecular components (typically proteins) that comprise those targets.[2] CheMBL-30 data visualization see Figure: 3

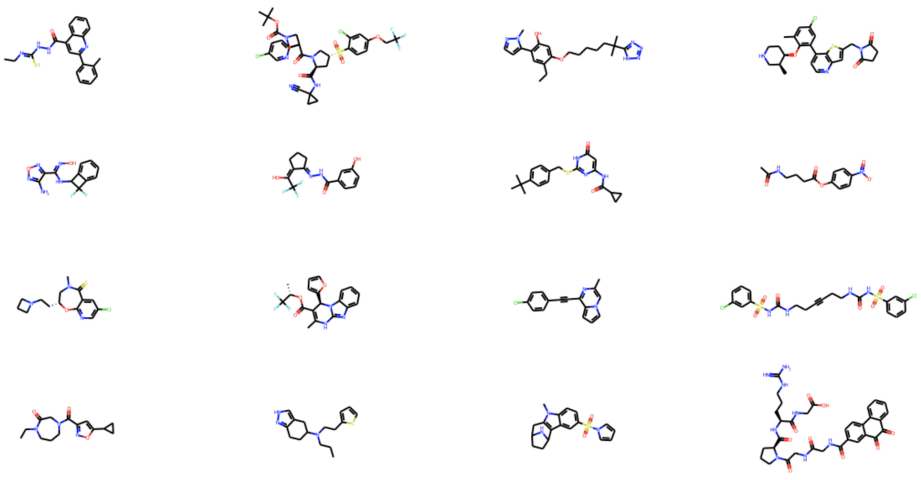


Figure 3: ChEMBL_30 Compound Visualization

#### 3.2.2 MoleculeNet

MoleculeNet is a large-scale benchmark for molecular machine learning. MoleculeNet curates multiple public datasets, establishes metrics for evaluation and offers high-quality open-source

implementations of various previously proposed molecular featurization and learning algorithms.[32] MoleculeNet is built upon multiple public databases. The entire collection currently includes over 700,000 compounds tested on various properties. Data are being collected in six major categories: quantum mechanics, physical chemistry, biophysics, and physiology.[32] Most of the input molecules in databases are represented using SMILES strings. 3D coordinates are also included in part of the collection as molecular features, enabling different methods to be applied. Properties, or output labels, are 0/1 for classification tasks or floating-point numbers for regression tasks.[32] For our tasks we used three datasets from MoleculeNet to evaluate our extracted fingerprints (see table 2)

Table 2: Summarized information of MoleculeNet [32, 10]. We chose three classification datasets for analyses

| Category | Dataset | Tasks | Mols | Metric | Description |
|---|---|---|---|---|---|
| Biophysics | HIV | 1 | 41913 | ROC-AUC | Ability to inhibit HIV replication |
| | BACE | 1 | 1522 | ROC-AUC | Binding results for inhibitors of human BACE-1 |
| Physiology | BBBP | 1 | 2053 | ROC-AUC | Blood-brain barrier penetration |

.

### 3.3 Baseline Selection

Character RNNs are the best available model for predicting molecular properties[32]. Since SMILES Transformer is the enhanced form of Character-RNNs, so we are using it as the baseline model we are using for our study. For SMILES representation it is the state-of-the-art model available for predicting molecular properties. The baseline model has been implemented from scratch so we are better able to compare our results with the baseline model.

### 3.4 Evaluation Metrics

We performed classification prediction on three tasks and then obtained the area under the receiver operation characteristics (ROC-AUC) which shows the performance of a classification model at all classification thresholds. In the manuscript that we compare our results to, the authors argue in favor of using the Data Efficiency Metric (DEM) as opposed to a single ROC-AUC. Their reasoning is that they use various datasets with a wide range of sizes for the different tasks.

Thus, an ROC-AUC should be calculated after training the MLP on a varying percentage of the data. We followed the manuscript by training on the percentage from $1.25\%$ to $80\%$, and doubling the percentage each time. The DEM can then be formulated as $M_{DE}(f, m) = \frac{1}{|I|}\sum_{i \in I} m(f_i, X_i, Y_i)$, where $(X_i, Y_i)$ denote the test data sampled from whole dataset $(X, Y)$ at the rate of $1 - i$.

### 3.5 Alternative Model Exploration

Initially we investigated in Transformer-XL and its ability to train on longer sequences which would theoretically improve the performance when being used on SMILES data. We tried to implement transformer-XL on SMILES vocabulary to improve prediction accuracy of SMILES-transformer. However, after more in-depth investigation, we found that Transformer-XL is a decoder only model which proscribes the possibility of encoding SMILES structures. Hence, we decided to consider the approach of improvising using Sandwich transformer.

## 4 Experiments Setup

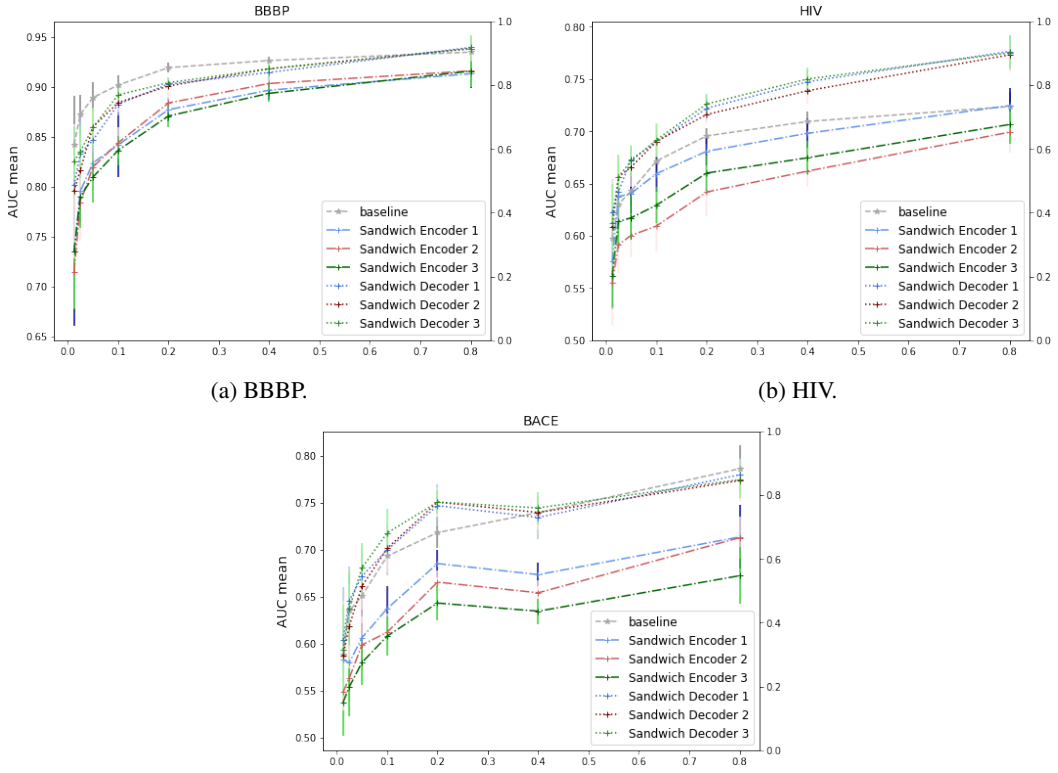
We experimented with sandwich coefficients-1,2,3 which are more probable to perform better than sandwich coefficients-4 because a pure sandwich structure without any original interleaving layers does not perform well in general [24]. Therefore we train our model with sandwich coefficients-1,2,3 and compare the results with our baseline SMILES transformer model. Separately, we changed the encoder layers or the decoder layers to sandwich structure with sandwich coefficient-1,2,3. For decoder layers in transformer, each layer contains three sub-layers: self-attention, cross-attention, and feeds-forward. When building sandwich decoders, we grouped cross-attention and self-attention together for the purpose of reordering according to the sandwich transformer paper. In total we are training 6 models aside from the baseline model.

We pre-trained our proposed models on on the SMILES data from ChEMBL24 dataset. For our baseline model, we fine tuned our hyperparameters to best replicate the paper's results. Then for our

Table 3: Encoder(Decoder) sandwich model keeps the decoder (encoder) unmodified. We train the baseline model (Transformer-large with hyperparameters of 5 times with different random seeds)

| **Sandwich Coefficient** | **Encoder Sandwich** | **Decoder Sandwich** |
|---|---|---|
| 0(Baseline) | $28.74 \pm 0.15$ | |
| 1 | 28.71 | 28.64 |
| 2 | 28.71 | 28.56 |
| 3 | 28.81 | 28.67 |
| 4 | 28.48 | 28.66 |
| 5 | 28.45 | 28.76 |

proposed models, we used the same hyperparameters so we can do an apple-to-apple comparison between the Sandwich transformer and SMILES transformer on the SMILES data. We pre-trained the proposed models for a total of 12 epochs to minimize the Cross Entropy Loss between the input SMILES data and the output probability with AdamW optimizer and Cosine Annealing learning rate scheduler in an unsupervised fashion. We pre-trained the model with learning rate of 1e-5 and a weight decay of 1e-5. To further prevent overfitting, we also increased the dropout rate of transformer to 0.4 and the dropout rated of positional encoding to 0.1. We tune several more hyperparameters to help our model perform well (Supplement Table 6). Then we use a simple multilayer perceptron classifier to evaluate the extracted fingerprints. First we use the pre-trained model to perform fingerprint extraction from the following three datasets: HIV, BBBP, and BACE. Then these extracted fingerprints are inputed into a simple MLP provided by Scikit-Learn with all the default hyperparameters for classification tasks and their DEM score is calculated and compared. To explain differences in performance on MLP classification, we will visualize their latent space using t-SNE[30] provided by Scikit-Learn.




(a) BBBP.

(b) HIV.

(c) BACE.

Figure 4: AUC Mean for Different Models


(a) Encoder-3 (b) Decoder-3 (c) Baseline

Figure 5: Sandwich Transformer Models Latent Space BBBP Molecules

Table 4: Comparison of Sandwich transformer models data efficiency metric (DEM) with the baseline models on 3 datasets from MoleculeNet [32, 10]. The up/down arrows show that the higher/lower score is better, respectively.

| Dataset | HIV ↑ | BBBP ↓ | BACE ↑ |
|---|---|---|---|
| Baseline | 0.667 | **0.898** | 0.687 |
| Sandwich Decoder 1 | 0.696 | 0.874 | 0.698 |
| Sandwich Decoder 2 | 0.693 | 0.873 | 0.69 |
| Sandwich Decoder 3 | **0.698** | 0.882 | **0.7** |
| Sandwich Encoder 1 | 0.659 | 0.84 | 0.64 |
| Sandwich Encoder 2 | 0.623 | 0.838 | 0.622 |
| Sandwich Encoder 3 | 0.638 | 0.836 | 0.604 |

## 5 Results and Discussion

The plots for AUC scores vs. the fraction of data used for training is displayed in Figures4b 4a 4c and Table4 shows the DEM scores for each model trained. When compared to the baseline model, our sandwich decoder models improved on 2/3 of the MLP classification tasks. The DEM scores for all three decoder sandwich models tested performed better than baseline for the BACE and HIV datasets. The baseline model had better performance than all six of our models on the BBBP dataset. When comparing between the sandwich transformer models, the three encoder models performed worse than baseline and the three decoder models for each dataset.

To inspect why some models lead to improved predictive performance over baseline, we visualized the latent space for the BBBP dataset as seen in Figure 5. The categorical target values are successfully separated for the decoder-3 and baseline model but not for the encoder-3 model. Furthermore, the latent space appears to have a simpler representation where the molecules are spread out over a wide space as opposed to being clustered together for the decoder models and the baseline compared to the encoder models. This trend of a simple latent space for decoder and baseline models vs. encoder models continues for the other two datasets (Supplemental Figures 6 7). We hypothesize that the encoder sandwich models lead to a more complicated latent space and subsequently worse classification performance because we are shuffling the encoder part of the architecture. Even though the loss is relatively the same after training for the encoder vs. decoder models (Supplemental Table 5), the shuffling causes a feature extraction that is not as coherent as the other models. Meanwhile, the decoder sandwich model still has an encoder architecture similar to baseline, and shuffling in the decoder architecture leads to more efficient training of the model than the baseline model and thus better classification performance.

One of the main reasons that the encoder-only model performed worse is the method in which we extract the feature representations. In the baseline model and decoder sandwich model, we concatenate the output mean and max pooled output of the last layer, the first output of the last and the penultimate layer. Because of the reordering of the sublayers, this method is not feasible for the encoder sandwich model. Instead, we concatenated the last two feedforward layers in the encoder architecture.

Another interesting observation is the differences in the HIV and BACE datasets versus the BBBP dataset. We hypothesize that BBBP had different classification results compared to the other datasets because the length of molecules found in BBBP dataset are quite large compared to the other two. The difference in lengths might lead to context fragmentation, thus giving us at-par results with the baseline model. In the SMILES transformer paper, the authors observed that MLP performance depended on the length of the SMILES text, further backing the idea of fragmentation causing performance variations. Future studies will have to look more closely on why some performance is worse than others.

## 6 Conclusion

Our results show that we were able to modestly improve the generative models' prediction accuracy of molecular properties in the context of SMILES transformer by reordering the sublayers of the transformer model. The three decoder sandwich models performed better than the baseline in 2/3 tasks. Even though the encoder sandwich layer had a similar loss value at the end of training, the reordering of its sublayers caused the model to have a more complicated feature representation and thus a worse performance on classification tasks as it could not separate the data as well. Meanwhile, the decoder sandwich transformer model improved over baseline because it was able to improve auto-encoding during training due to the reordering of the sublayers and still encode a coherent latent space. Because of time constraints, we were only able to evaluate on three separate datasets. Future work includes testing the plethora of datasets available on MoleculeNet in order to definitively conclude that the sandwich, decoder-only transformer model improves performance over SMILES transformer.

The proposed work is limited to SMILES vocabulary encoder-decoder, but we have other forms of representation as well like Molecular Graphs. A cohesive wholesome model which can read multiple representations and predict molecular properties. Graph Neural Networks, Weave Networks and Message Passing Networks can be good models. In the future, we can also try to use both sandwich encoder and decoder instead of only using either sandwich encoder or decoder. Moreover, we are now grouping self-attention and cross-attention sub-layers together when performing reordering for sandwich decoder, we can consider these two sub-layers separately which can lead to different sandwich structures. We will need to perform exhausted experiments for all possible combinations as they are not being experimented in the sandwich transformer paper.

## References


[1] V. Bagal, R. Aggarwal, P. K. Vinod, and U. D. Priyakumar. Molgpt: Molecular generation using a transformer-decoder model. *J Chem Inf Model*, 2021.

[2] A. Patrícia Bento, Anna Gaulton, Anne Hersey, Louisa J. Bellis, Jon Chambers, Mark Davies, Felix A. Krüger, Yvonne Light, Lora Mak, Shaun McGlinchey, Michal Nowotka, George Papadatos, Rita Santos, and John P. Overington. The chembl bioactivity database: an update. *Nucleic Acids Research*, 42(D1):D1083–D1090, 2013.

[3] Esben Jannik Bjerrum. SMILES enumeration as data augmentation for neural network modeling of molecules. *CoRR*, abs/1703.07076, 2017.

[4] Hans-Joachim Böhm. The computer program ludi: a new method for the de novo design of enzyme inhibitors. *Journal of computer-aided molecular design*, 6(1):61–78, 1992.

[5] Xavier Bresson and Thomas Laurent. A two-step graph convolutional decoder for molecule generation. *arXiv preprint arXiv:1906.03412*, 2019.

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[7] Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2011.

[8] Valerie J Gillet, William Newell, Paulina Mata, Glenn Myatt, Sandor Sike, Zsolt Zsoldos, and A Peter Johnson. Sprout: recent developments in the de novo design of molecules. *Journal of chemical information and computer sciences*, 34(1):207–217, 1994.

[9] Markus Hartenfeller and Gisbert Schneider. De novo drug design. *Chemoinformatics and computational chemical biology*, pages 299–323, 2010.

[10] Shion Honda, Shoi Shi, and Hiroki R Ueda. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738*, 2019.

[11] ChEMBL Website https://www.ebi.ac.uk/chembl/. Chembl database explore, 2022/03/02.

[12] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

[13] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Zhavoronkov. 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.

[14] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Zhavoronkov. 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.

[15] Panagiotis-Christos Kotsias, Josep Arús-Pous, Hongming Chen, Ola Engkvist, Christian Tyrchan, and Esben Jannik Bjerrum. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence*, 2(5):254–265, 2020.

[16] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[17] Su-In Lee, Safiye Celik, Benjamin A Logsdon, Scott M Lundberg, Timothy J Martins, Vivian G Oehler, Elihu H Estey, Chris P Miller, Sylvia Chien, and Jin Dai. A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nature communications*, 9(1):1–13, 2018.

[18] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics*, 10(1):1–9, 2018.

[19] Shengchao Liu, Mehmet F Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in neural information processing systems*, 32, 2019.

[20] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, María Gordillo-Marañón, Fiona Hunter, Laura Junco, Grace Mugumbate, Milagros Rodriguez-Lopez, Francis Atkinson, Nicolas Bosc, Chris J Radoux, Aldo Segura-Cabrera, Anne Hersey, and Andrew R Leach. Chembl: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940, 2018.

[21] Varnavas D Mouchlis, Antreas Afantitis, Angela Serra, Michele Fratello, Anastasios G Papadiamantis, Vassilis Aidinis, Iseult Lynch, Dario Greco, and Georgia Melagraki. Advances in de novo drug design: from conventional to machine learning methods. *International journal of molecular sciences*, 22(4):1676, 2021.

[22] Peter Pogány, Navot Arad, Sam Genway, and Stephen D Pickett. De novo molecule design by translating from reduced graphs to smiles. *Journal of chemical information and modeling*, 59(3):1136–1146, 2018.

[23] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (moses): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11, 2020.

[24] Ofir Press, Noah A. Smith, and Omer Levy. Improving transformer models by reordering their sublayers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2996–3005, Online, July 2020. Association for Computational Linguistics.

[25] Oleksii Prykhodko, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1):1–13, 2019.

[26] Oleksii Prykhodko, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1):74, 2019.

[27] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

[28] Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1):120–131, 2018.

[29] M. Sicho, X. Liu, D. Svozil, and G. J. P. van Westen. Genui: interactive and extensible open source software platform for de novo molecular generation and cheminformatics. *J Cheminform*, 13(1):73, 2021.

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. *neurips*, 30, 2017.

[32] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

# 7 Appendix

Table 5: Validation Loss for different structure (En-1 represents sandwich encoder with coefficient 1 and de-1 represents sandwich decoder with coefficient 1)

| Validation Loss | Baseline | En-1 | En-2 | En-3 | De-1 | De-2 | Decoder-3 |
|---|---|---|---|---|---|---|---|
| Loss Value | 5.46e-6 | 5.3e-6 | 3.64e-6 | 6.73e-6 | 4.76e-6 | 5.97e-6 | 4.37e-6 |

Table 6: (pre-)training config for Sandwich Smile Transformer

| (pre-)training config | Sandwich Smile Transformer |
|---|---|
| optimizer | AdamW |
| base learning rate | 1e-5 |
| weight decay | 1e-5 |
| optimizer momentum | $\beta_1, \beta_2$=0.9, 0.999 |
| batch size | 8 |
| training epochs | 12 |
| learning rate schedule | cosine decay |
| transformer dropout | 0.4 |
| positional encoding dropout | 0.1 |
| warmup epochs | None |


(a) Encoder-3 (b) Decoder-3 (c) Baseline

Figure 6: Sandwich Transformer Models Latent Space HIV Molecules


(a) Encoder-3 (b) Decoder-3 (c) Baseline

Figure 7: Sandwich Transformer Models Latent Space BACE Molecules