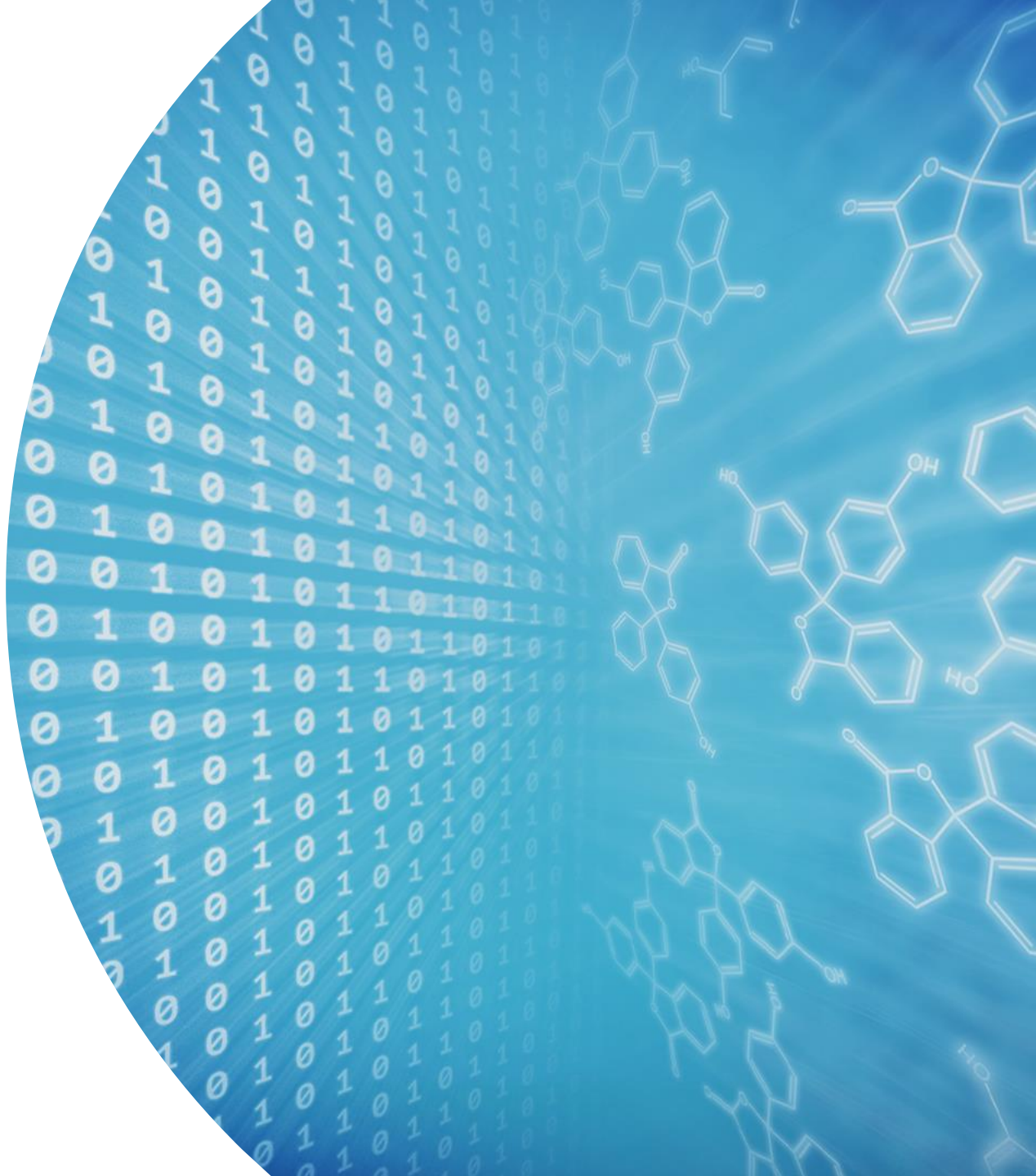
AstraZeneca

SMILES based deep modelling of molecules

Principal Scientist Esben Jannik Bjerrum

Deep Chemistry, Molecular, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden.



About the Speaker

1997-2003
- Master of Pharmaceutical Sciences

2004-2008
- Ph.D Computational Chemistry

2008-2010
- Pharma Industry IT Applications Specialist

2011-2012
- Postdoc Protein NMR

2013
- Internships in Pharma and Biotech

2014-2015
- Postdoc Chemometrics

2015-2018
- Independent Pharma IT Consultant and Researcher

2018 -
- Principal Scientist Astrazeneca

Esben Jannik Bjerrum

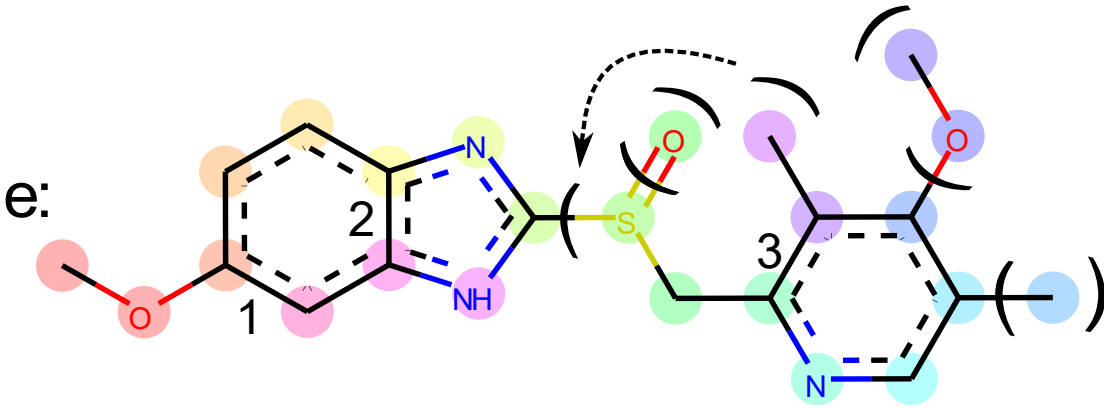
Sometimes blogging at www.cheminformania.com

Agenda

1. SMILES
2. How to turn SMILES into numbers
3. PySMILESUtils
4. Data Augmentation of SMILES
5. Reading in and Reading out SMILES
6. Sampling Chemical Space
7. Advanced Architectures
8. Encoder Architectures
9. Self-supervised Pretraining
10. Alternatives to SMILES

Simplified Molecular Input Line Entry System (SMILES)

Molecule:



SMILES: COc1ccc2nc(S(=O)Cc3ncc(C)c(OC)c3C)[nH]c2c1

- **A sequence format for molecules**
- Letters denotes atoms (Single e.g. ”C” or dual e.g. ”Si”), lowercase signifies ”aromatic”
- () start and end a branch
- Bonds -, = , # (triple), : (aromatic)
- Numbers make extra bonds (most often rings, but consider C1.C1)
- [] contains atoms with specified properties (e.g. charges and hydrogens)
- Chirality possible with @ and @@, and cis/trans with /=/ and \=/
- A good web-ressource: https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html

Weininger, D. SMILES, **a Chemical Language** and Information System: 1: Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28* (1), 31–36. https://doi.org/10.1021/ci00057a005 .

Handling with toolkits – RDKit example

```
from rdkit import Chem

supplier = Chem.SDMolSupplier("../../rdkit/Docs
/Book/data/bzr.sdf")

for mol in supplier:
    if mol: # parse error gives "None"
        print(Chem.MolToSmiles(mol))
```

```
CN(C)Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2
Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2
O=C1CN=C(c2ccccn2)c2cc(Br)ccc2N1
CNC1=Nc2ccc(Cl)cc2C(c2ccccc2)=[N+]([O-])C1
CN1C(=O)CC(=O)N(c2ccccc2)c2cc(Cl)ccc21
O=C1CN=C(c2ccccc2Cl)c2cc([N+](=O)[O-])ccc2N1
…
```

Other toolkits can also handle SMILES: see examples at Chemistry Toolkit Rosetta: https://ctr.fandom.com/wiki/Chemistry_Toolkit_Rosetta_Wiki

Beware: Not a universal standard, differences may exists between toolkits, e.g. Canocalization!

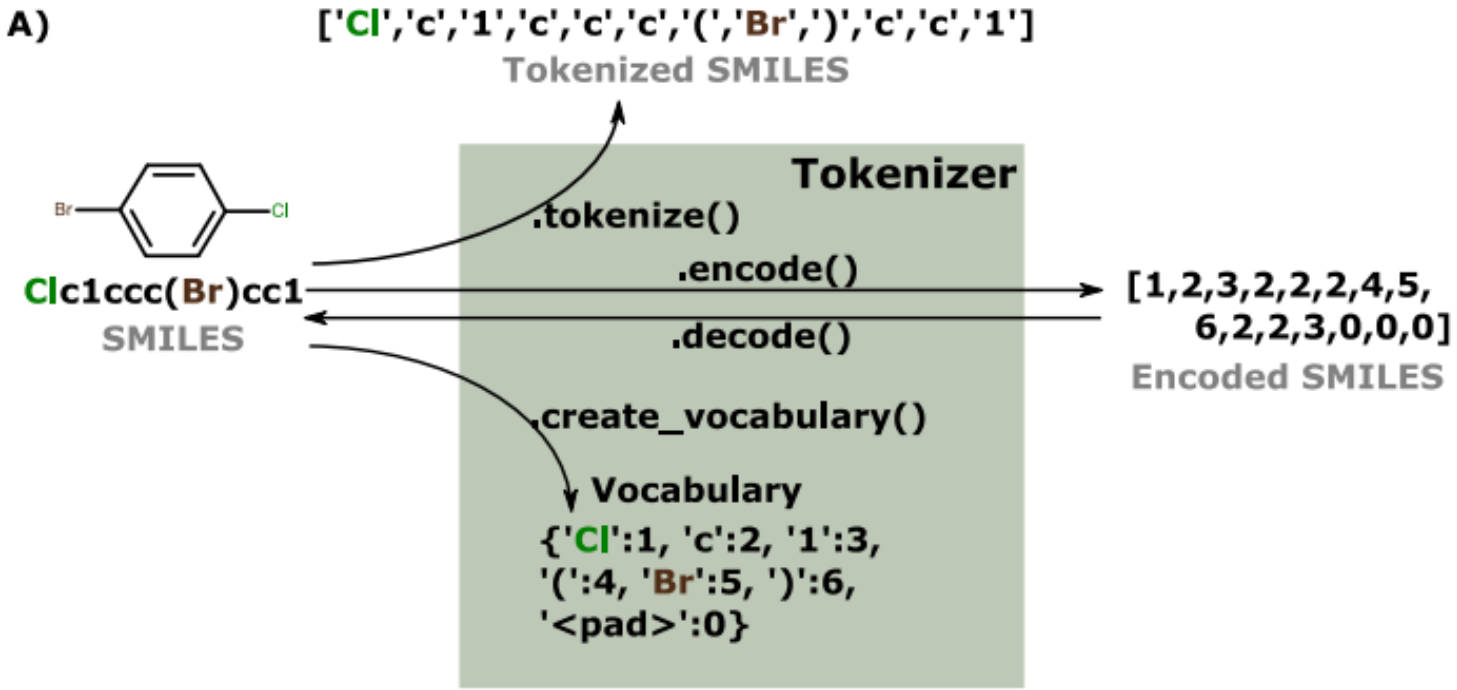
Tokenization - Vectorization

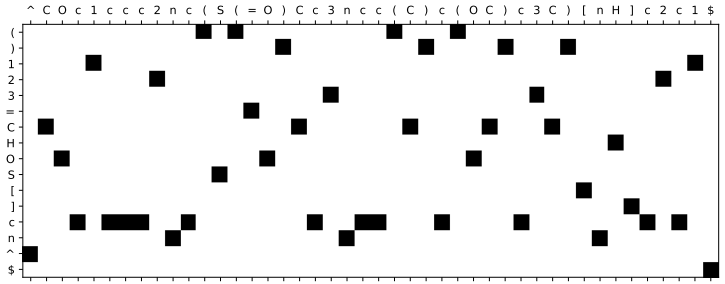
First the strings are divided into tokens, single characters or smaller units of text, e.g. ”Cl”, ”Br”

All identified tokens are stored in an indexed list (vocabulary)

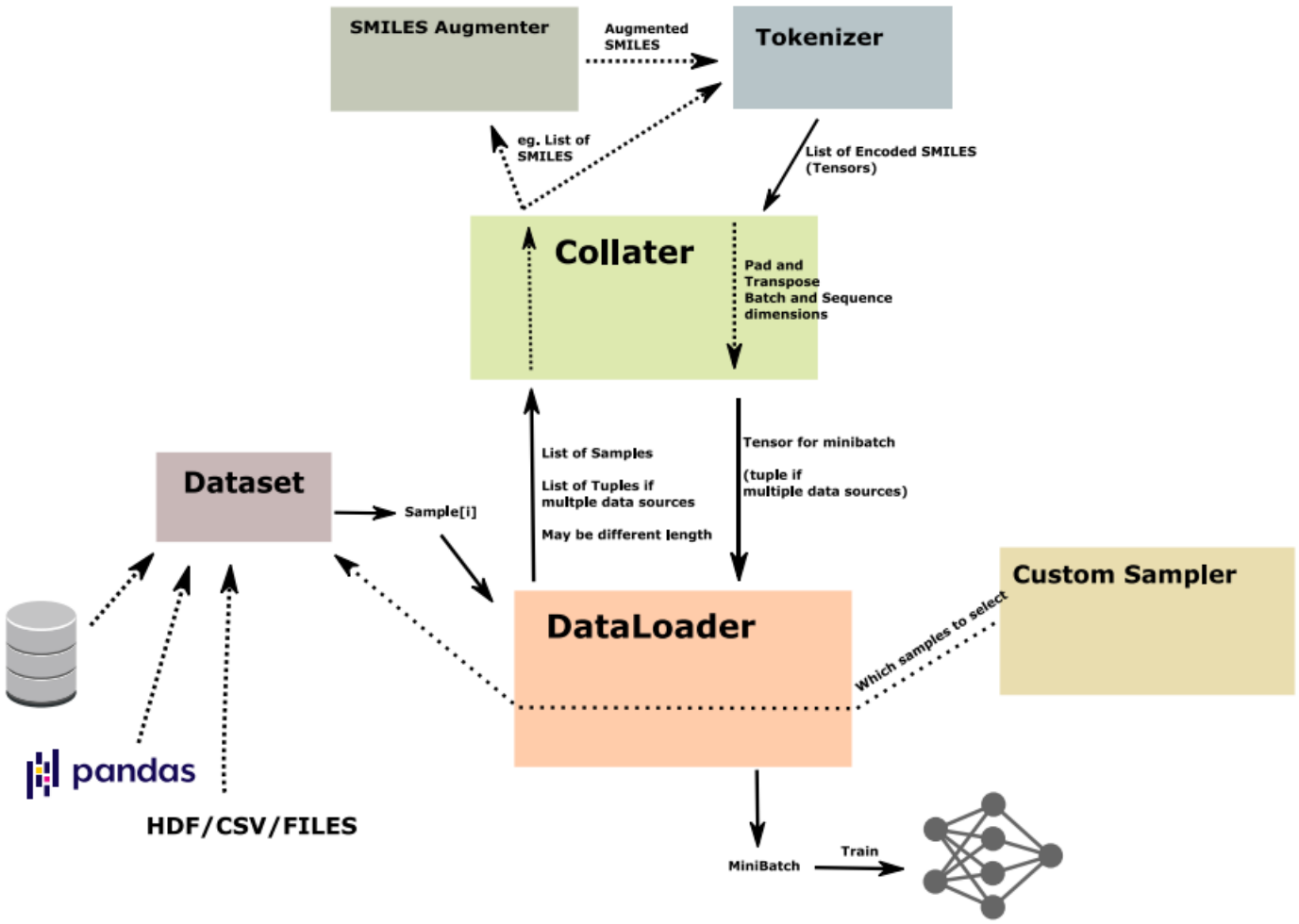
The vocabulary is used to encode and decode the SMILES into integers

Numbers can be one-hot encoded or used in an embedding layer





PySMILESUtils



Package for SMILES based vectorization and data handling

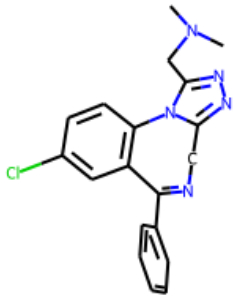
Aimed at PyTorch but otherwise model agnostic

Open Source: https://github.com/MolecularAI/pysmilesutils

Classes for Tokenization, Encoding/decoding, datasets, custom data loaders and batch samplers

Tokenization – Vectorization Example

```
from rdkit.Chem import PandasTools
data = PandasTools.LoadSDF("../../rdkit/Docs/Book/data/bzr.sdf")
data.ACTIVITY = pd.to_numeric(data.ACTIVITY)
data["SMILES"] = data.ROMol.apply(Chem.MolToSmiles)
data.head(1)
```

	ACTIVITY	ID	ROMol	SMILES
0	6.87	Adinazolam		CN(C)Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2

```
from pysmilesutils.tokenize import SMILESAtomTokenizer

tokenizer = SMILESAtomTokenizer(smiles=list(data.SMILES.values))
print(tokenizer.vocabulary)

encoded = tokenizer.encode(list(data.SMILES.values))
print(encoded[0])
print(tokenizer.decode(encoded[0:1]))
```

{' ': 0, '^': 1, '&': 2, '?': 3, 'C': 4, 'N': 5, '(': 6, ')': 7, 'c': 8, '1': 9, 'n': 10, '2': 11, '-': 12, 'Cl': 13, '=': 14, 'O': 15, 'Br': 16, '[': 17, '+': 18, ']': 19, 's': 20, 'F': 21, '#': 22, 'S': 23, 'H': 24, '3': 25}

tensor([1, 4, 5, 6, 4, 7, 4, 8, 9, 10, 10, 8, 11, 10, 9, 12, 8, 9,
8, 8, 8, 6, 13, 7, 8, 8, 9, 4, 6, 8, 9, 8, 8, 8, 8, 8,
9, 7, 14, 5, 4, 11, 2])

['CN(C)Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2']

Efficient integration into deep learning – Molecular Transformer example

#1: Customizing the collate function in the data loader enables CPU parallelism in Pytorch dataloader queing.

#2: Transformers are squared complexity on input length (self attention layers)

Different SMILES length are padded to same size in mini-batches

If approximately similar sized SMILES gets bundled, a big speedup can be achieved.

These classes are available in PySMILESutils

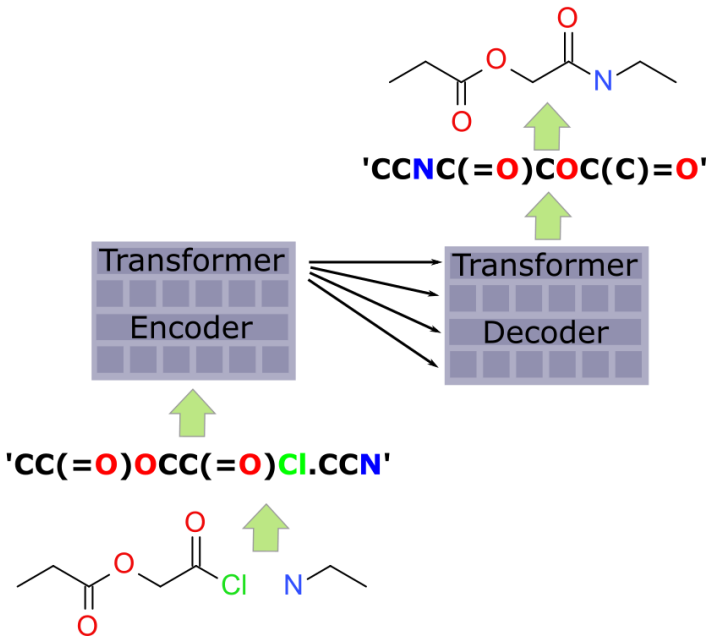
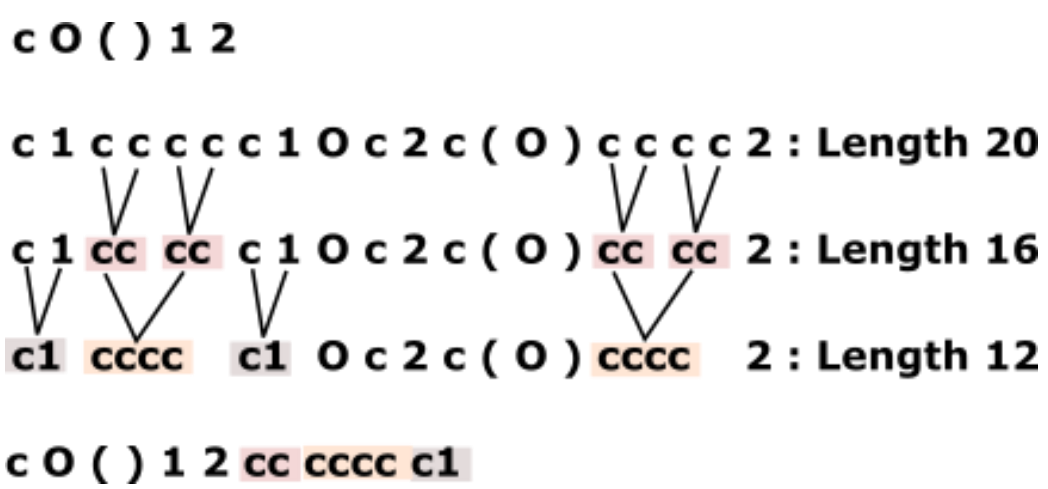


Table 1: Training efficiency comparison of different code organizations.

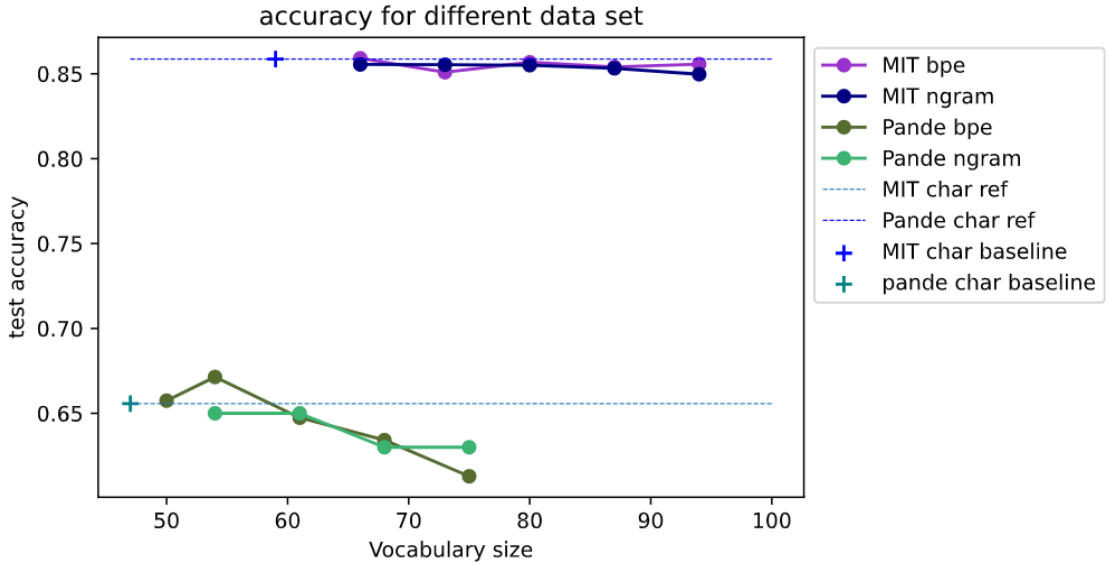
Code strategy	Epoch time (s)	Samples/s	Avg. batch length
In training loop	110	454	113
In Dataset	115	434	113
In Collate function	66	757	113
+ Sorted lengths	36	1390	60
+ BucketBatchSampling	39	1299	61

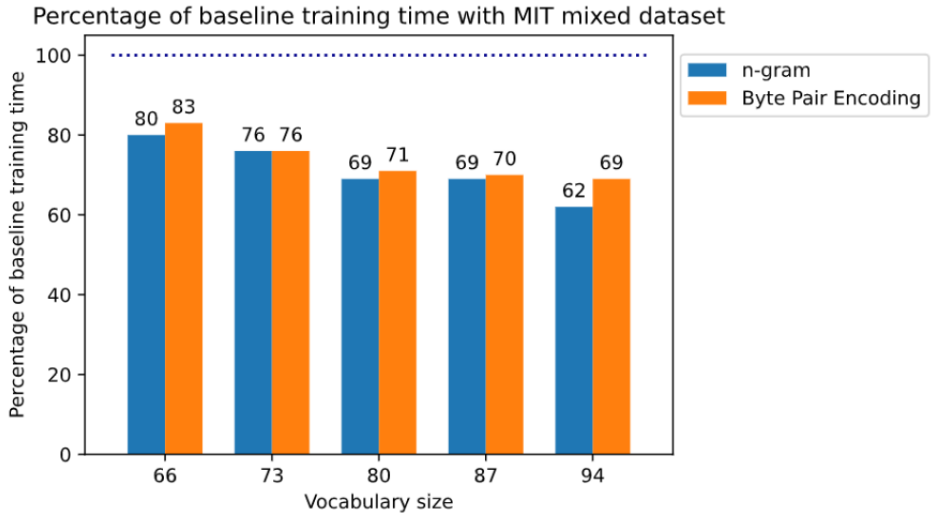
[1] E. J. Bjerrum, T. Rastemo, R. Irwin, C. Kannas, and S. Genheden, "PySMILESUtils – Enabling deep learning with the SMILES chemical language," *ChemRxiv*, 2021.

Other Tokenization Schemes – Byte Pair Encoding



- Iteratively looking for common pairs of tokens and combining
- Expands vocabulary – shrinks length
- Quickly seems detrimental for small datasets, but decreased training time



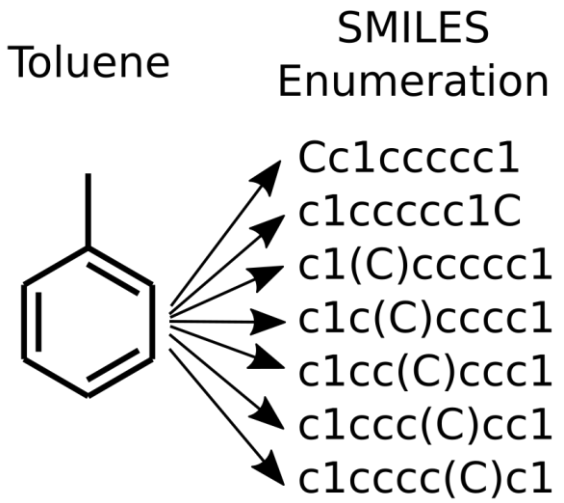


T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *E*
71, 2018, doi: 10.18653/v1/d18-2012.

X. Li and D. Fourches, "SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning," 2020, doi: 10.26434/chemrxiv.12339368.

Data augmentation

- Zooming, cropping, mirroring, flipping, rotation, hue, color, contrast, etc. + combinations



- Canonical SMILES ensures a 1:1 relationship between molecule and SMILES
- SMILES enumeration generate multiple SMILES for the same molecule
- Many names: SMILES enumeration, SMILES multiplicity, SMILES randomization, SMILES augmentation

Bjerrum, Esben Jannik. 2017. "SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules." http://arxiv.org/abs/1703.07076

SMILES augmenttion in practice

[1]:
```
from rdkit import Chem
from rdkit.Chem.Draw import IPythonConsole
```

[2]:
```
drugname = "Omeprazol"
mol = Chem.MolFromSmiles("CC1=CN=C(C(=C1OC)C)CS(=O)C2=NC3=C(N2)C=C(C=C3)OC")
mol
```

[2]:

[3]:
```
print("%i Atoms, %i Rings"%(mol.GetNumAtoms(), Chem.GetSSSR(mol)))
```

```
24 Atoms, 3 Rings
```

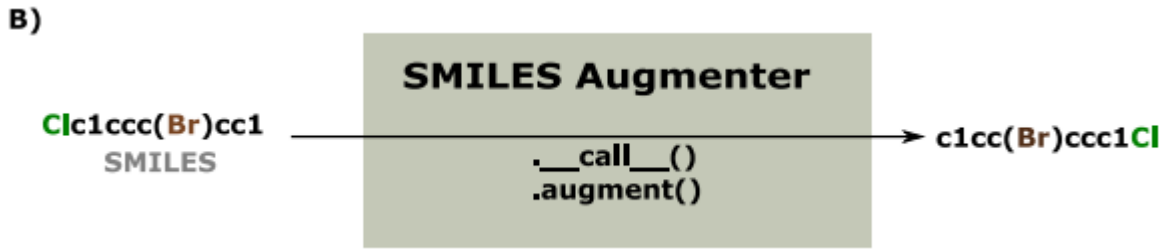
[ ]:
```
s = set()
i = 0

while True:
    l = len(s)

    smiles = Chem.MolToSmiles(mol, doRandom = True)

    s.add(smiles)
    if len(s) > l:
        print("\r %i \t %s"% ( len(s), smiles), end = '')
        i = 0
    else:
        i = i + 1
    if i > 1000:
        break
print()
print("Done")
```

[ ]:

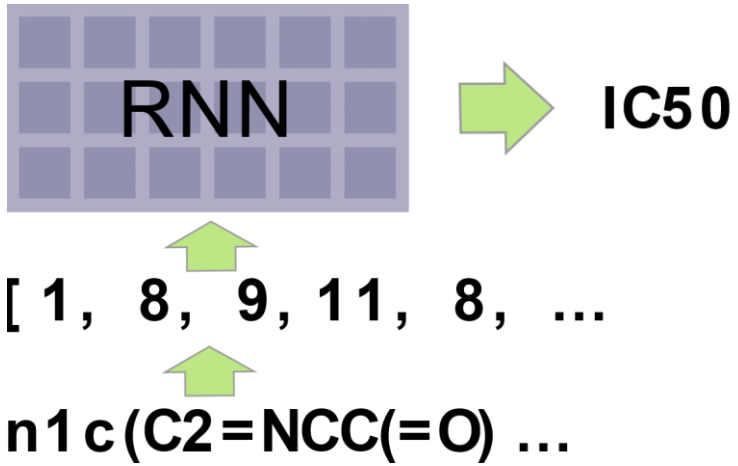
PySMILES Example

```
from pysmilesutils.augment import SMILESAugmenter

augmenter = SMILESAugmenter(restricted=True)
print(augmenter([ data.iloc[2].SMILES] * 2), end = "\n\n")

#Chainable
print(tokenizer(augmenter([data.iloc[2].SMILES] * 2)), end =
 "\n\n")

#Useful for inference and testing
augmenter.active = False
print(augmenter([ data.iloc[2].SMILES] * 2), end = "\n\n")
```

```
['n1c(C2=NCC(=O)Nc3ccc(Br)cc32)cccc1',
'N1C(=O)CN=C(c2ccccn2)c2c1ccc(Br)c2']

[tensor([ 1,  8,  9, 11,  8,  6,  8,  8,  8,  6, 16,  7,  8,  9,  7,  5,  4,  6,
         14, 15,  7,  4,  5, 14,  4, 11,  8,  9, 10,  8,  8,  8,  8,  9,  2]),
 tensor([ 1,  4,  9,  5, 14,  4,  6,  8, 11,  8,  8,  8,  8, 10, 11,  7,  8, 11,
          8,  8,  6, 16,  7,  8,  8,  8, 11,  5,  4,  9, 14, 15,  2])]

['O=C1CN=C(c2ccccn2)c2cc(Br)ccc2N1',
'O=C1CN=C(c2ccccn2)c2cc(Br)ccc2N1']
```



Improving Results on Read-IN

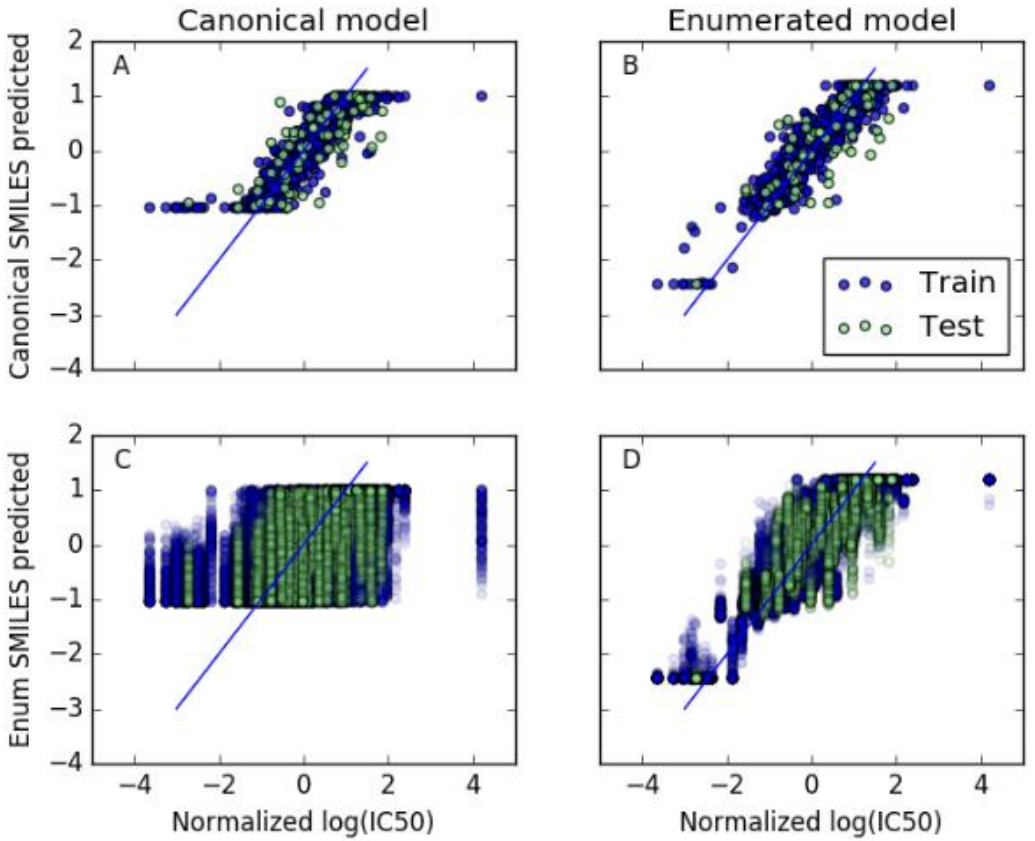


SMILES modelled as either Canonical or Augmented using LSTM RNN

Augmented Model perfectly handle Canonical SMILES but not the other way

Augmentation improved statistics on test set

Sampling multiple SMILES forms and averaging further improves results



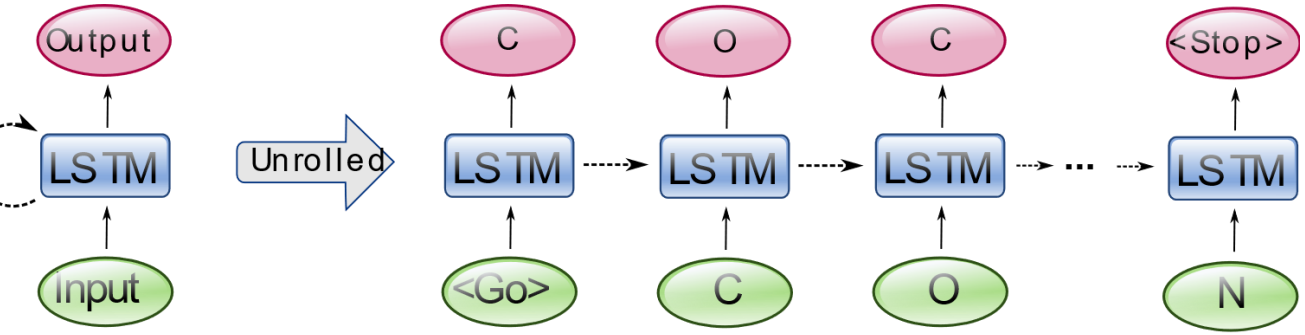
Test set performance

Canonical	R2 0.56, RMS 0.62
Augmented	R2 0.66, RMS 0.55

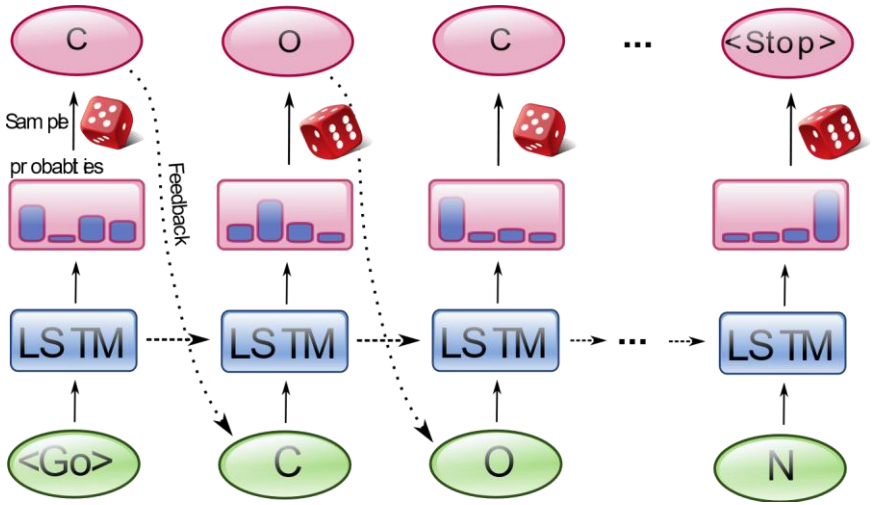
[1] E. J. Bjerrum, "SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules," Mar. 2017, Accessed: Mar. 06, 2019. [Online]. Available: http://arxiv.org/abs/1703.07076.

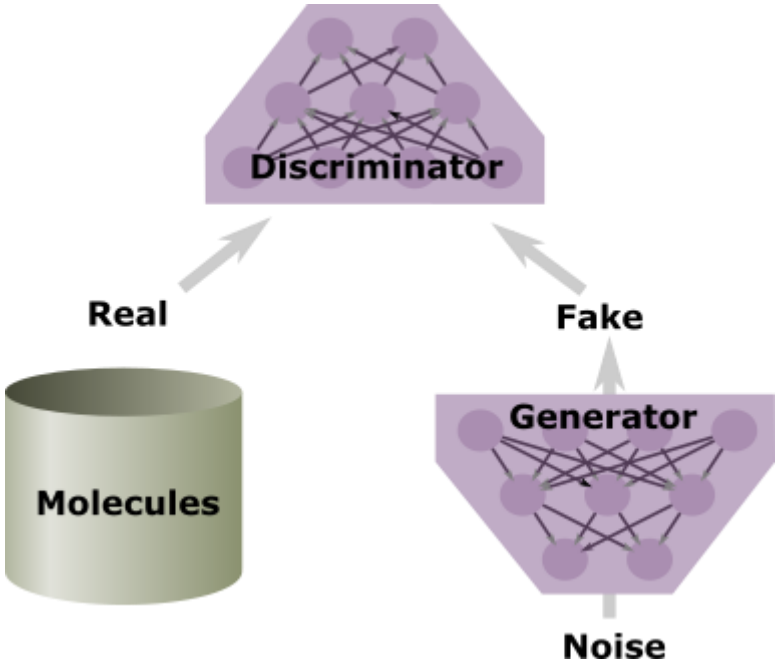
Training and Sampling using Deep Learning on SMILES

Training:



Sampling:





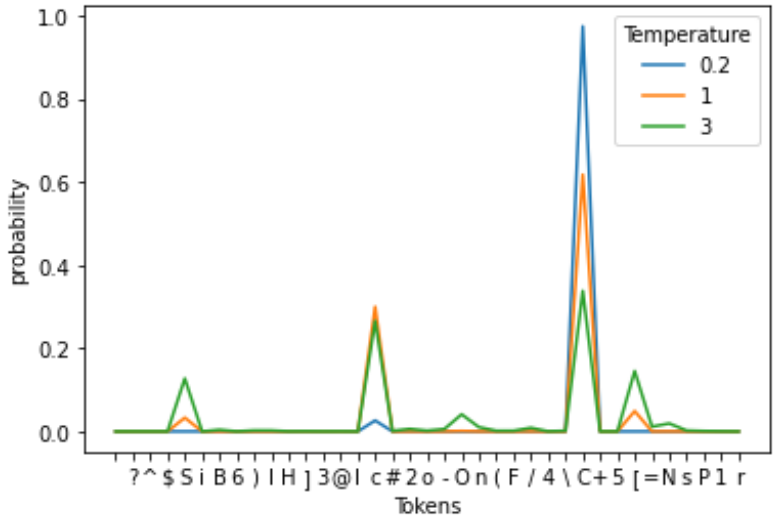
- Started with recurrent neural networks or GANs a few years ago
- Allows to read-in and **read-out** molecules
- Models learn the rules of chemistry AND the properties of the dataset

Sampling – Greedy & Multinomial - Temperature

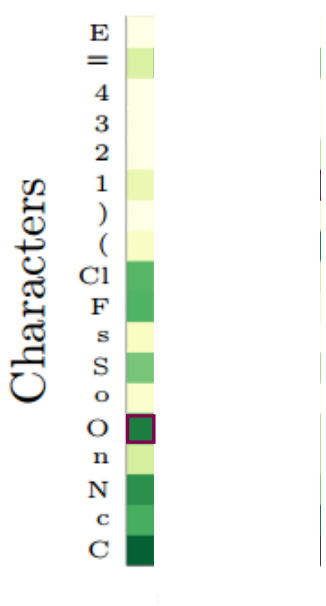
- **Greedy**: Take most probable token as next sample
  - One output
- **Multinomial**: Sample next token using probability distribution
  - Multiple outputs
  - Nondeterministic
- **Temperature scaling** can be used to ”tune” the output

Softmax with temperature scaling

$$\sigma(z)_i = \frac{e^{z_i/T}}{\sum_{J=1}^{k} e^{z_i/T}}$$



De novo compound generation using recurrent neural networks



Sampled Characters

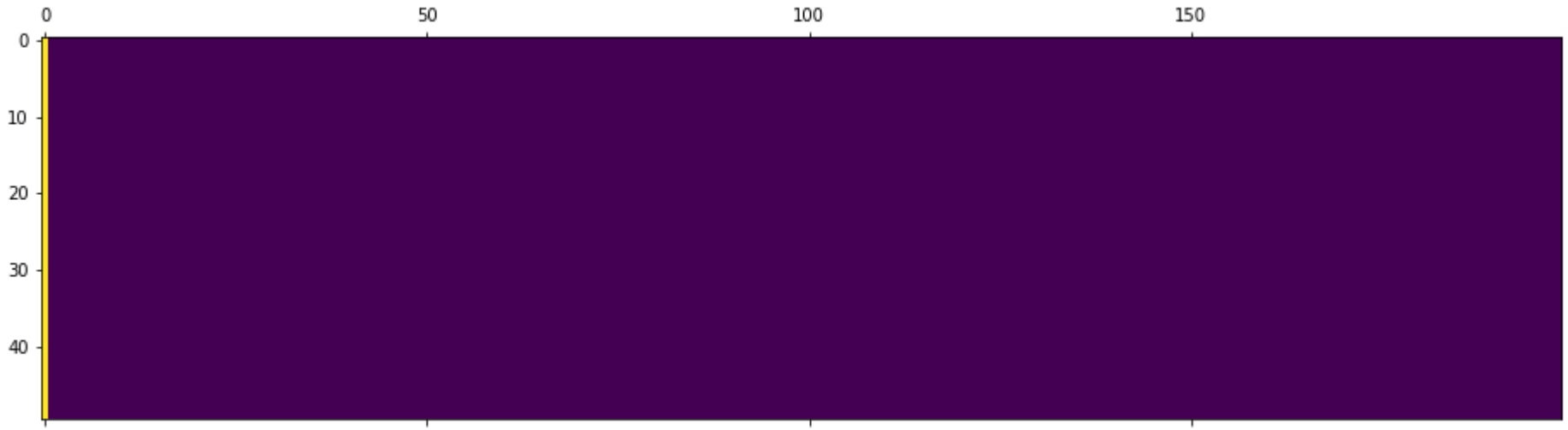
Probabilities

Molecule

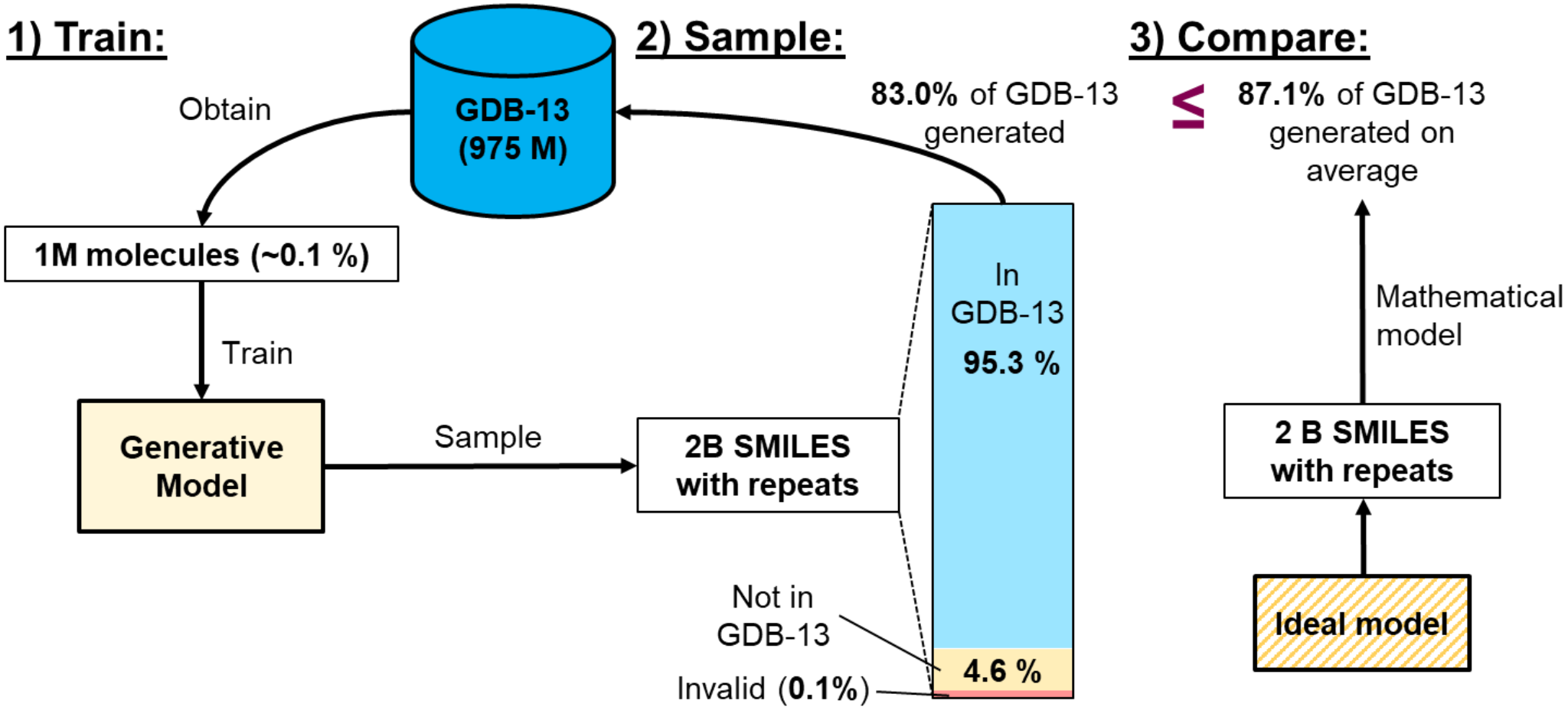
Beam Search algorithm

1. Keep top-K most probable predictions so far
2. Sample probability distributiion of each
3. sort by summed probability (log likelihood) => 3 until end

- Multiple Outputs, Deterministic, Sorted by probability



Assessing the quality of molecular generative models

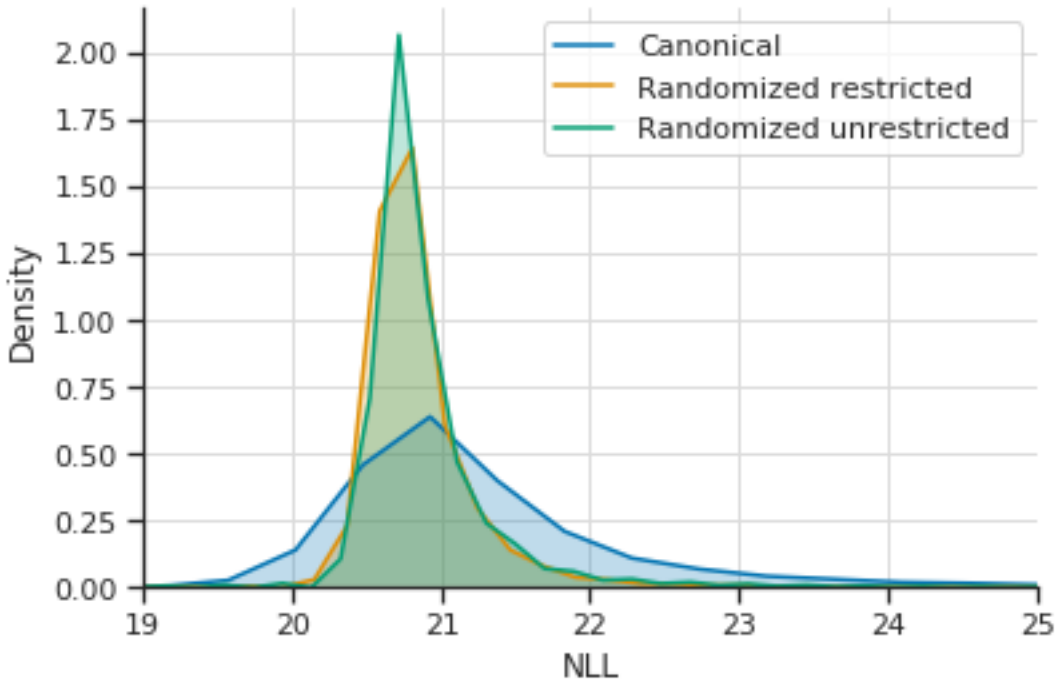


1) Arús-Pous J, Blaschke T, Ulander S, et al (2019) Exploring the GDB-13 chemical space using deep generative models. J Cheminform 11:20. https://doi.org/10.1186/s13321-019-0341-z

2) Arús-Pous J, Johansson SV, Prykhodko O, et al (2019) Randomized SMILES strings improve the quality of molecular generative models. J Cheminform 11:71. https://doi.org/10.1186/s13321-019-0393-0

SMILES enumeration increases Chemical Space Coverage

More uniform



More Complete

Set	SMILES	Validity	Completeness
1M	Canonical	0.994	0.836
	Randomized	**0.999**	**0.953**
10K	Canonical	0.905	0.445
	Randomized	0.974	0.715
1K	Canonical	0.504	0.167
	Randomized	0.812	0.392

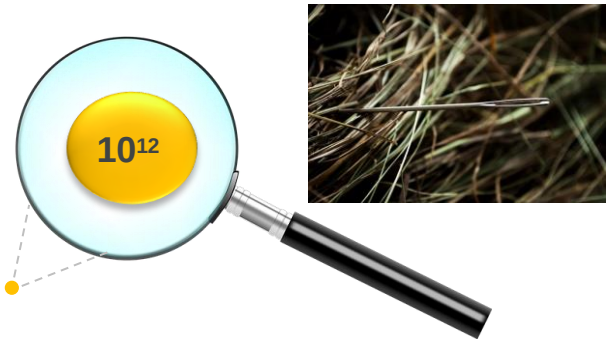
GDB-13 is 975 million molecules

(1) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. Randomized SMILES Strings Improve the Quality of Molecular Generative Models. J. Cheminform. 2019, 11 (1), 71. https://doi.org/10.1186/s13321-019-0393-0

Generative Model vs Enumeration for molecular discovery

Physical Storage Size **Size of Molecular Space**

Databases/ Traditional Enumeration



0.0017 ng of H

Generative Model

Drug-like chemical space estimated between 10^{30} to 10^{60}

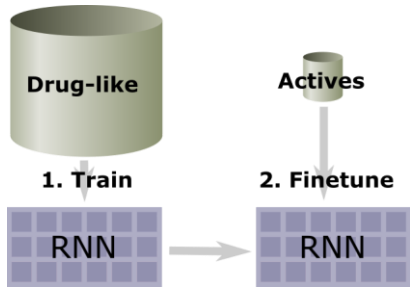
Generative models **do not contain any** explicit molecules but generate them probabilistically

Generative models can sample **practically unlimited** chemical space

All of chemical space, or target desireable chemical space?

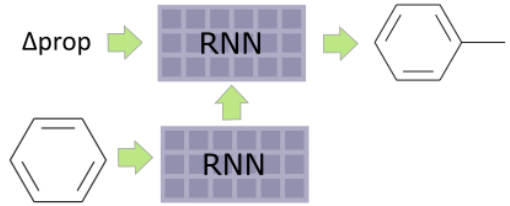
Advanced architectures and steering of generation

Transfer Learning

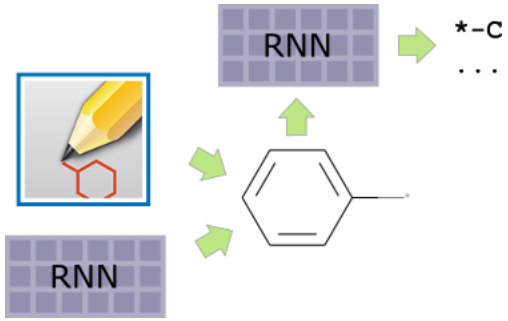


Directly Steered Optimization

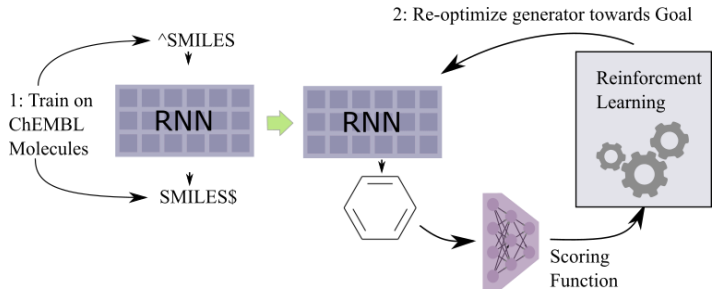
Chemical intuition via MMP Transcoder



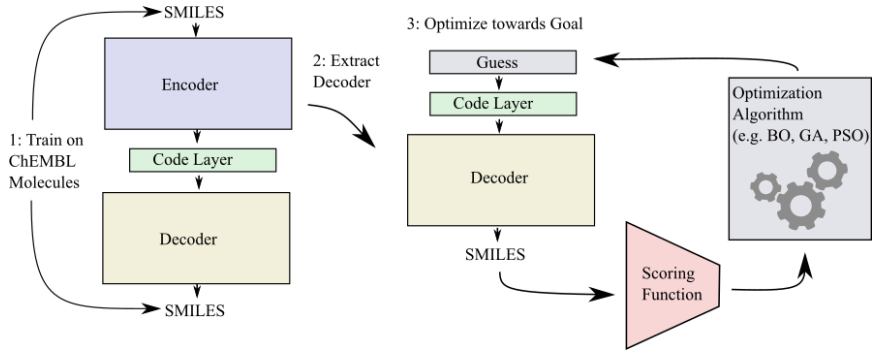
AI Library Generation



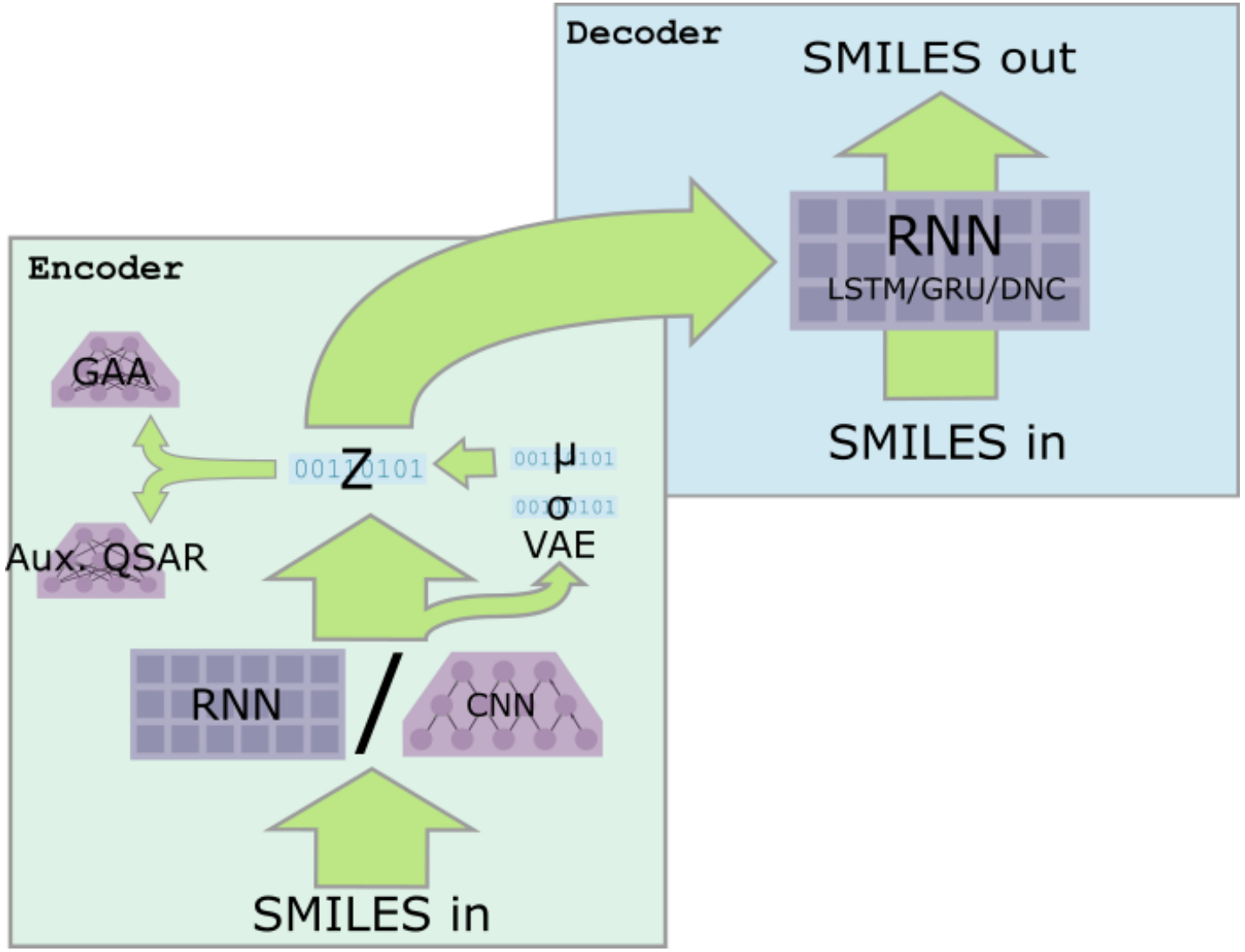
Optimize using reinforcement learning



Optimize using latent space of an autoencoder

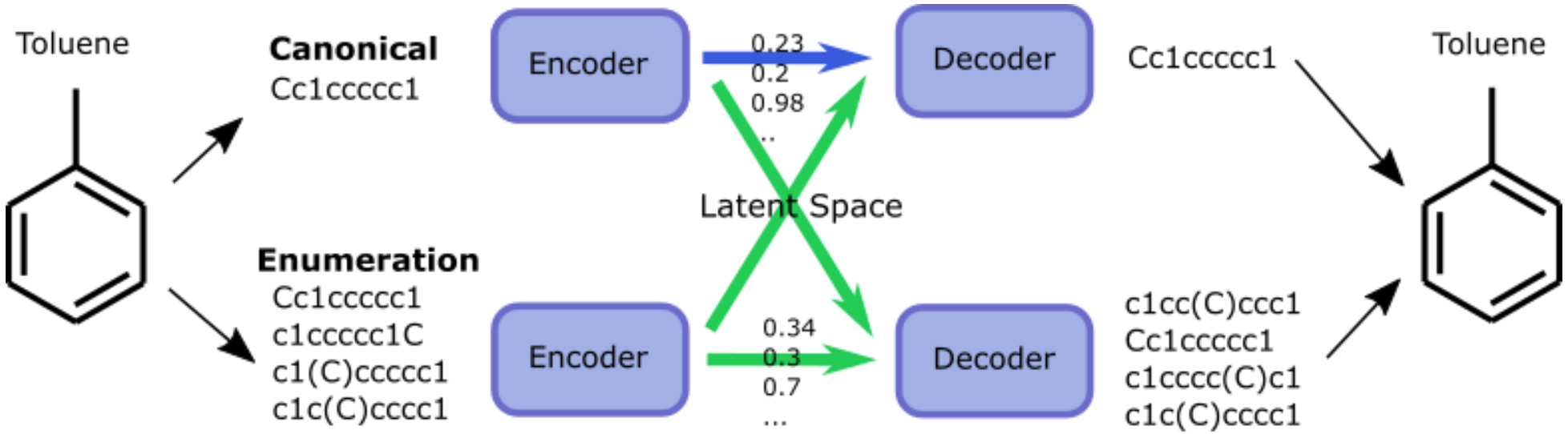


SMILES Based Autoencoders



Example: Gómez-Bombarelli, Rafael et al. 2018. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules." *ACS Central Science* 4(2): 268–76. Preprint from 2016!

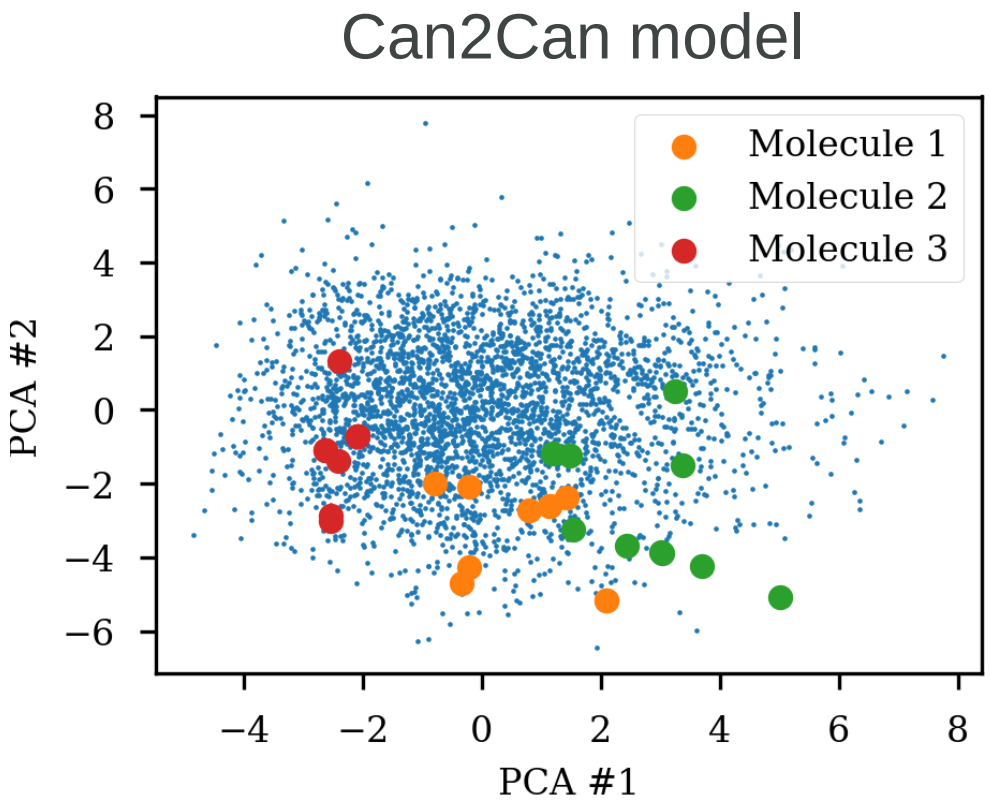
HeteroEncoders

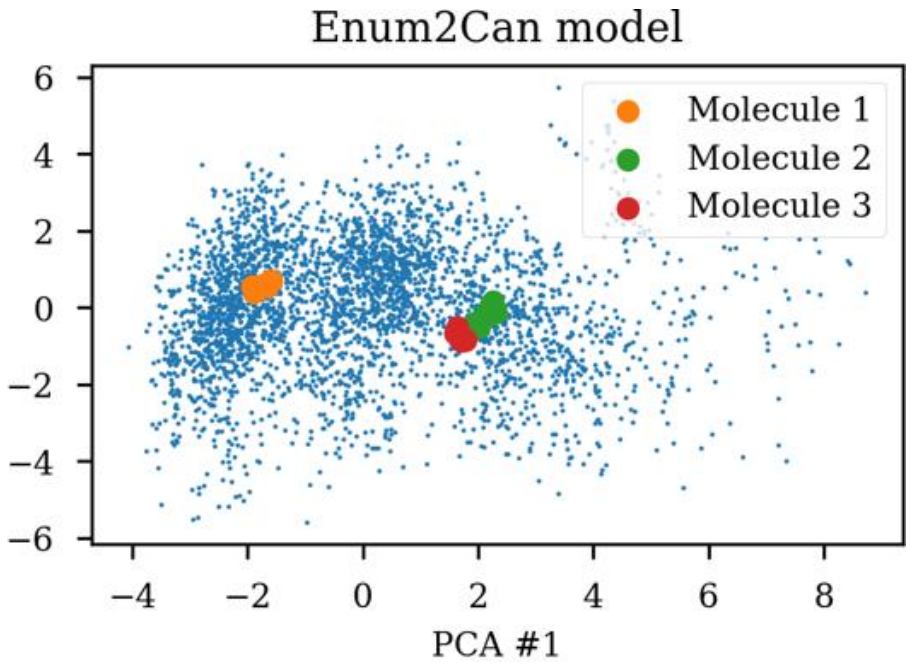


Also possible with InChi's and from chemical images

[1] E. J. Bjerrum and B. Sattarov, "Improving Chemical Autoencoder Latent Space and Molecular De Novo Generation Diversity with Heteroencoders," Biomolecules, 2018, doi: 10.3390/biom8040131.

Projection of non-canonical SMILES into latent space





Using SMILES augmentation ensures that the same molecule get same position in latent space

[1] E. J. Bjerrum and B. Sattarov, "Improving Chemical Autoencoder Latent Space and Molecular De Novo Generation Diversity with Heteroencoders," Biomolecules, 2018, doi: 10.3390/biom8040131.

Latent vectors as a base for Quantitative structure-activity models (QSAR)

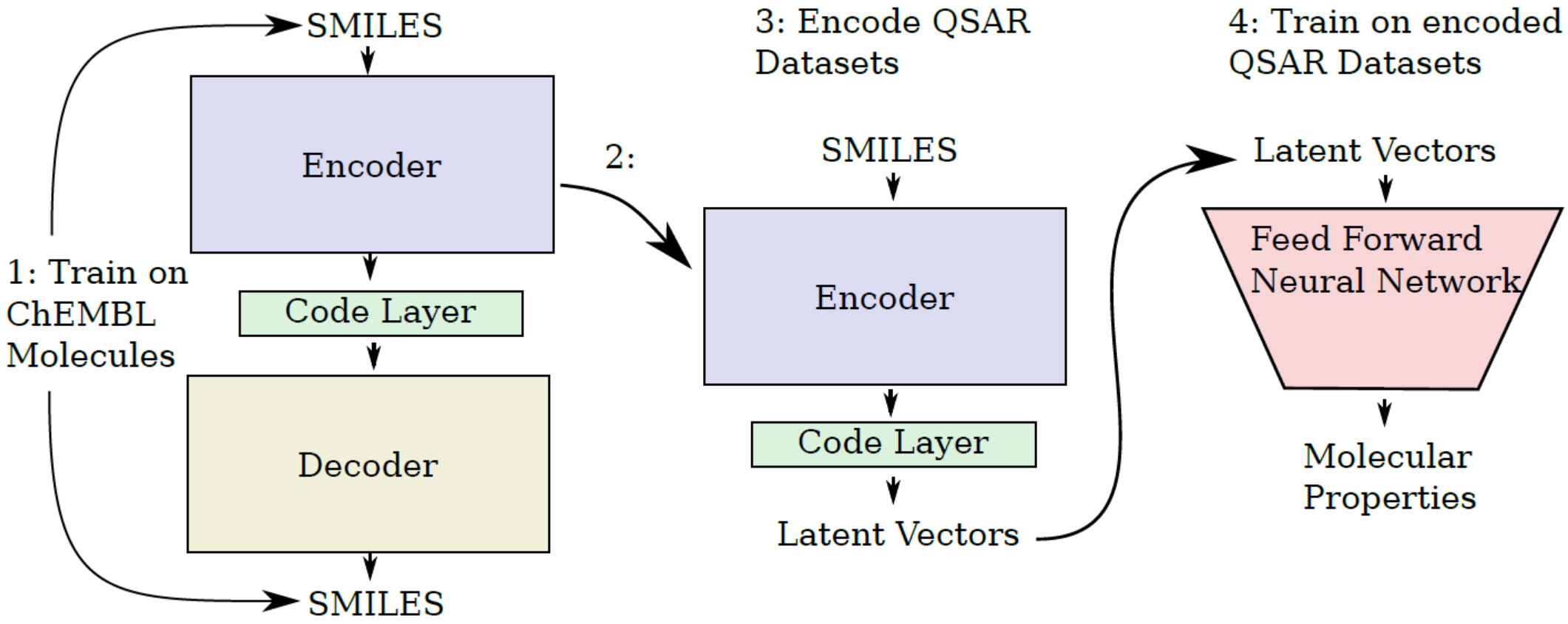


Figure adapted from : Bjerrum, Esben Jannik, and Boris Sattarov. 2018. “Improving Chemical Autoencoder Latent Space and Molecular De Novo Generation Diversity with Heteroencoders.” *Biomolecules*.

Latent Space gets more chemicaly relevant

RMSEP of 5 datasets modelled using deep neural networks

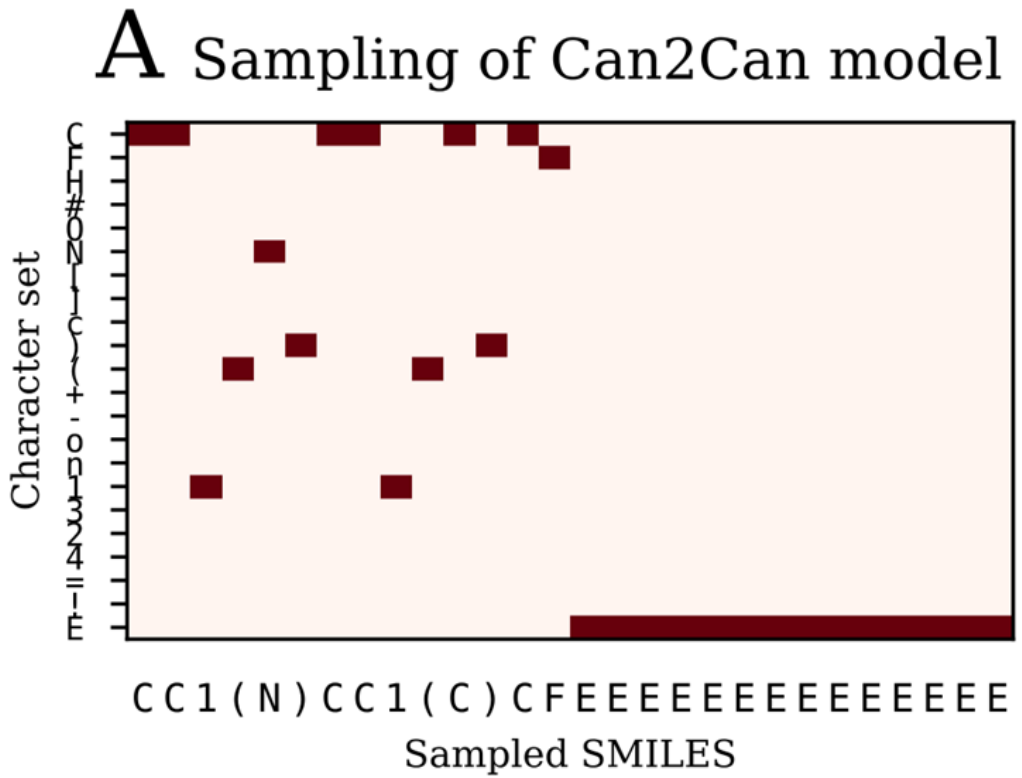
improvement

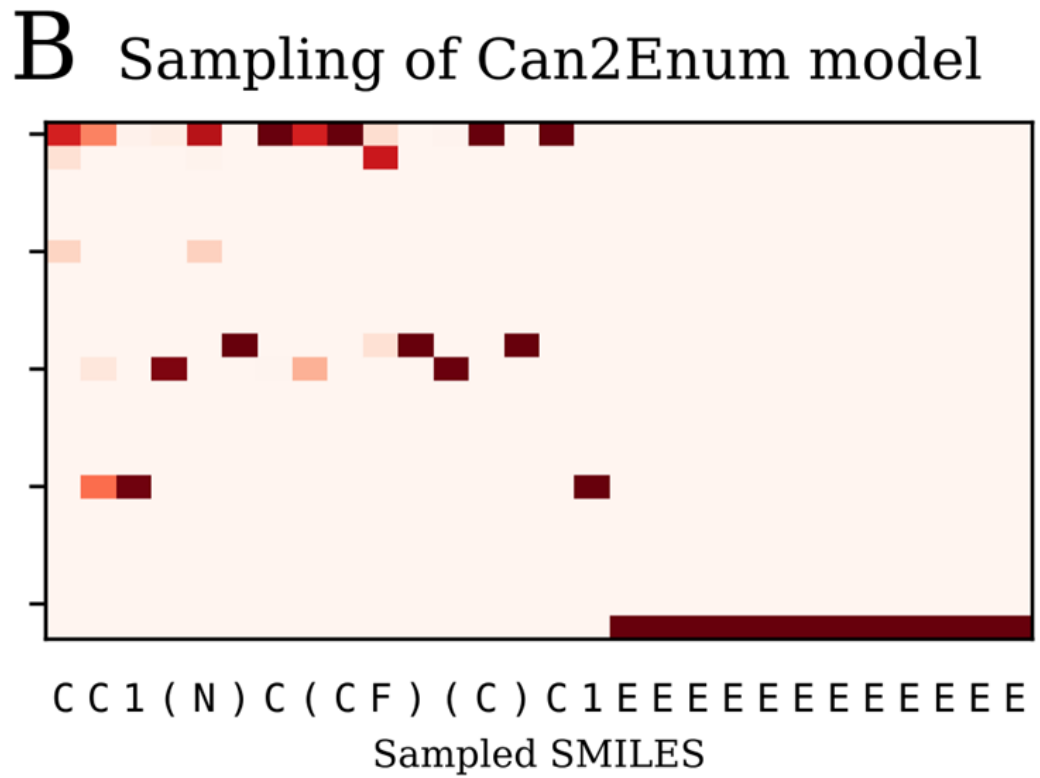
	IGC50	LD50	BCF	Solubility	MP	Norm Mean
Enum2Enum	**0.43**	**0.54**	0.71	**0.65**	**37**	0.75
Can2Enum	0.46	**0.54**	**0.69**	0.69	**37**	0.77
Enum2Can	0.46	0.57	0.71	0.66	38	0.78
Can2Can	0.53	0.62	0.79	0.87	43	0.89
ECFP4	0.62	0.59	0.94	1.21	43	1.00

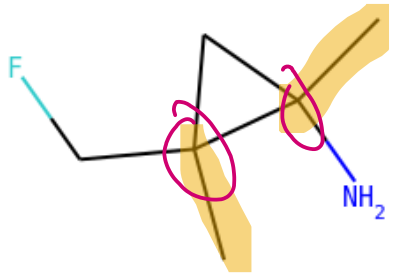
ECFP4 performance low when compared to literature, Enum2Enum close

Bjerrum, Esben Jannik, and Boris Sattarov. 2018. "Improving Chemical Autoencoder Latent Space and Molecular De Novo Generation Diversity with Heteroencoders." *Biomolecules*.

One:Many sampling from latent space point



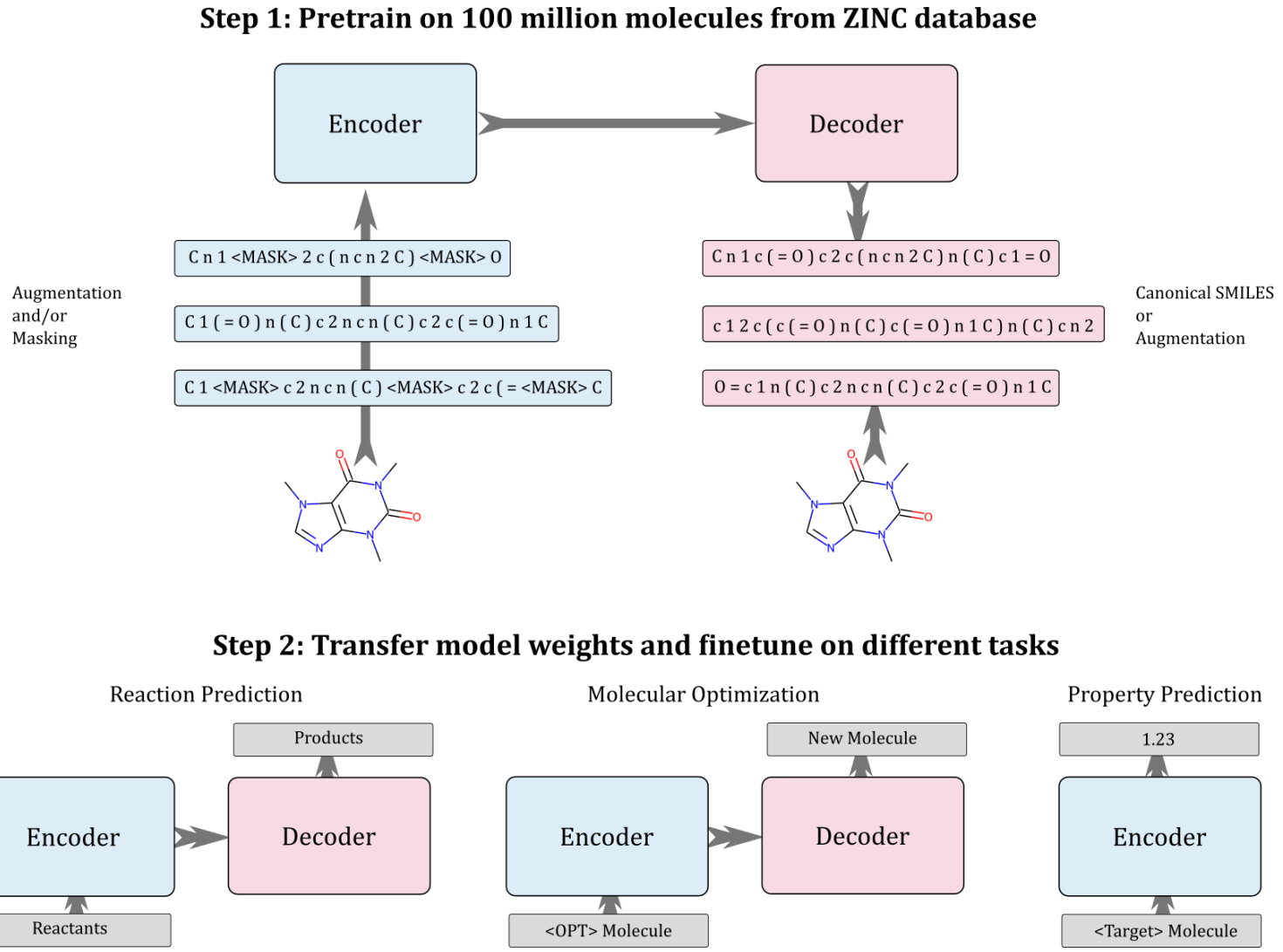




Using augmentation for Pretraining – then Finetuning

Combining token masking and

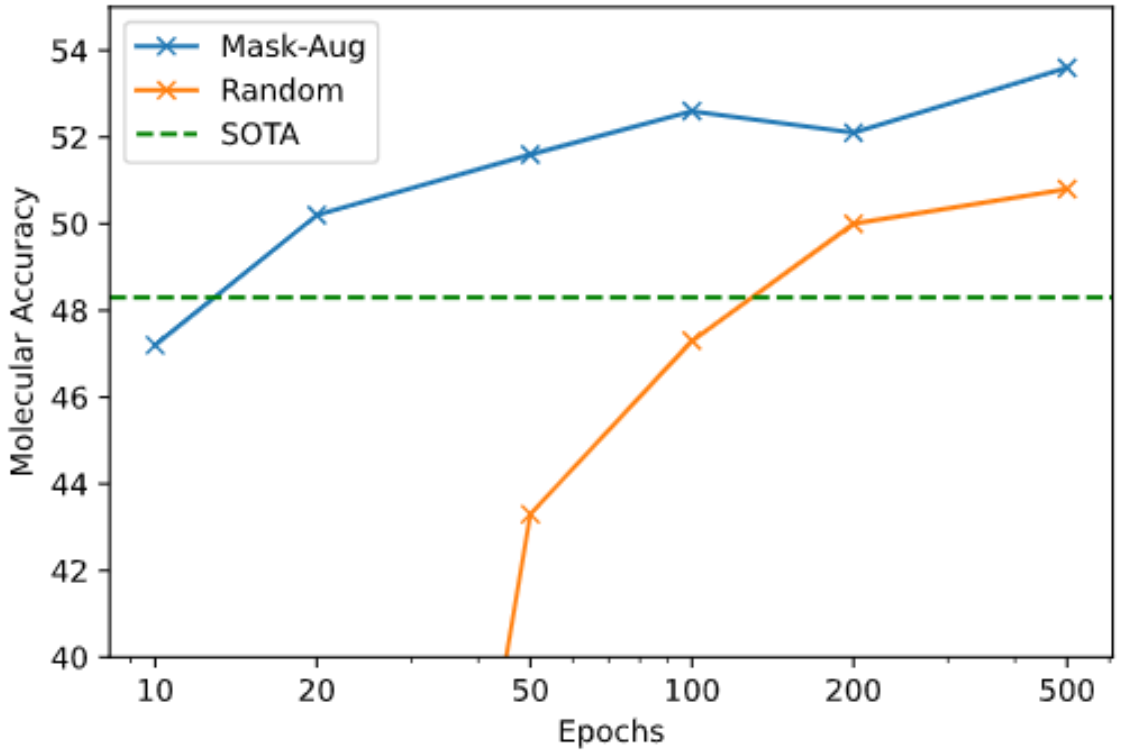
SMILES augmentation



[1] R. Irwin, S. Dimitriadis, J. He, and E. Bjerrum, "Chemformer: A Pre-Trained Transformer for Computational Chemistry," *ChemRxiv*, 2021, doi: 10.33774/chemrxiv-2021-v2pnn.

Pretraining decreases fine-tuning training time and increases performance

- Top-1 molecular accuracy Retrosynthesis Prediction
- The pre-trained model outperform state-of-the-art with less than 30 minutes of fine-tuning
- 50 epochs of fine-tuning provides better performance than 500 epochs from random initialisation



Random: Randomly initialized weights (No Pretraining).

SOTA: Tetko, Igor V., et al. "State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis." *Nature communications* 11.1 (2020): 1-11.

[1] R. Irwin, S. Dimitriadis, J. He, and E. Bjerrum, “Chemformer: A Pre-Trained Transformer for Computational Chemistry,” *ChemRxiv*, 2021, doi: 10.33774/chemrxiv-2021-v2pnn.

General performance improvement across different transformation tasks

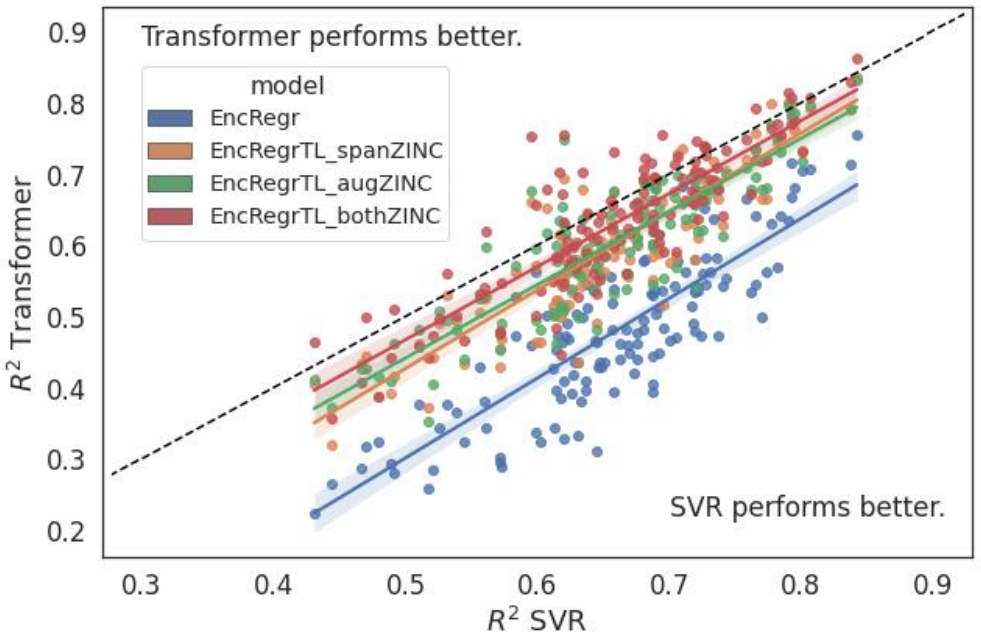
- Top-1 performance after 12 hours of training

Pre-Training	Forward Prediction (%) ~500k samples	Retrosynthesis (%) ~50k samples	Molecular Optimisation (%)
-	91.1	50.8	69.5
Mask	91.2	52.1	**72.1**
Augmentation	91.1	51.8	71.2
Combined	**91.8**	**53.6**	69.7
SOTA	91.1*	48.3*	66.6%.‡

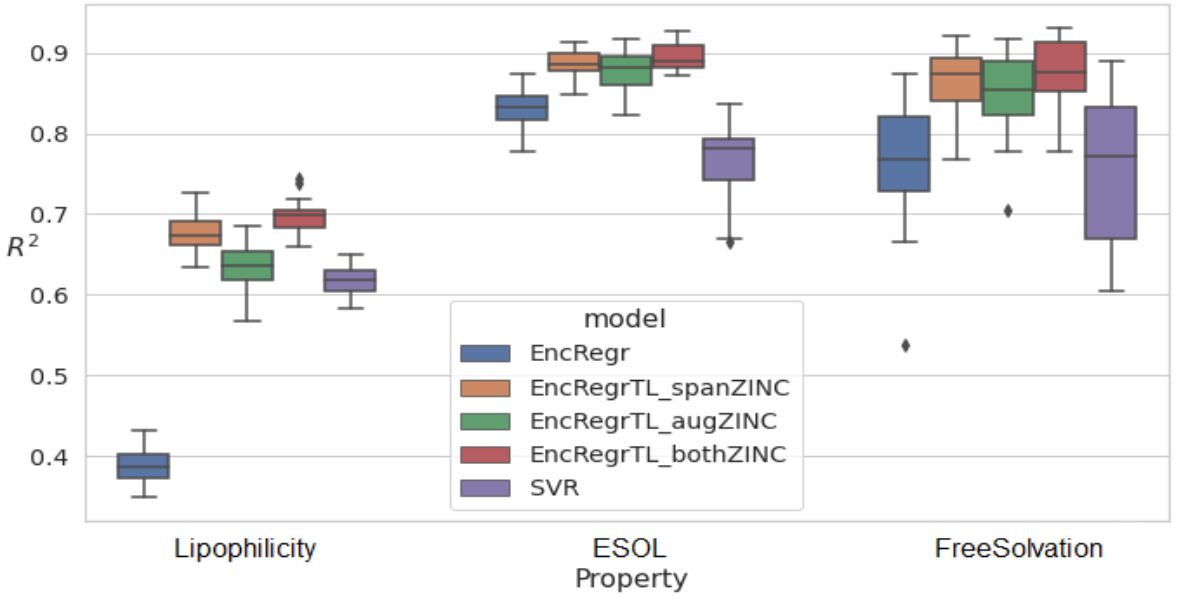
- * Tetko, Igor V., et al. "State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis." *Nature communications* 11.1 (2020): 1-11.
- ‡ He, Jiazhen, et al. "Molecular optimization by capturing chemist's intuition using deep neural networks." Journal of cheminformatics 13.1 (2021): 1-17.

R. Irwin, S. Dimitriadis, J. He, and E. Bjerrum, "Chemformer: A Pre-Trained Transformer for Computational Chemistry," *ChemRxiv*, 2021, doi: 10.33774/chemrxiv-2021-v2pnn.

Pretraining crucial for property prediction



133 bioactivitiy datasets from ExCAPE database



3 phys-chem properties from MoleculeNet

Pretraining increases performance, but not always better than ”old-school” machine learning models

Taking it to the next level on the Cambridge One cluster – Collaboration with Nvidia

"The MegaMolBART drug discovery model being developed by NVIDIA and AstraZeneca is slated for use in reaction prediction, molecular optimization and de novo molecular generation. It's based on AstraZeneca's MolBART transformer model and is being trained on the ZINC chemical compound database — using NVIDIA's Megatron framework to enable massively scaled-out training on supercomputing infrastructure.

Alternative Formats

DeepSmiles

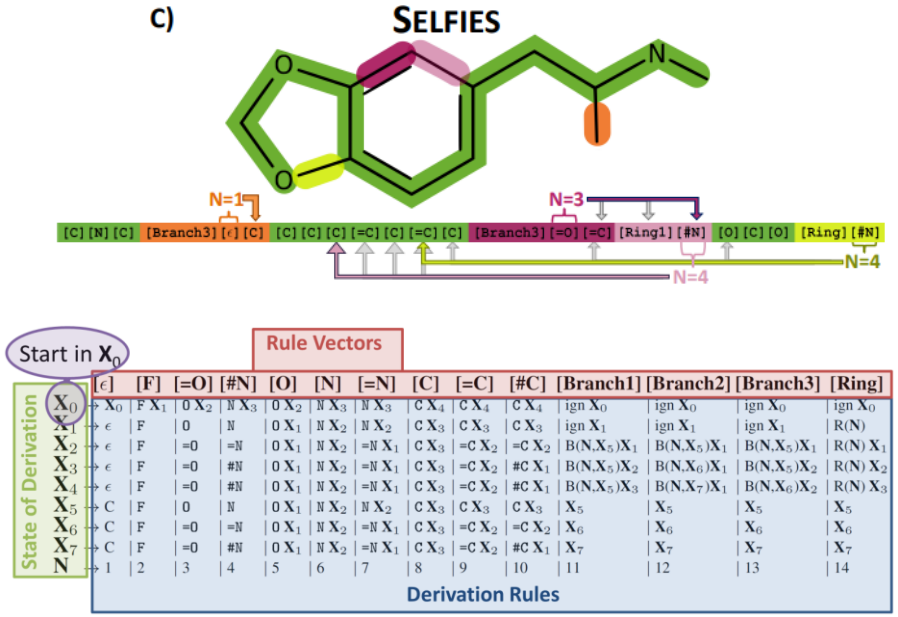
Attempts to make an easier to learn format by avoiding paired symbols (brackets and numbers)

Preprint only suggested but didn't test it

SMILES	DeepSMILES
C1CCCC1	CCCCC5
C1CCCCCCCCC1	CCCCCCCCCC%10
C(O)C	CO)C
C(OF)C	COF))C
C(F)(F)C	CF)F)C
C(=O)Cl	C=O)Cl
C(OC(=O)Cl)I	COC=O)Cl)))I
C1CC(OC)CC1	CCCOC))CC5
C1=C/CCCCCC/1	C=C/CCCCCC/8
C\1=C/CCCCCC1	C=C/CCCCCC/8
B(c1ccccc1)(O)O	Bcccccc6))))))O)O
Cn1cccc-2nccc12	Cncccncccc9-5

Selfies

Designed to always be valid through complex dynamic rule tables



Performance variations, but no clear winner, a solid benchmark and comparison paper is in great need.

[1] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry," 2019, [Online]. Available: http://arxiv.org/abs/1905.13741.

[1] N. O'Boyle and A. Dalke, "DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures," *chemRxiv: 10.26434*, 2018, doi: 10.26434/chemrxiv.7097960.v1.

Conclusion

- SMILES is a useful format for deep learning
- Data augmentation increases performance
- We can use SMILES to generate molecules (read-out)
- Advanced architectures can be designed for specific tasks
- Unsupervised pretraining increases performance and shortens training time

Acknowledgements

Jiazhen He, Postdoc
Rocío Mercado, Postdoc
Josep Arus Pous, Ph.D student, BIGCHEM
Amol Thakkar, Ph.D student, BIGCHEM
Panagiotis-Christos Kotsias, Graduate Scientist,
Ross Irwin, Graduate Scientist
Tobias Rastemo, Master Student
Samuel Genheden, Data Scientist/Software Engineer
Christos Kannas, ML/Cheminformatics Expert
Atanas Patronov, Associate Principal Scientist

Rest of Molecular AI department

External Collaborators:
Prof. Dr. Jean-Louis Reymond · Dept. of Chemistry & Biochemistry, University of Berne
Christian Tyrchan, Director - Computational Chemistry
Boris Sattarov, Informatics Programmer, Science Data Software LLC
Hongming Chen, Professor, Centre of Chemistry and Chemical Biology, Guangzhou, China
Nidhal Selmi, Research Outsourcing Specialist, Hit Discovery

Thank you for your attention! Questions?

Papers of interest

(1) Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *arXiv* **2017**.

(2) Bjerrum, E. J.; Threlfall, R. Molecular Generation with Recurrent Neural Networks (RNNs). *arXiv* **2017**.

(3) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular De-Novo Design through Deep Reinforcement Learning. *J. Cheminform.* **2017**. https://doi.org/10.1186/s13321-017-0235-x.

(4) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **2018**, *4* (1), 120–131. https://doi.org/10.1021/acscentsci.7b00512.

(5) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4* (2), 268–276. https://doi.org/10.1021/acscentsci.7b00572.

(6) Bjerrum, E. J.; Sattarov, B. Improving Chemical Autoencoder Latent Space and Molecular de Novo Generation Diversity with Heteroencoders. *Biomolecules* **2018**, *8* (4), 131.

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

(8) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. Randomized SMILES Strings Improve the Quality of Molecular Generative Models. *J. Cheminform.* **2019**, *11* (1). https://doi.org/10.1186/s13321-019-0393-0.

(9) Kotsias, P.-C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E. J. Direct Steering of de Novo Molecular Generation with Descriptor Conditional Recurrent Neural Networks. *Nat. Mach. Intell.* **2020**, *2* (5), 254–265. https://doi.org/10.1038/s42256-020-0174-5.

(10) Genheden, S.; Thakkar, A.; Chadimova, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. J. AiZynthFinder: A Fast Robust and Flexible Open-Source Software for Retrosynthetic Planning. **2020**. https://doi.org/10.26434/chemrxiv.12465371.v1.

(11) Blaschke, T.; Arús-Pous, J.; Chen, H.; Margreitter, C.; Tyrchan, C.; Engkvist, O.; Papadopoulos, K.; Patronov, A. REINVENT 2.0: An AI Tool for de Novo Drug Design. *J. Chem. Inf. Model.* **2020**. https://doi.org/10.1021/acs.jcim.0c00915.

(12) He, J.; Mattsson, F.; Forsberg, M.; Bjerrum, E. J.; Engkvist, O.; Nittinger, E.; Tyrchan, C.; Czechtizky, W. Transformer Neural Network for Structure Constrained Molecular Optimization. *ChemRxiv* **2021**.

(13) Irwin, R.; Dimitriadis, S.; He, J.; Bjerrum, E. Chemformer: A Pre-Trained Transformer for Computational Chemistry. *ChemRxiv* **2021**. https://doi.org/10.33774/chemrxiv-2021-v2pnn.