NVIDIA: GTC 2019
Generative Molecular DLNN

Ellen Du, Joey Storer, and Abe Stern*

*NVIDIA

The Dow Chemical Company

Our Team

DOW TEAM:

Ellen Du, Joey Storer

Sukrit Mukhopadhyay

Matthew Christianson

Hein Koelman, Ryan Marson

William Edsall, Bart Rijksen
Jonathan Moore

Christopher Roth, Peter Margl

Clark Cummins, Dave Magley

NVIDIA TEAM/Alumni:

Abe Stern, Michelle Gill,
John Ashley, Alex Volkov

Outline

Problem statement

Efforts in generative molecular deep learning methods

Our approach

- Hardware/software
- Tooling
- Data curation
- Model Training and convergence
- Latent space analysis and inference
- Generative capability evaluation

Problem statement

Can a molecular generative deep learning system be trained to deliver new molecular designs relevant to our research needs?

Introduction: Generative Molecular Systems

Challenges:

- Molecular encoding (Canonical SMILES)
- Molecular descriptors (100's)
- Vastness of chemical search space (10^{60})
- Unknown structure/property relationships f(n)
- Promise of the latent space dimensionality (32-bit)
- Limits on data set used for training (ChEMBL, ZINC)
- Organization of target properties within the latent space (AlogP)
- Molecule discovery workflow (post-filtering)

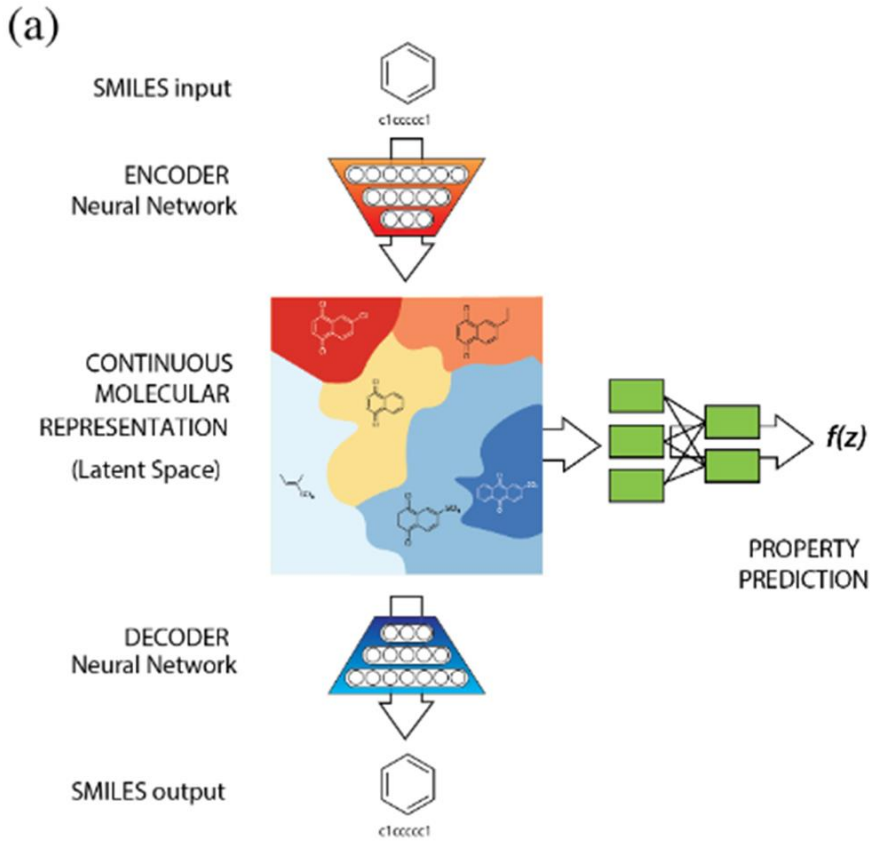
Attraction of Molecular VAE/GANs

Convert discrete molecules to continuous latent representations

- Molecules are discrete entities
- Subtle molecular transformations have large differences in performance

Undocumented benefit to using negative data in ml/dl

- Availability of a molecular structure axis in DL that is not generally available to ML
- Tendency in science to “move on” relative to negative or poor results



Gomez-Bombarelli, et al., ACS Cent. Sci., **2018**, 4 (2), pp 268–276

General intro on methods: VAEs

Generally there are numerous methods appearing in the open literature:

- Chemical VAE
- Grammar VAE
- Junction Tree
- ATNC RL
- FC-NN (NVIDIA-Dow)

The best way to go is not entirely clear.

Junction Tree – may be best because of the more natural graph representation – but it may constrain diversity

FC-NN is potentially more efficient.

Inferencing Comparison to Literature

Method	Reconstruction	Validity
	Knowns	Inferenced(unknown)
Chem-VAE	44 %	1 %[lit.]
Dow-Chem-VAE	94 %	10 %
Grammar-VAE	54 %	7 % [lit.]
SD-VAE	76 %	44 % [lit.]
Graph-VAE	-	14 % [lit.]
JT-VAE	77 %	100 % [lit.]
Dow-FC-NN	90 %	-- %
ATNC-RL	-	71 % [lit.]

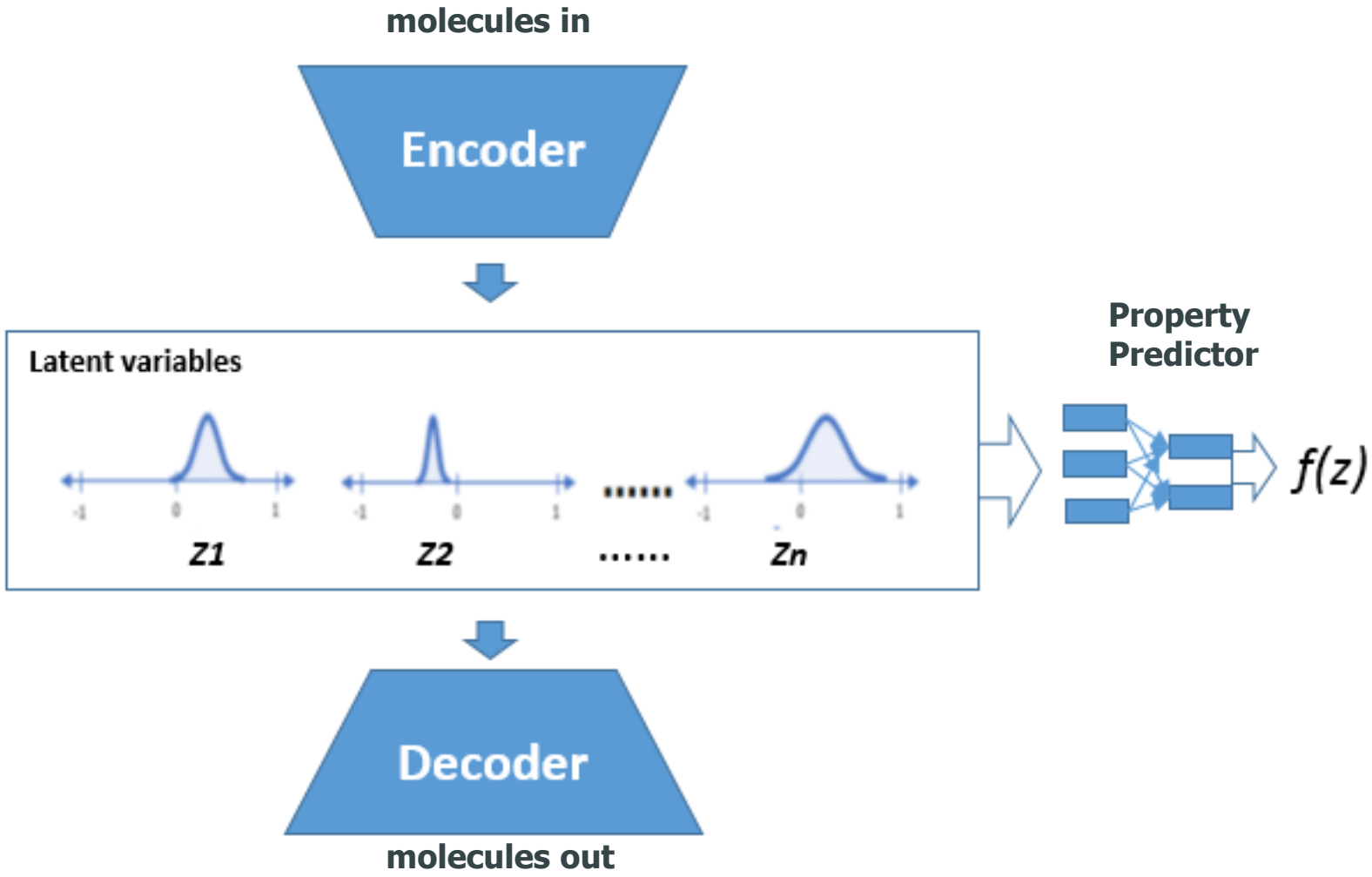
Models and Training Details

Model Details: Architectures Explored

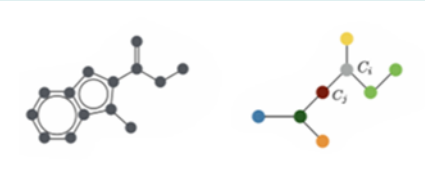
3 Variational AutoEncoders

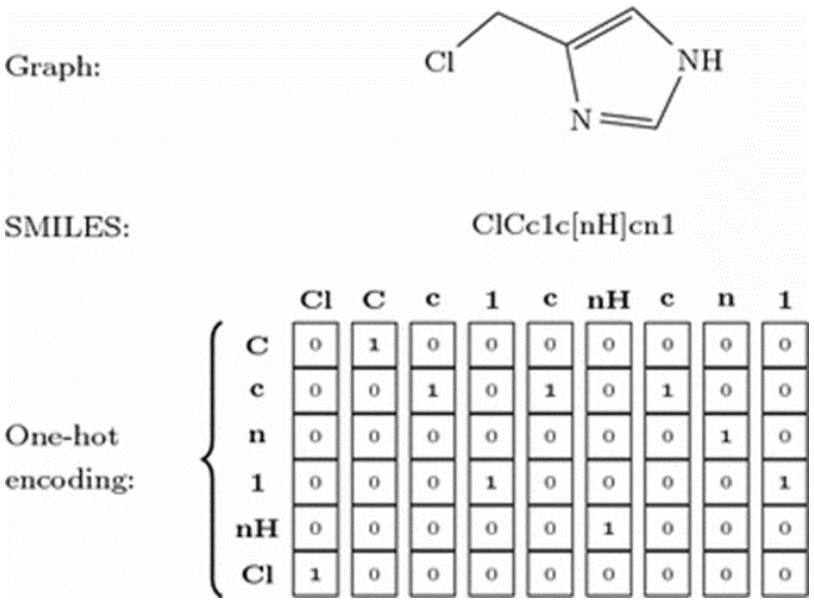
- chemVAE
  Gomez-Bombarelli, et al, 2018 (Harvard)
- Junction Tree VAE
  Jin et al, 2018 (MIT)
- Fully convolutional VAE
  (NVIDIA-Dow)

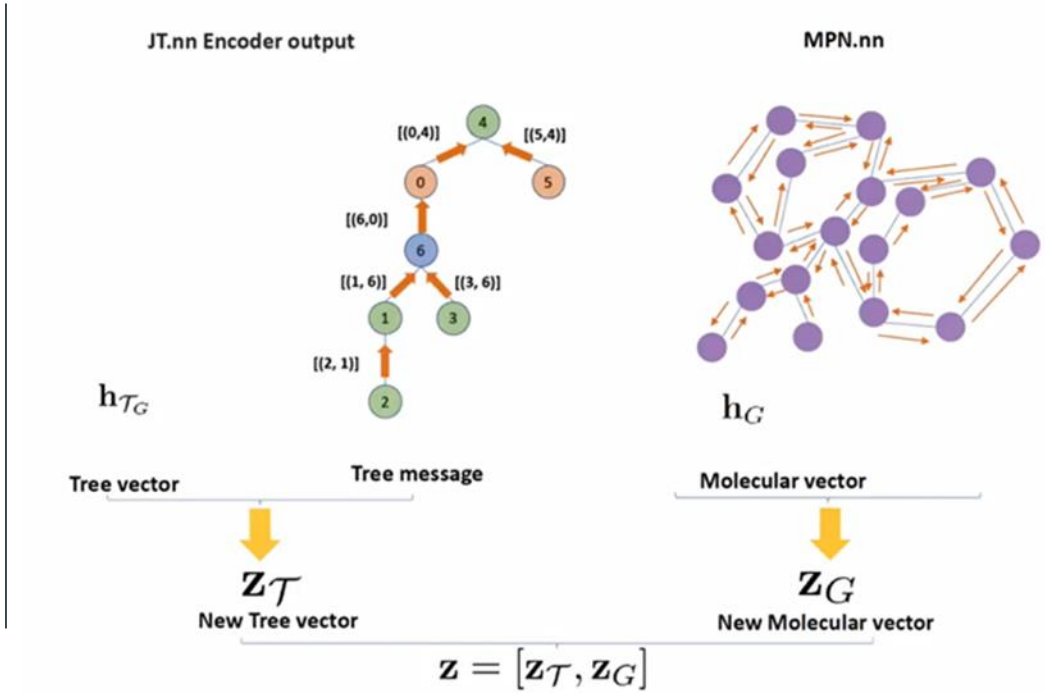
Similar in setup
Different in details



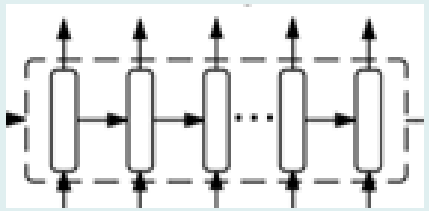
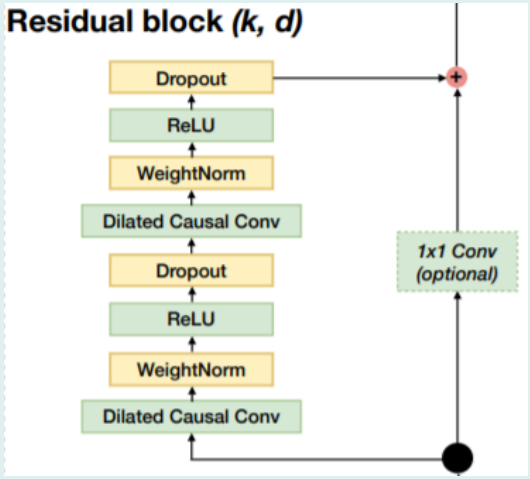
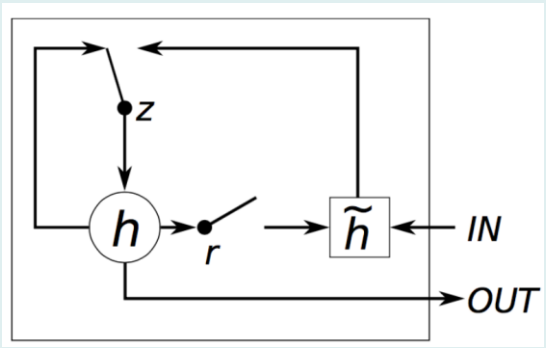
Model Details: Differences In Inputs

	chemVAE	fcVAE	jtVAE
Input	Smiles ClCc1c[nH]cn1	Smiles ClCc1c[nH]cn1	Molecular Graph 



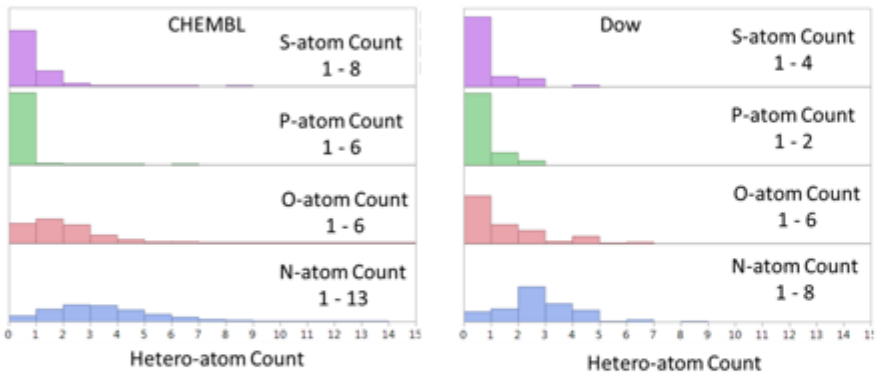


Model Details: Differences In Sequence Modeling

	chemVAE	fcnVAE	jtnn
Layer type used for sequence modeling	Teachers forcing  *Lamb et al, 2016*	Residual block  *Bai et al, 2018*	Gated Recurrent Unit  *Cho et al, 2014*

Model Training Details: Data Compilation

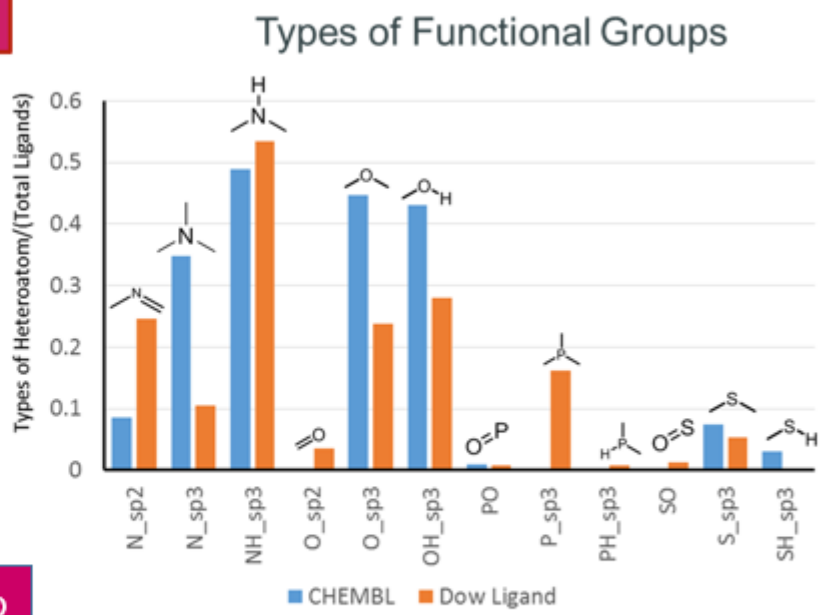
ChEMBL



CHEMBL Database (1.4×10^6)

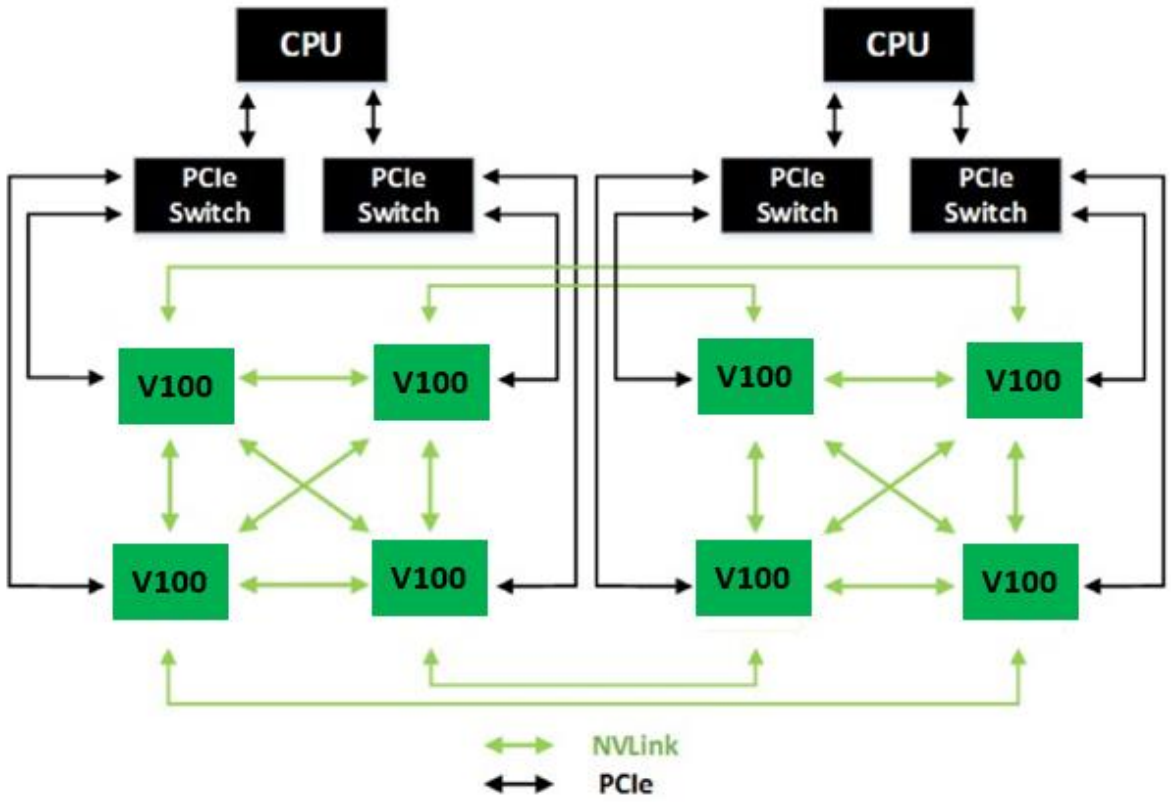
Screening using Designed Metrics

Molecules Similar to Dow Set (118,000)



Model Training Details: Hardware

NVIDIA DGX-1



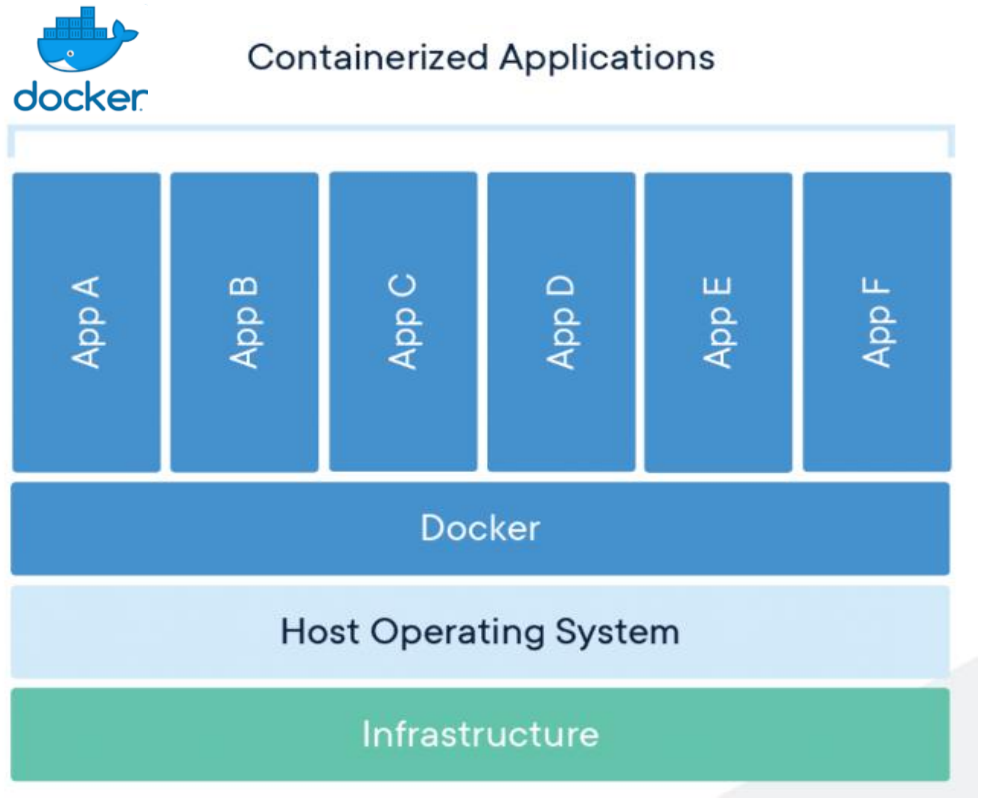
NVIDIA

NVIDIA DGX-1 Specifications	
CPUs	2x Intel Xeon E5-2698 v3 (16 core, Haswell-EP)
GPUs	8x NVIDIA Tesla P100 (3584 CUDA Cores)
System Memory	512GB DDR4-2133 (LRDIMM)
GPU Memory	128GB HBM2 (8x 16GB)
Storage	4x Samsung PM863 1.92TB SSDs
Networking	4x Infiniband EDR 2x 10GigE
Power	3200W
Size	3U Rackmount
GPU Throughput	FP16: 170 TFLOPs FP32: 85 TFLOPs FP64: 42.5 TFLOPs

Model Training Details: Software Environment

Container: Docker container

Standard
Lightweight
Secure

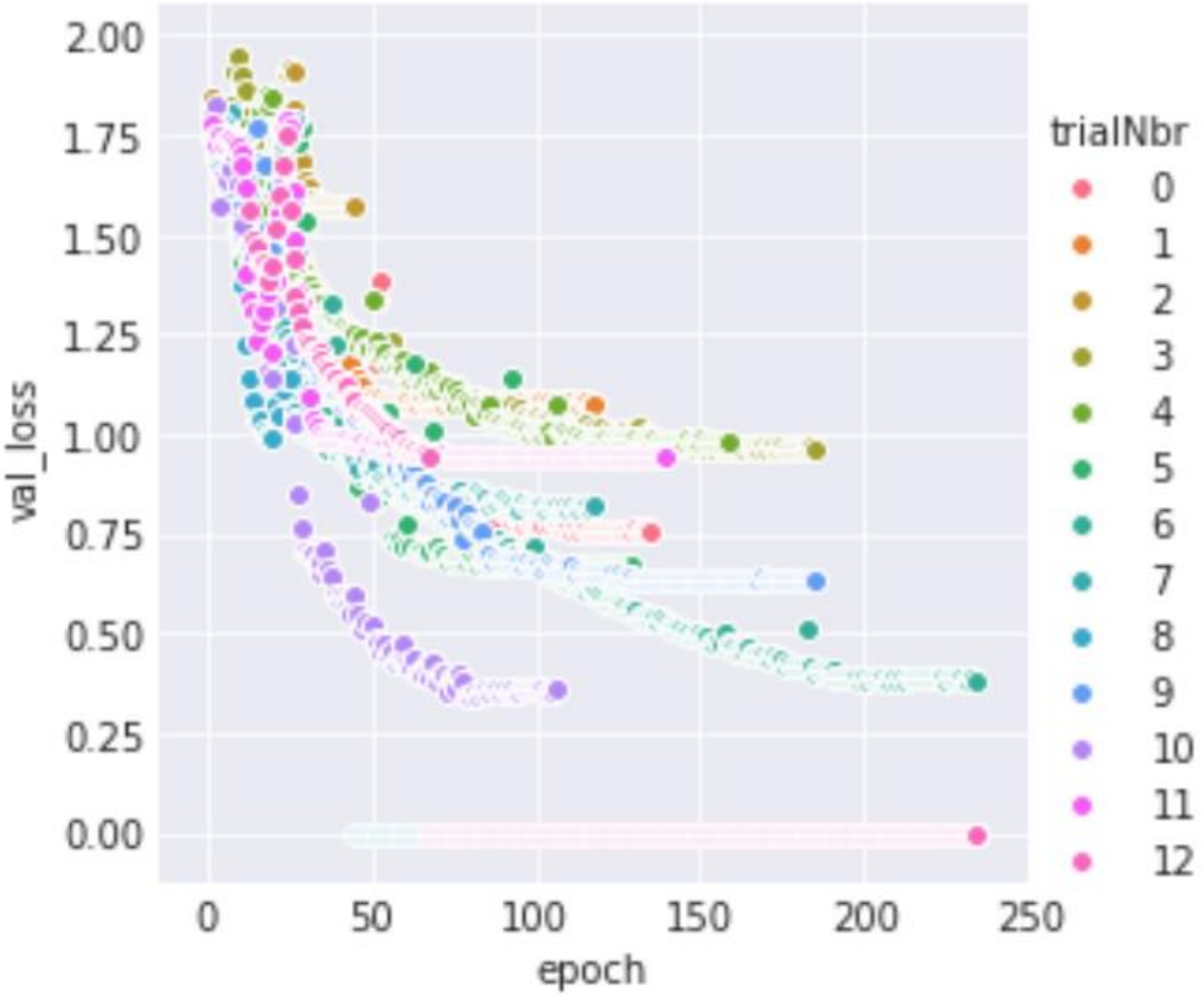


Packages

Chemistry: RDKit, DeepChem
Data Processing: Numpy, Pandas, Rapids
ML/DL: SciKitLearn, Keras, Tensorflow, Pytorch, XGBoost
Tuning/Scaling Up: Hyperopt, Horovod

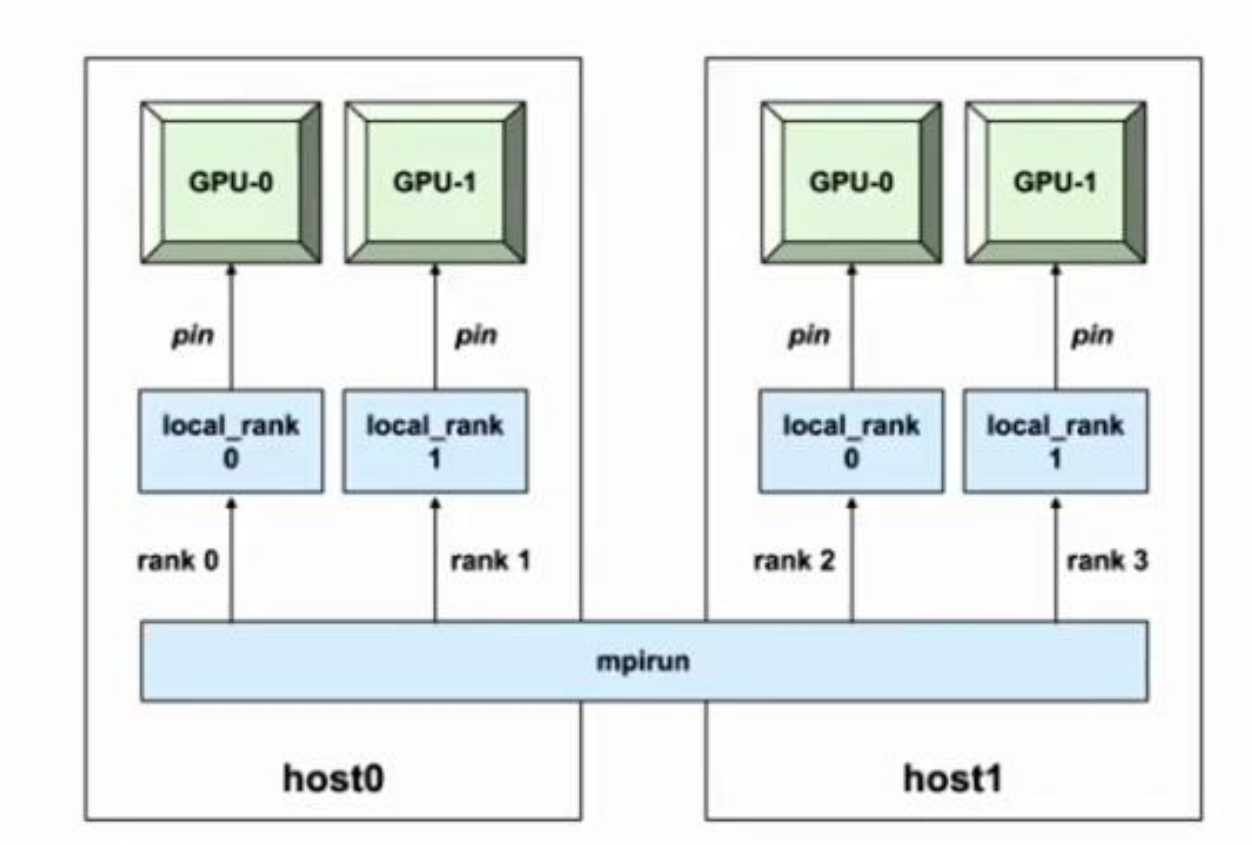
Model Training Details: Hyperparameter Optimization

Hyperopt



Model Training Details: Distributed Model Training

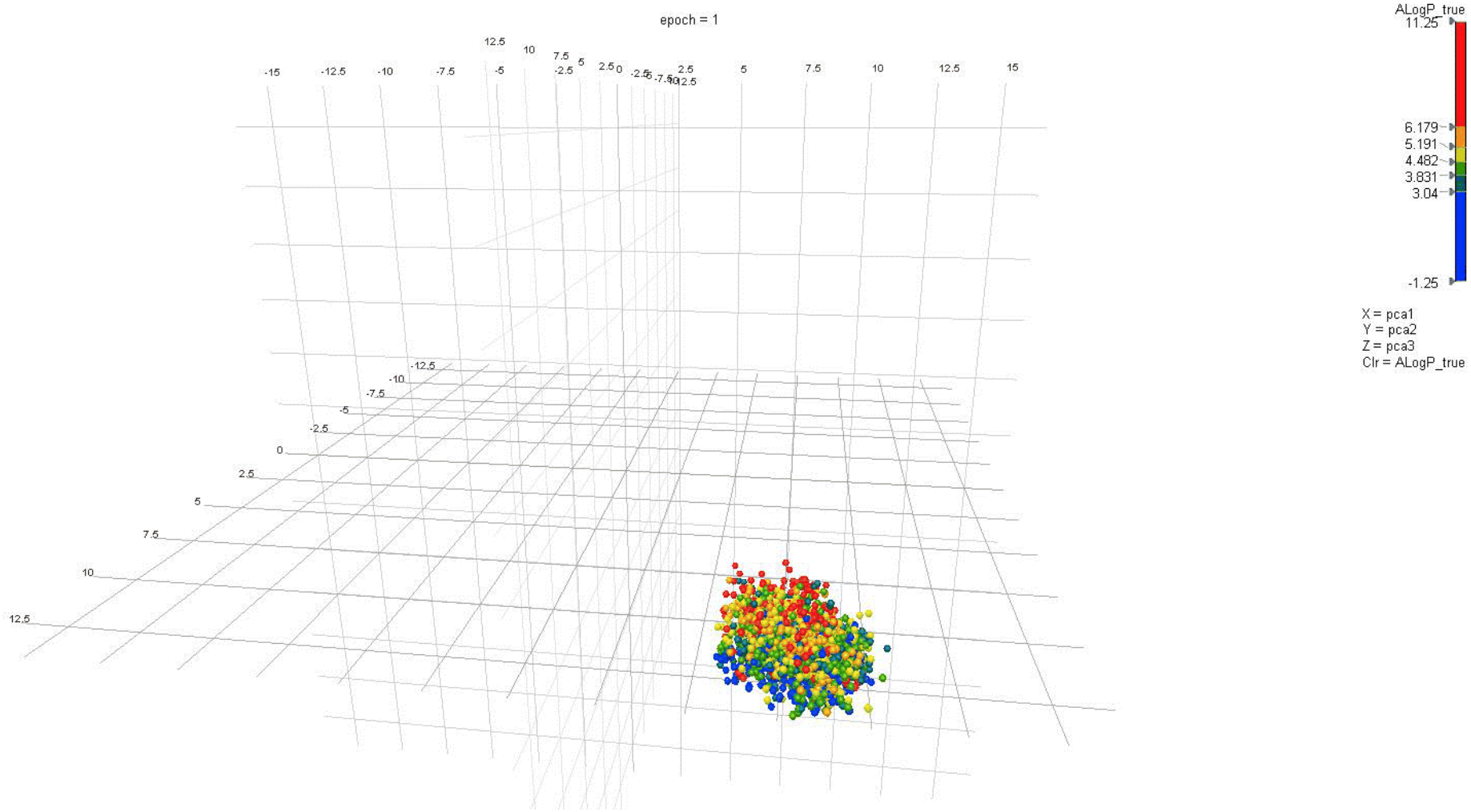
Horovod



HOROVOD

- Data Parallelism
- Network Optimal
- User friendly

Model Training Details: Latent Space Organization

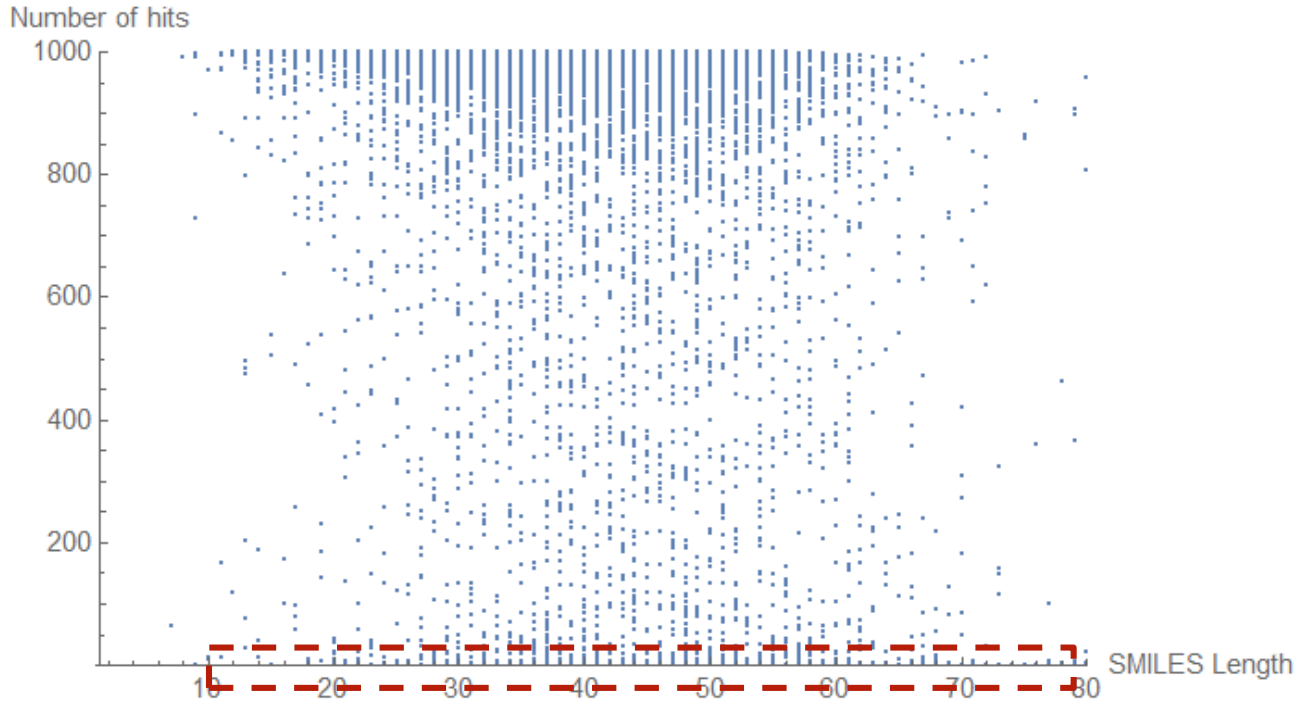


Generative Capability Evaluation

Hit Rate Analysis (> 0 hits/1000 attempts)

ChEMBL TEST = 11800 test molecules inferenced (1000 attempts)

Model	Hit Rate
C-VAE	
55550	94.4 %
JT-NN	
7	100 %
FC-NN	
14587*	*94 %*

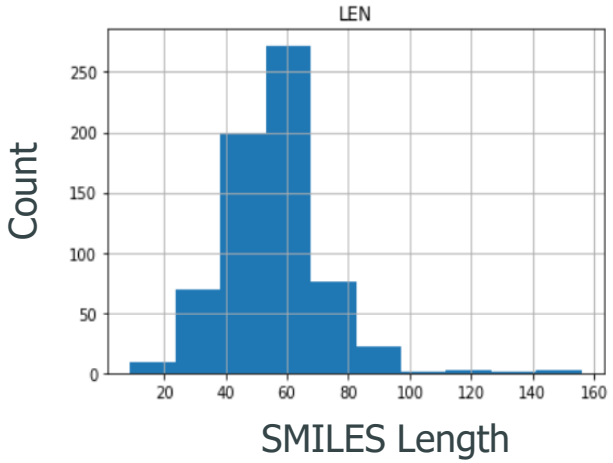


C-VAE
655 TEST molecules not decoding

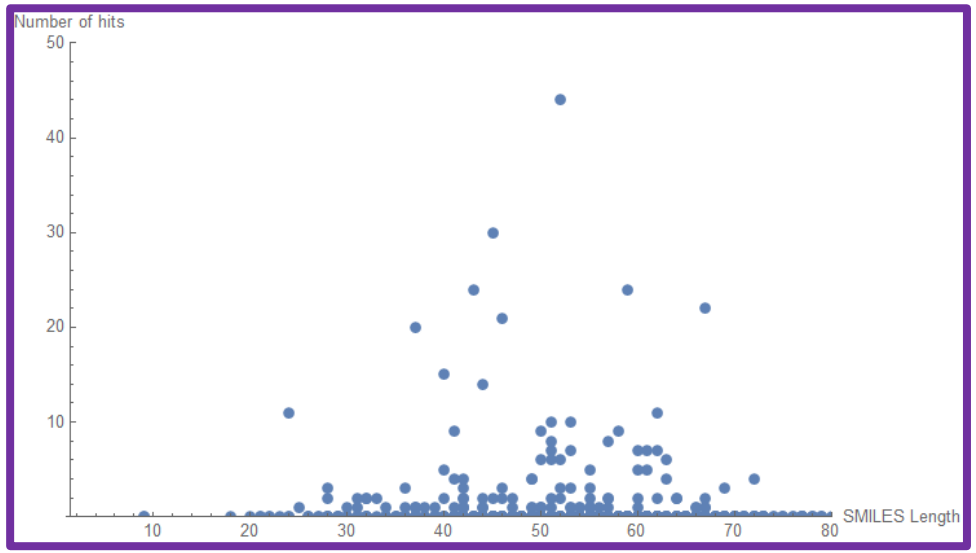
VAE Hit rate: Molecules that never decoded

Analysis of molecules from ChEMBL-TEST (655) that did not decode with 1000 attempts:

1. SMILES string length distribution for the non-decoding molecules
2. Inference study **increased to 10,000** attempts/molecule
   a. 549/655 still never decoded
   b. 16 % successful decoded at least once on 10,000 additional attempts
   c. One molecule decoded an additional 44 times



Distribution is not remarkable compared to ChEMBL

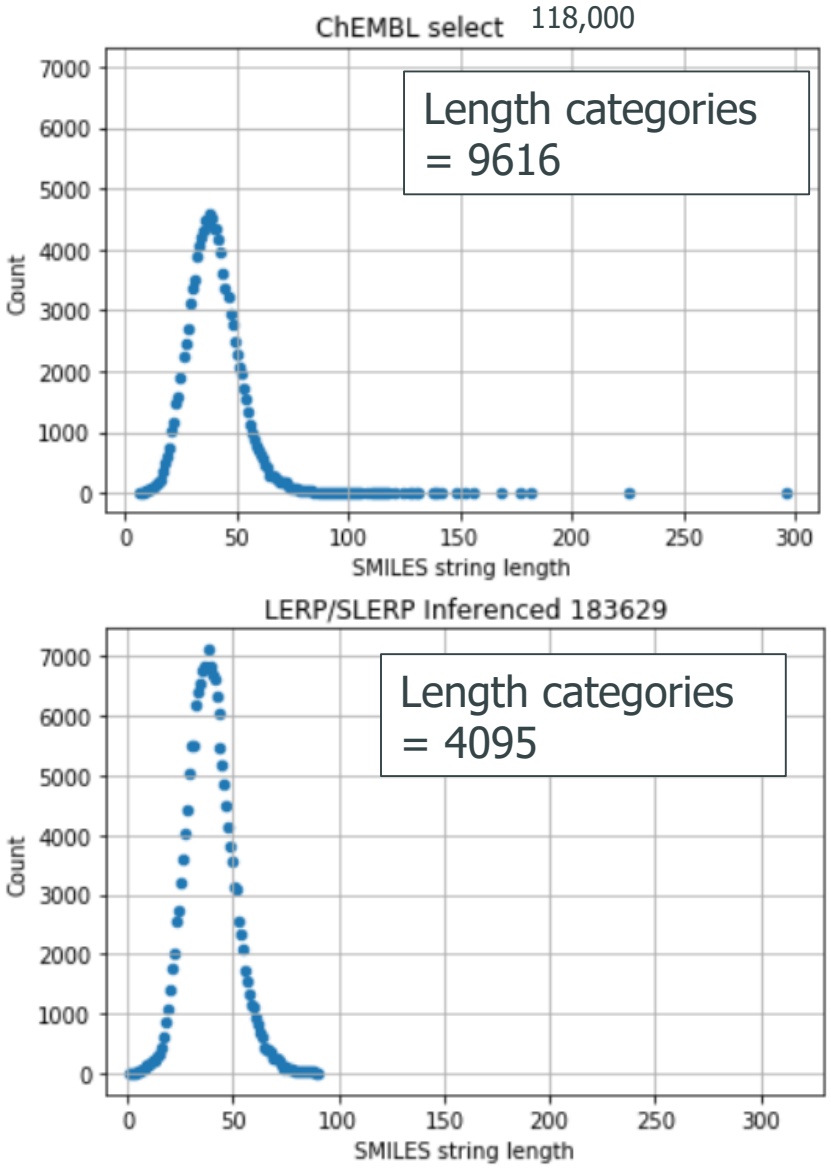


Distribution of SMILES string lengths

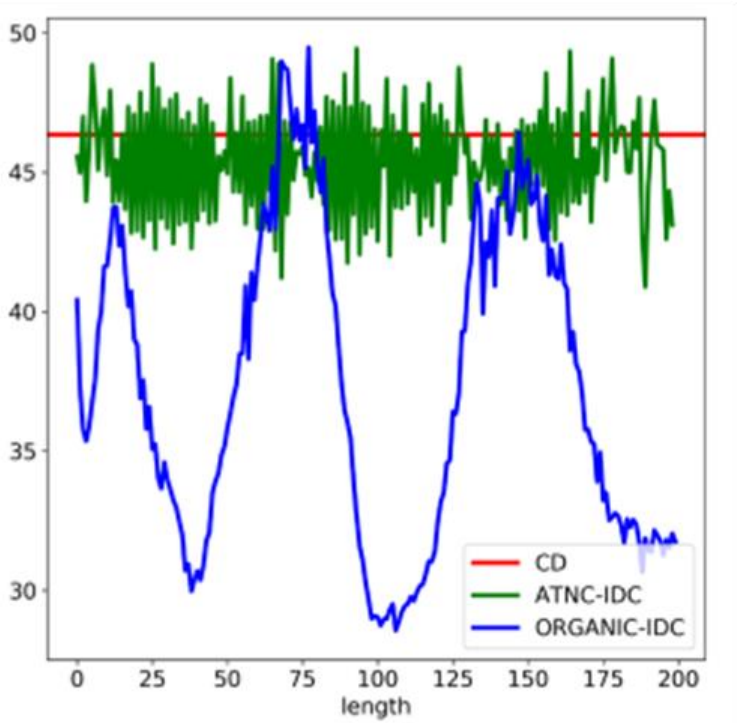
C-VAE
55550

DL
INPUT

DL
OUTPUT



Distribution of SMILES string length w.r.t model type



E. Putin, et al.,
Mol. Pharmaceutics 2018, 15, 4386-4397

Distance calculation and performance

GPU enabled–distance matrix calculation:

1. Characterizing latent space
2. Support inferencing
   a. Nearest neighbor analysis
   b. Gaussian process support

Rough method comparison:

(30,000 molecules, 900 x 10^6 distances)

Python (Simple, non-vectorized)

5 x 10^5 (DGX-1)

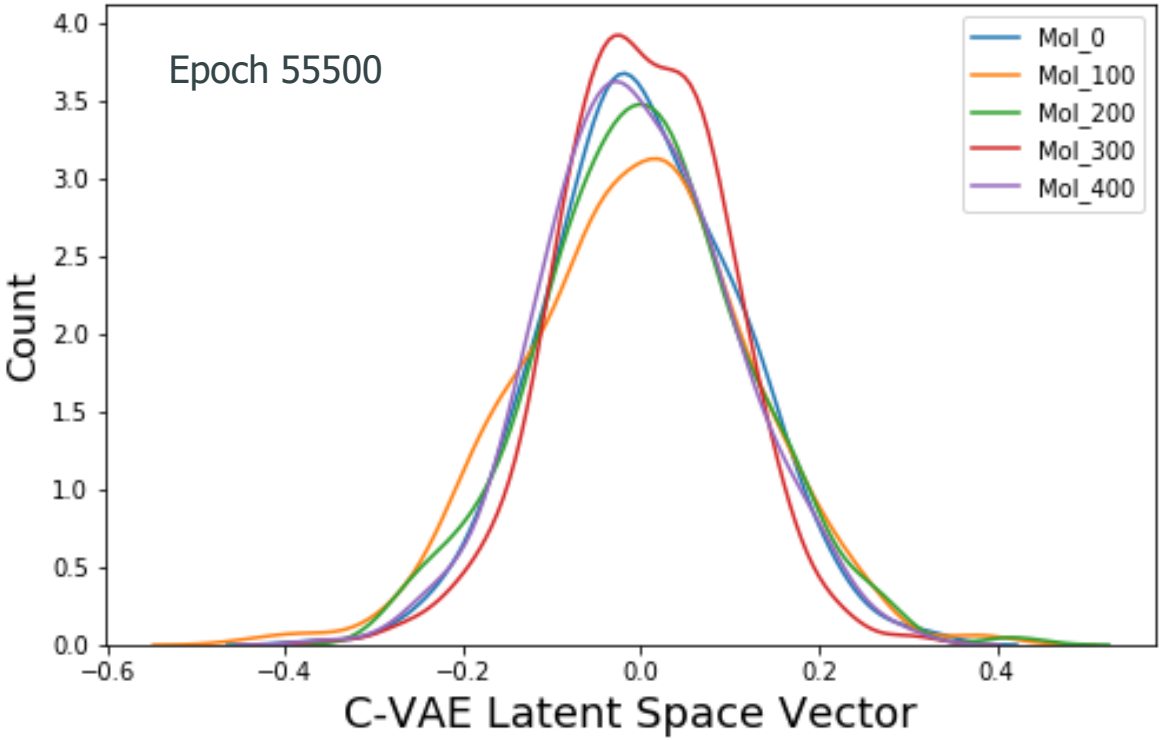
Scipy.spatial.distance.euclidean

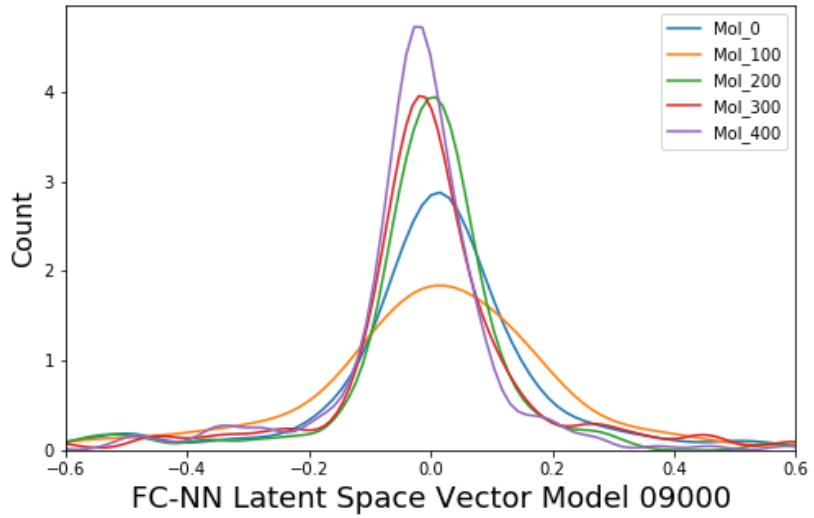
10^4 (DGX-1)

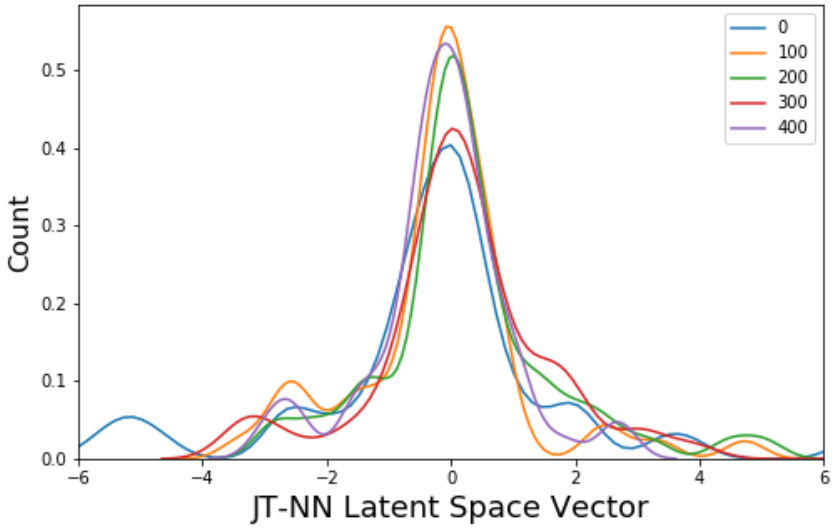
Numba/CUDA

1 (DGX-1)

Latent Space Vectors (Kernel Density Est)
C-VAE, JT-NN, FC-NN







How far apart are the molecules in the Latent Space?

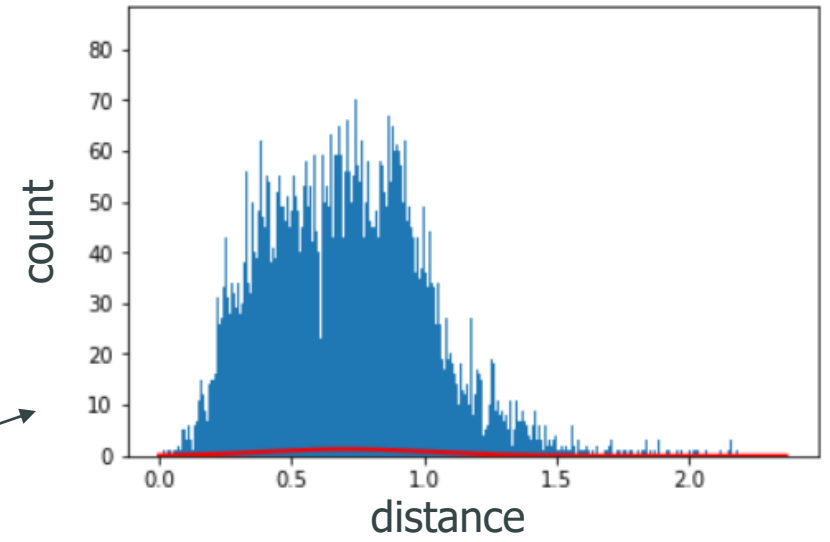
ChEMBL (118,000 molecules)

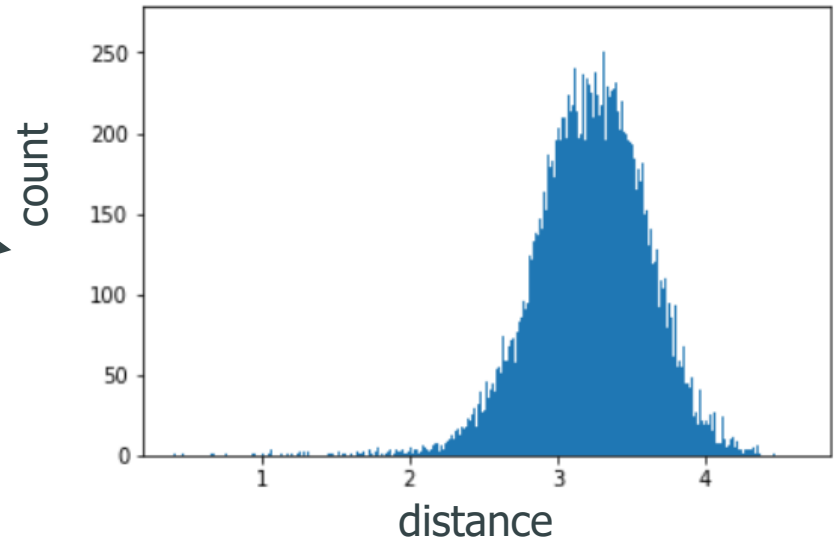
Select 1000 molecules

Calc. Dist. Matrix & plot

Early epoch 2500

Epoch 55500





Mean = 3.2
Std. dev. = 0.38
Max = 4.8
Min = 0.3

Interpolation from the Latent Space

Linear interpolation

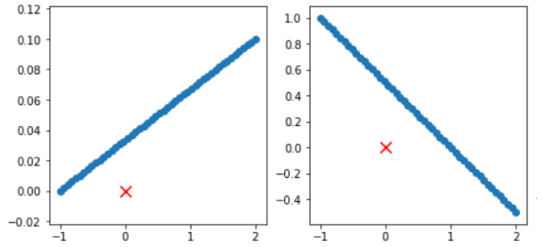
- Stepping through training set and linearly interpolating between endpoints chosen from the training set

Spherical-linear interpolation

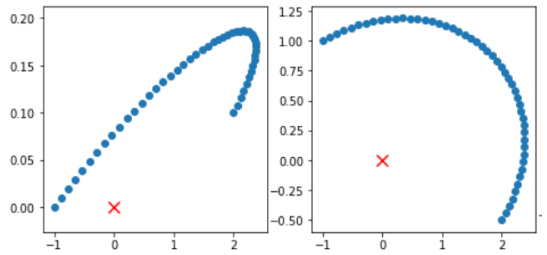
- Stepping through training set and spherical-linearly interpolating between endpoints chosen from the training set

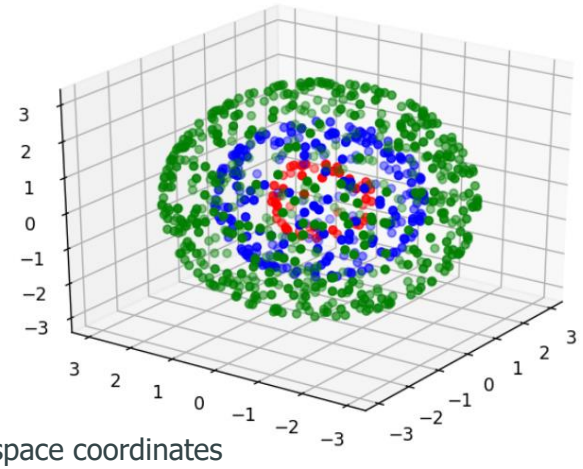
Hyperspheres

- Utilizing the distance matrix to selec point for expanding hyperspheres

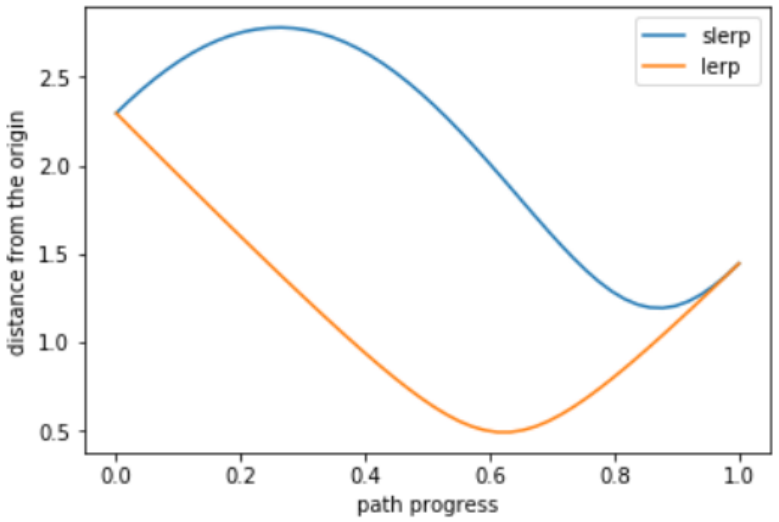


Latent space coordinates





Latent space coordinates

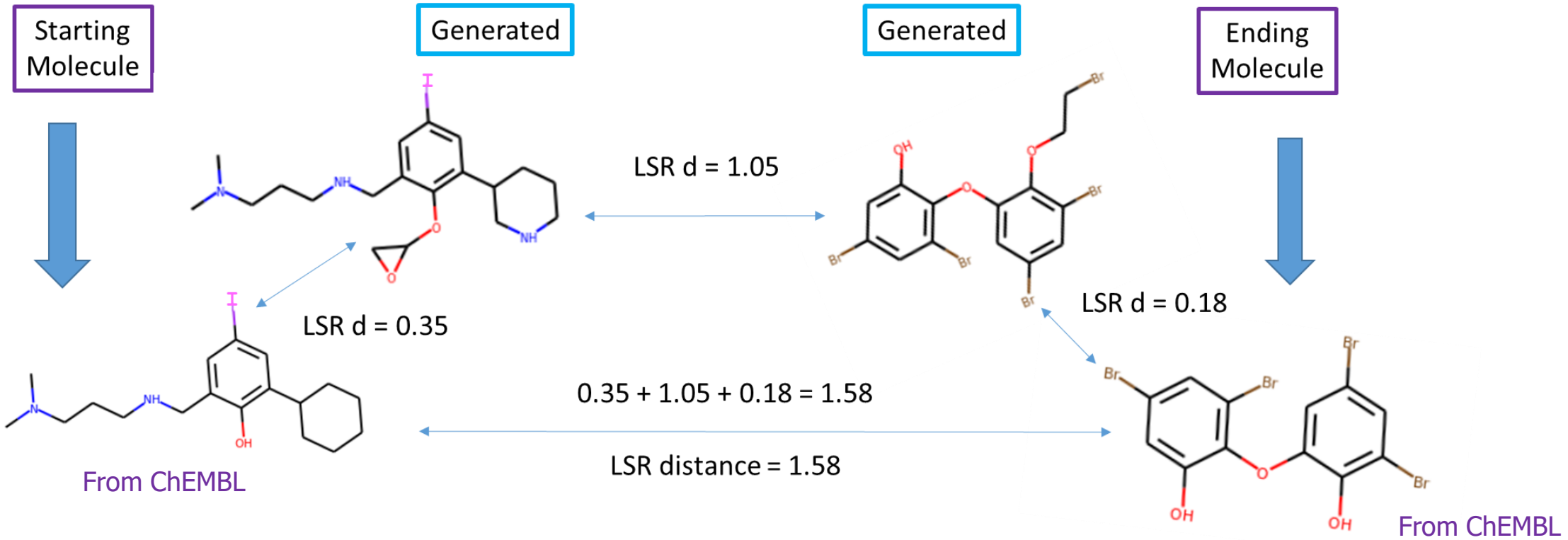


Comment on JT-NN BO search & expand approach

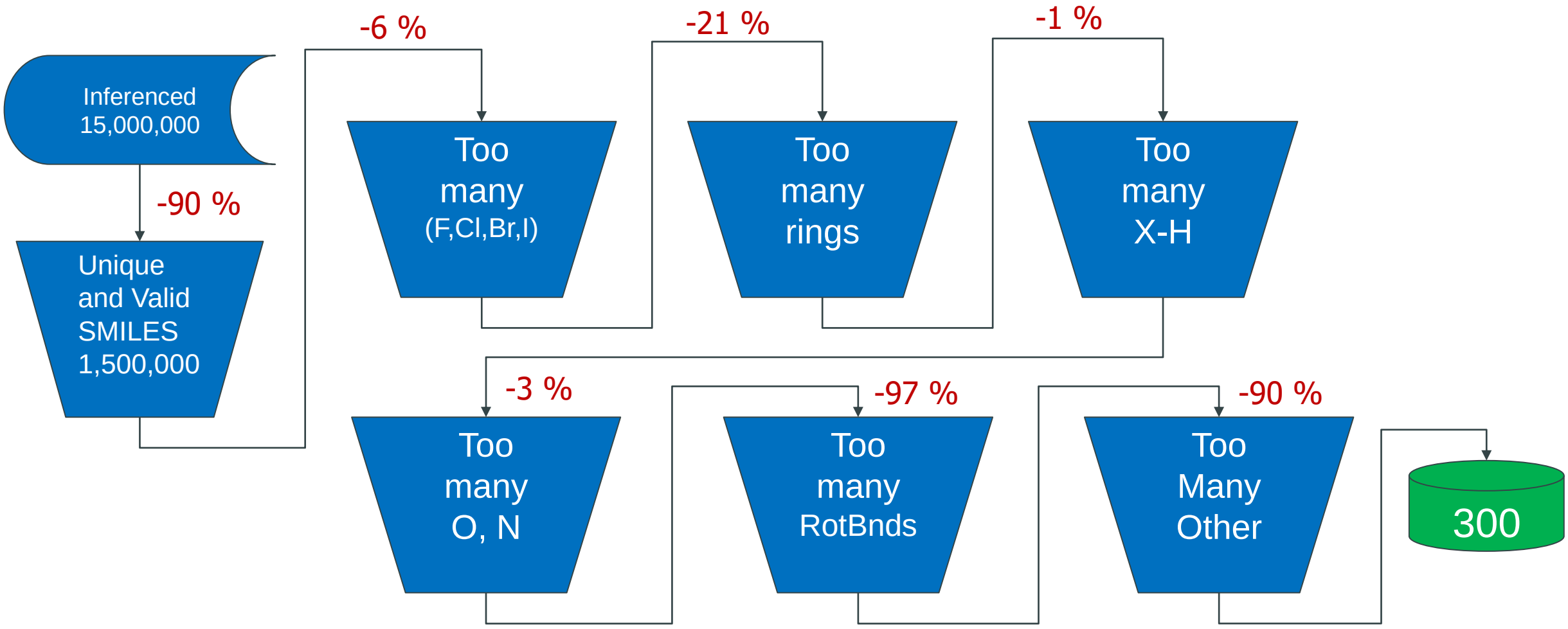
Molecular Interpolation in a Continuous Design Space

LERP/SLERP

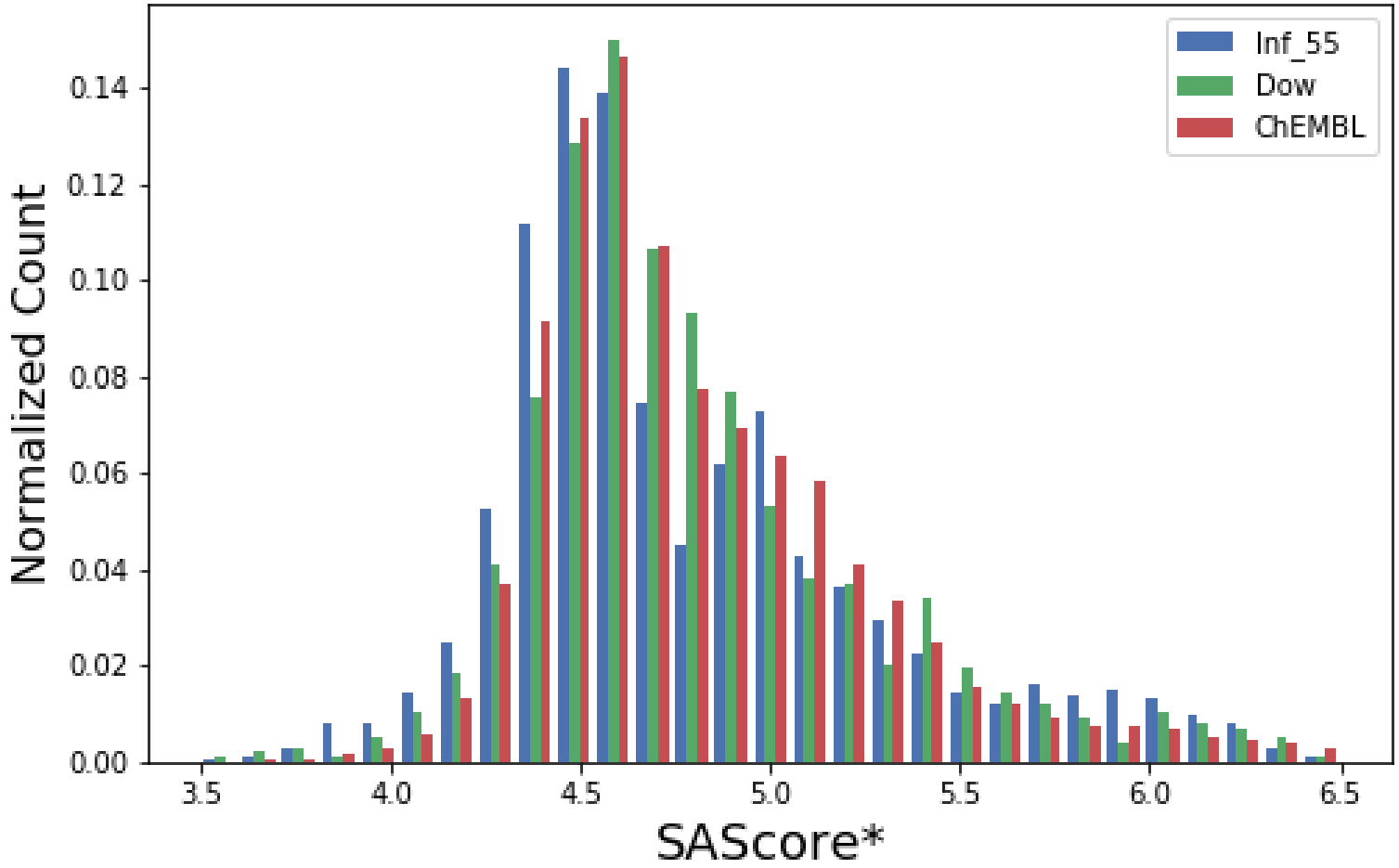
Algorithm only chose points across the whole of the training set (118,000 molecules) and then interpolated between points in ranges to ensure that, at a minimum, each molecule became an end-point for interpolation



Inferencing followed by molecular filtering



Synthetic Accessibility Score



The SAScore across:

INPUT: ChEMBL (118,000)
OUTPUT: Inferenced_e55500
TEST: Dow

- Ertl, P.; Schuffenhauer, A., *J. Cheminf.* **2009**, 1:8
- Ertl, P.; Landrum, G. https://github.com/rdkit/rdkit/tree/master/Contrib/SA_Score

Conclusions

C-VAE

Chem-VAE modeled after Bombarelli works better than reported and delivers good molecules. The time/epoch is high and the number of epochs needed is ~ 50,000.

JT-NN

Junction Tree converges faster, is a more natural representation of molecules, and delivers good molecules.

FC-NN

Fully Convolutional works well, converges faster than C-VAE, and delivers good molecules.

END