

# **Automatic chemical design using a data-driven continuous representation of molecules**

**Paper Presentation: Team Live and Learn**

# Problems

- **PROBLEM STATEMENT:** Overcome challenges in the problem of drug and material design
  - Since making and testing new compounds is costly and time consuming, need to **find a set of promising candidates**
  - Need to **propose novel molecules whose measurable properties optimize an objective function**
  - Optimization in the molecular space for drug design is extremely challenging, because the space is:
    - Large (only  $10^8$  substances out of  $10^{23}$  -  $10^{60}$  have been synthesized),
    - Discrete,
    - Unstructured

# Previous strategies

Useful in finding new molecules, **but** faces major challenges.

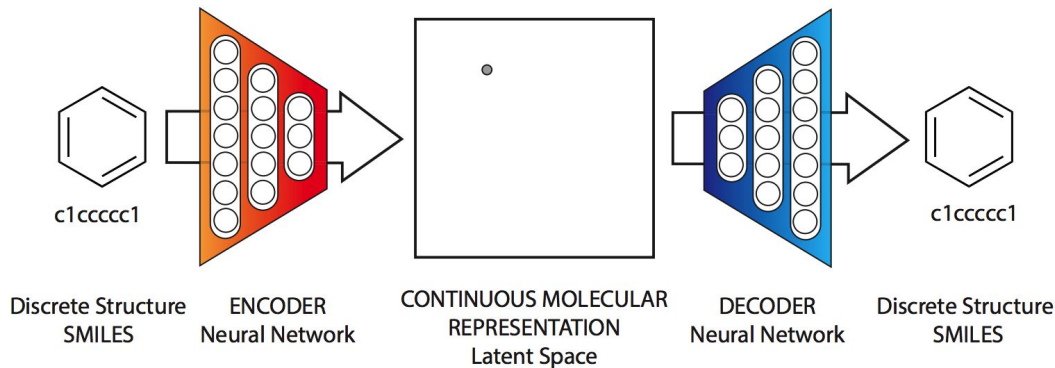
Strategies	Limitations
1. <b><u>Exhaustive search</u></b> through a fixed library (Hachmann et al., 2011)	<ul style="list-style-type: none"><li>● Monolithic</li><li>● Costly to explore fully</li><li>● Requires manual assembly to avoid impractical chemistries</li></ul>
2. <b><u>Discrete local search</u></b> methods such as <ul style="list-style-type: none"><li>● The <b>genetic algorithm</b> (e.g., Virshup et al, 2013, Rupakhti et al. 2015) and</li><li>● <b>Discrete interpolation</b> technique</li></ul>	<ul style="list-style-type: none"><li>● Difficulty in effective search (no guidance for directions)</li></ul>

# Proposed solutions

- Convert discrete representations of molecules to and from a multidimensional **continuous** representation with the following properties:
  - **Data-driven** (less manual work)
  - **Reversible** (interpretable outputs)
  - **Differentiable** (help with search)
  - **Unconstrained** (help with search)

# Proposed solutions

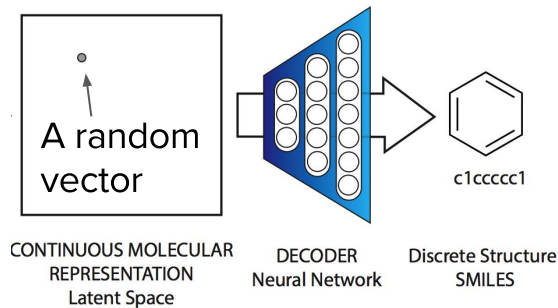
- Convert discrete representations of molecules to and from a multidimensional **continuous** representation with the following properties:
  - **Data-driven** (less manual work) → Deep learning to rescue
  - **Reversible** (interpretable outputs) → Autoencoder
  - **Differentiable** (help with search) → Variational autoencoder
  - **Unconstrained** (help with search)



# Outcomes

- **Ability to automatically generating novel chemical structure**

- Decode random vectors
- Perturb known chemical structures
- Interpolate between molecules

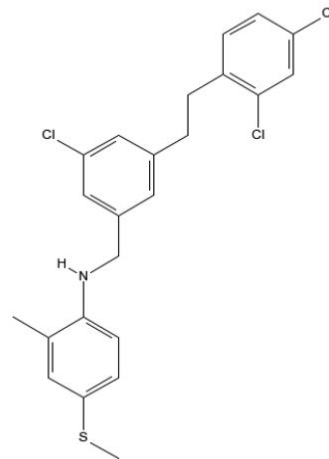


- **Ability to to optimize for desired properties**

- Use stochastic gradient descent and Bayesian optimization

- **Success story:**

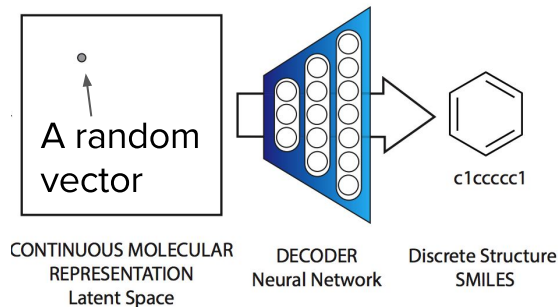
- Found (potentially) better drug-like molecules



# Outcomes

- **Ability to automatically generating novel chemical structure**

- Decode random vectors
- Perturb known chemical structures
- Interpolate between molecules



- **Ability to to optimize for desired properties**

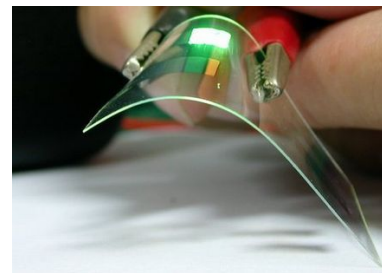
- Use stochastic gradient descent and Bayesian optimization

- **Success story:**

- Found (potentially) better drug-like molecules

- **Failure story:**

- Did not find better organic light-emitting diodes (OLED)



Demonstration of an OLED device

# TECHNICAL APPROACH OVERVIEW



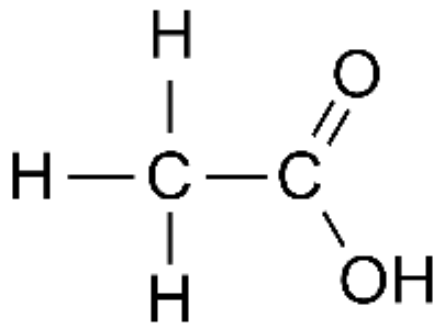
# Molecule Representation

- Want a **discrete** molecular representation for the autoencoder to learn
- **Sequence-to-sequence autoencoders** have been successfully used to represent text
- Thus, use a **text-based** representation of molecules: **SMILES strings**
- **Training data**
  - 250k drug-like commercially available molecules
  - 100k computationally generated OLED molecules

# Molecule Representation: SMILES strings

- **SMILES string representation**
  - 35 different characters
  - Maximum string length of 120 characters
- **Problem:** SMILES string is **fragile** due to character-based representation
  - Network can output invalid strings

**A SMILES string encodes the bonds present in a molecule**



**CC(=O)O**

# Training an autoencoder

- **Architecture**

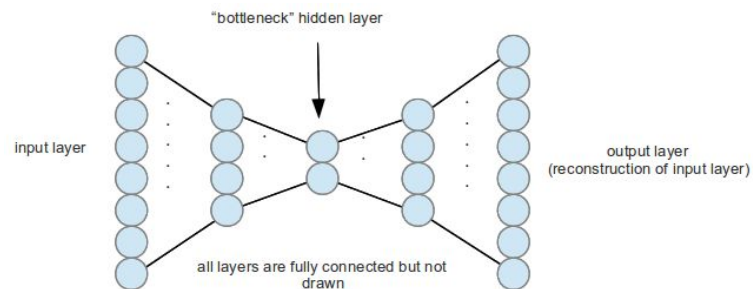
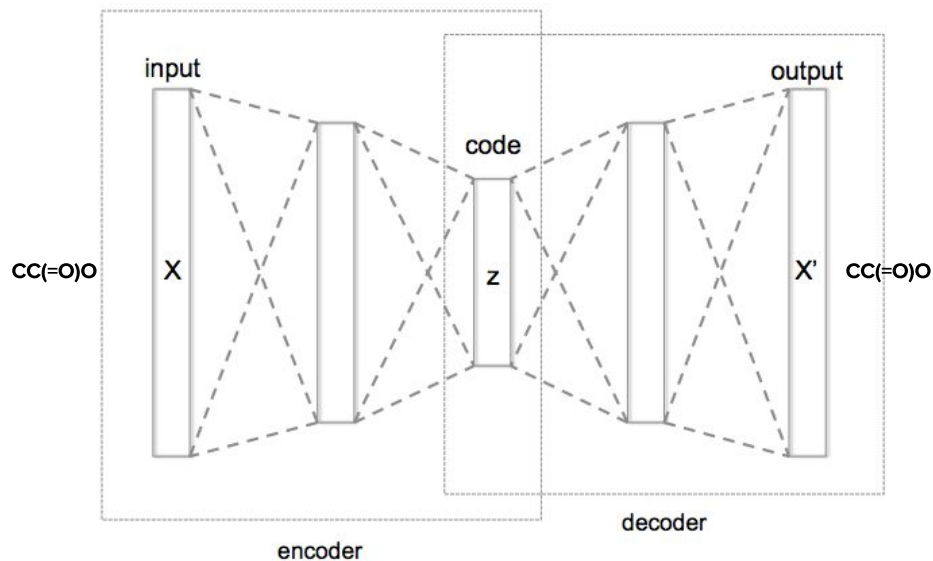
- **Encoder** that converts the the SMILES string into a **fixed-length** vector representation (**latent representation**)
  - Three 1D conv layers, followed by two fully connected layers
- **Decoder** that converts the vector representation back to a string
  - One fully-connected layer, followed by 3 gated recurrent units, and a softmax
  - Last layer of the decoder defines a probability distribution over all possible characters at each position in the SMILES string

# Bayesian Optimization of Molecules

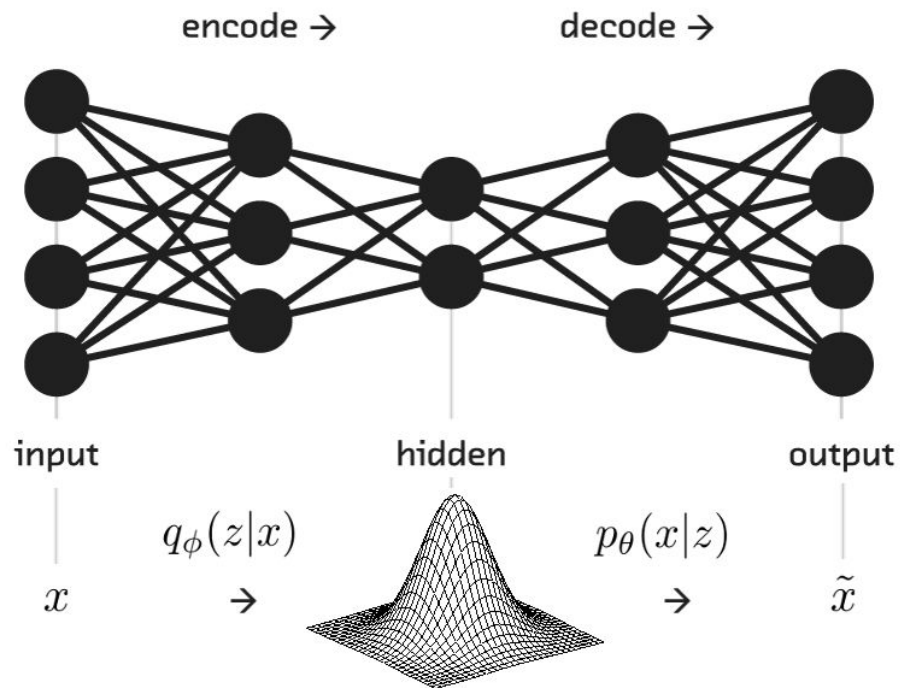
- **Train** a sparse Gaussian process to predict the **cost** (however defined) of each molecule using the molecule's feature vector (**latent representation**)
  - Start with 500 inducing points
- Use **Bayesian optimization** to:
  - **Finetune hyperparameters:**
    - CNN vs. RNN encoder
    - Number of hidden layers
    - Layer size
    - ... and many more!
  - **Maximize other defined cost:** maximize the **expected improvement** heuristic to find molecules that maximize the **EI acquisition function**

# VARIATIONAL AUTOENCODER

## Using variational autoencoders to produce a compact representation

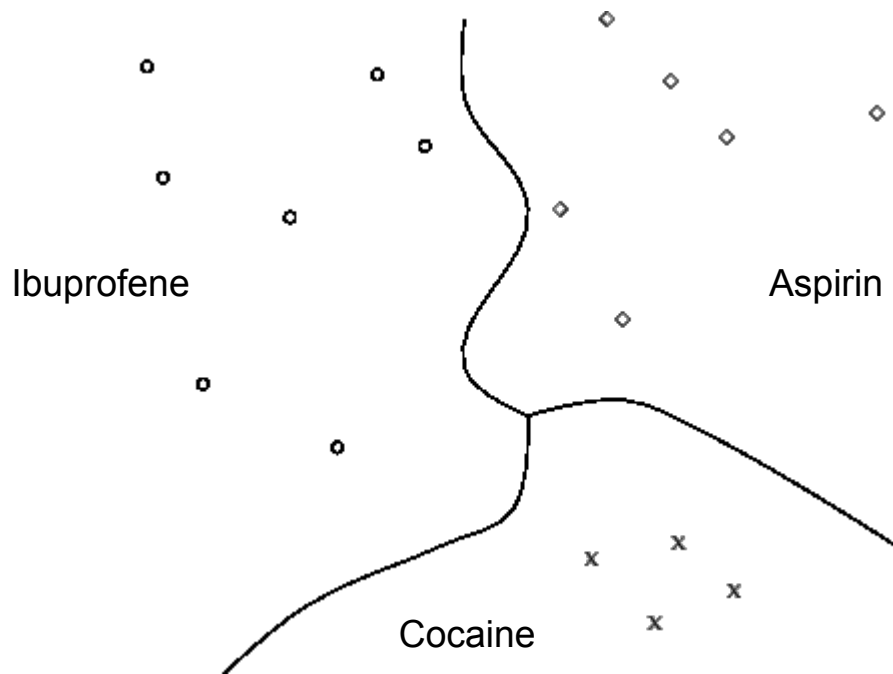


To perform unconstrained optimization in the latent space, most points in the latent space should decode into valid SMILES strings



VAEs generalize autoencoders, adding stochasticity to the encoder, and adding a penalty term encouraging all areas of the latent space to correspond to a valid decoding.

Adding noise to the encoded molecules forces the decoder to learn how to decode a wider variety of latent points. This also encourages the encodings to spread out over the entire latent space to avoid overlap [since two molecules could be stochastically brought close in the latent vector space]

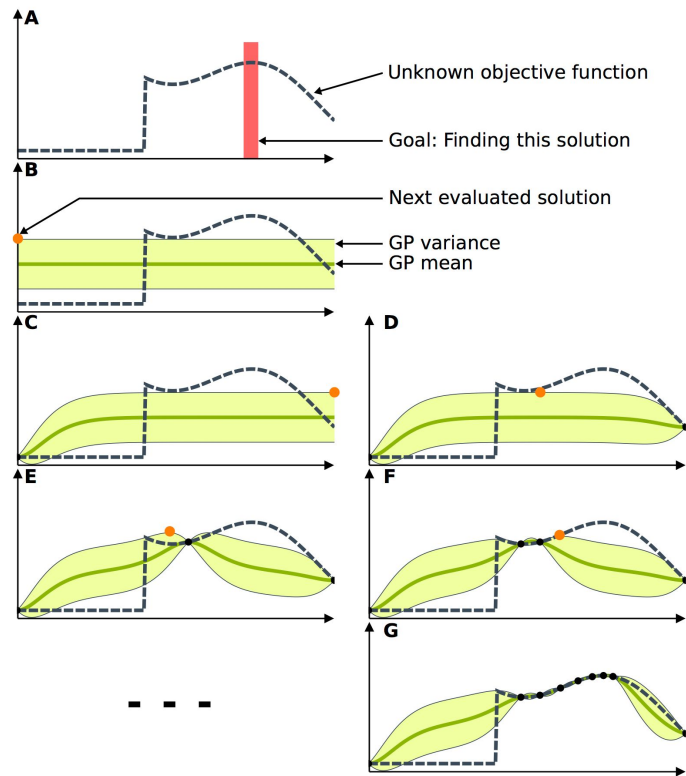




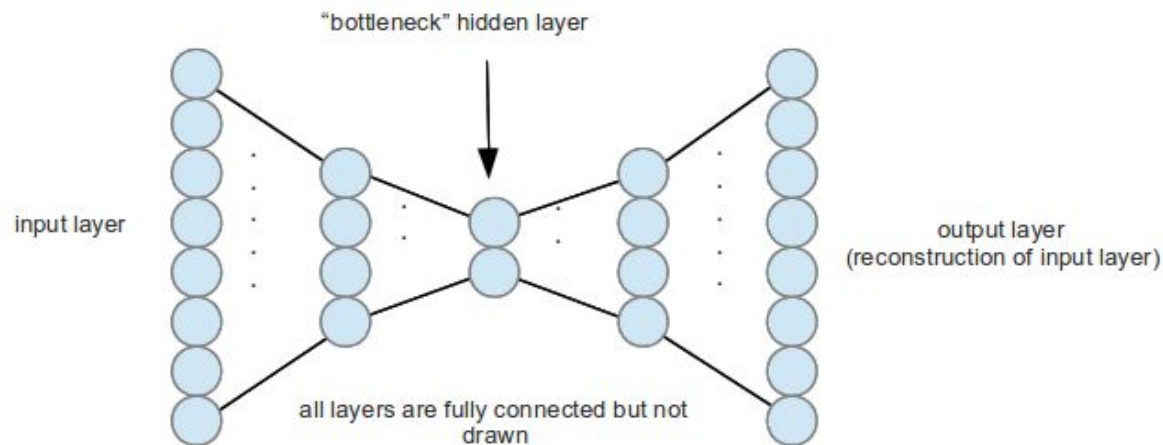
# Bayesian optimization

Optimized:

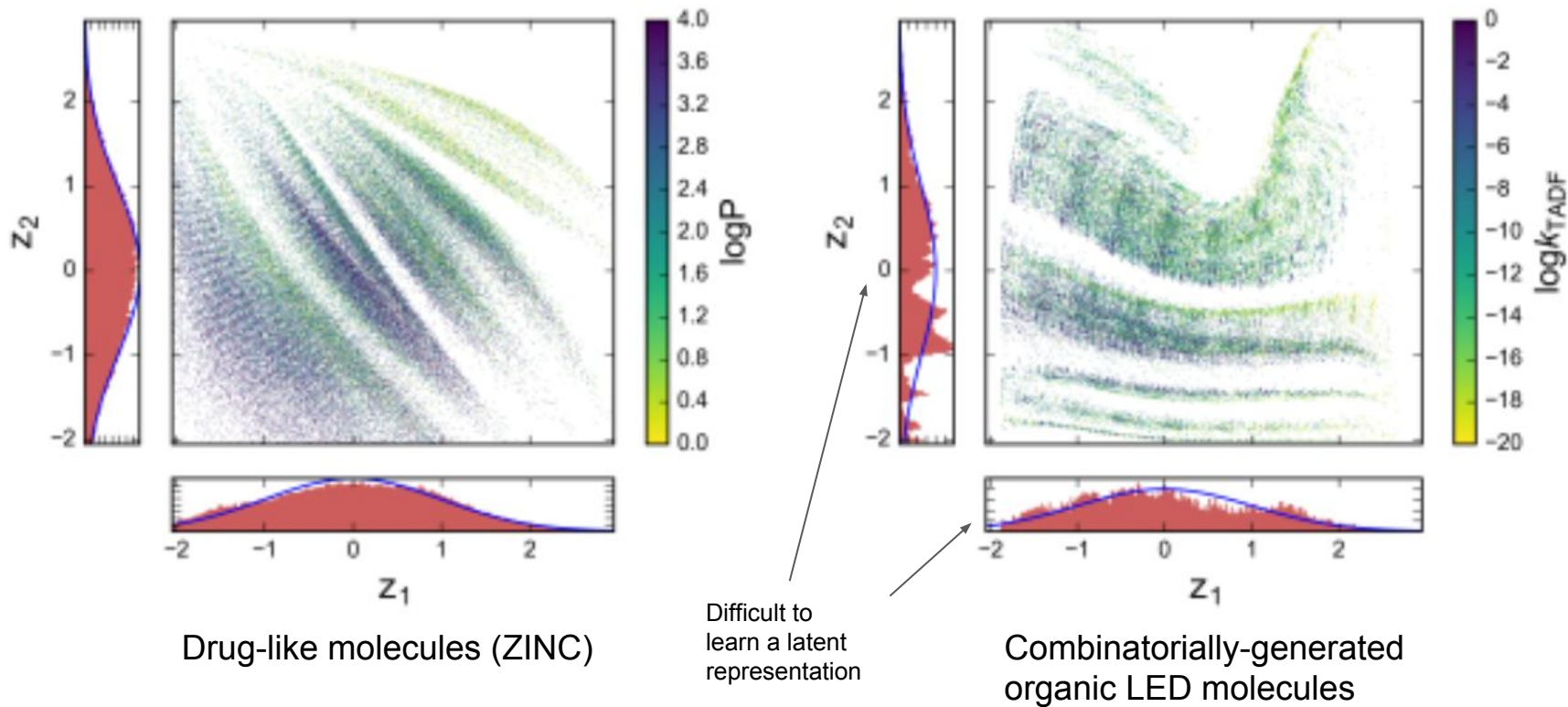
- RNN vs CNN encoder
- number of hidden layers
- layer size
- regularization
- learning rate



Another outer optimization loop to determine how small the latent dimension could be while still producing reasonable reconstruction error



## 2D latent space after training an autoencoder



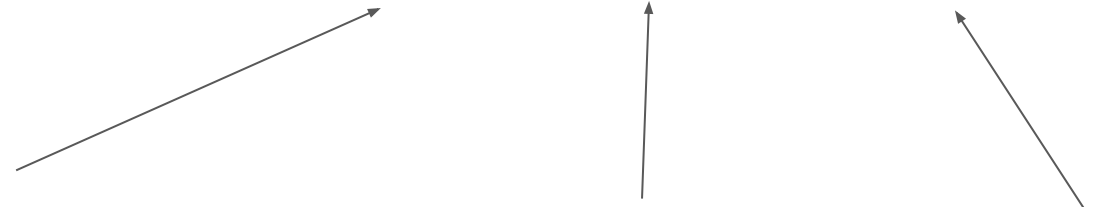
More than half of the 500 latent feature vectors produced a valid SMILES string.

## Bayesian optimization of drug-like molecules

The proposed autoencoder can be used to discover new molecules with desired properties.

Thus our preferred objective is, for a given molecule  $m$ , given by:

$$J(m) = \log P(m) - \text{SA}(m) - \text{ring-penalty}(m),$$



Maximize **water-octanol partition coefficient**, as estimated by RDkit. This is of interest in drug design.

To ensure that the resulting molecules are easy to synthesize, we incorporate the synthetic accessibility score

Initial experiments produced molecules with unrealistically large rings of carbon atoms, so we added a penalty for rings larger than 6

# RESULTS

The autoencoder was trained on:

250k drug-like molecules from the ZINC database.

100k OLED molecules generated computationally.

Molecular family	Autoencoder training loss	Latent dimension	Training set reconstruction %	Test set reconstruction %
drug-like	naïve	56	99.1	98.3
drug-like	variational	292	96.4	95.3
OLED	naïve	56	96.7	91.2
OLED	variational	292	91.4	79.4

Accuracy is defined as the percentage of correct characters in decoded SMILES strings. An autoencoder with a large enough latent dimension could achieve perfect reconstruction, but exploration of the latent space tends to become more difficult as the latent dimension increases.

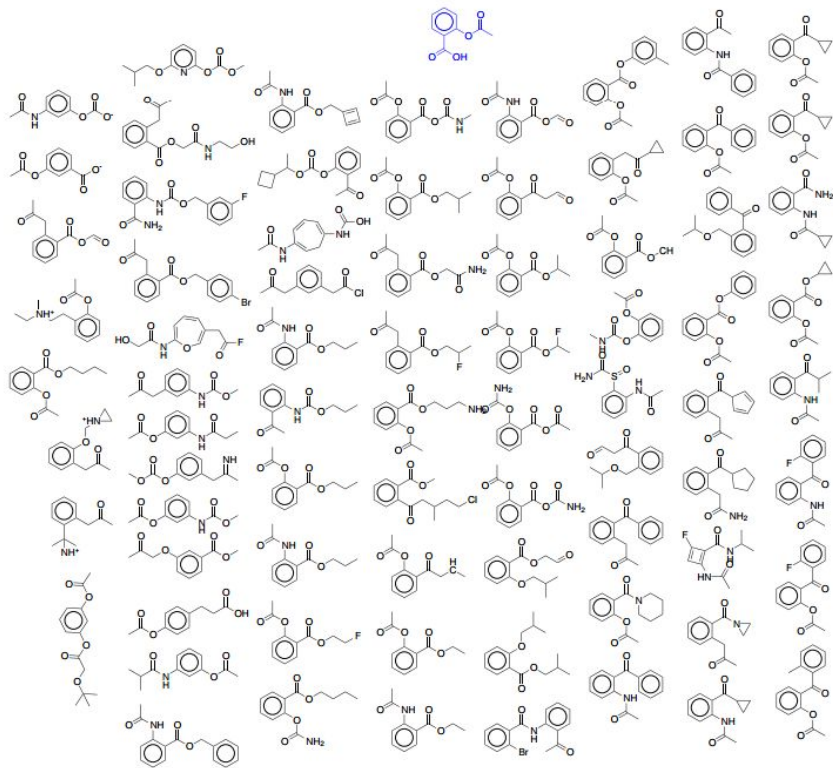


Figure 4: Molecules decoded from randomly-sampled points in the latent space of a variational autoencoder, near to a given molecule (aspirin [2-(acetyloxy)benzoic acid], highlighted in blue).

The slight variation within the molecules decoded for each point in latent space is due to the stochastic nature of the decoder.

Multiple chemical variations of the original compounds were obtained as a consequence of both the probabilistic sampling of the VAE and stochastic nature of the SMILES string decoding.



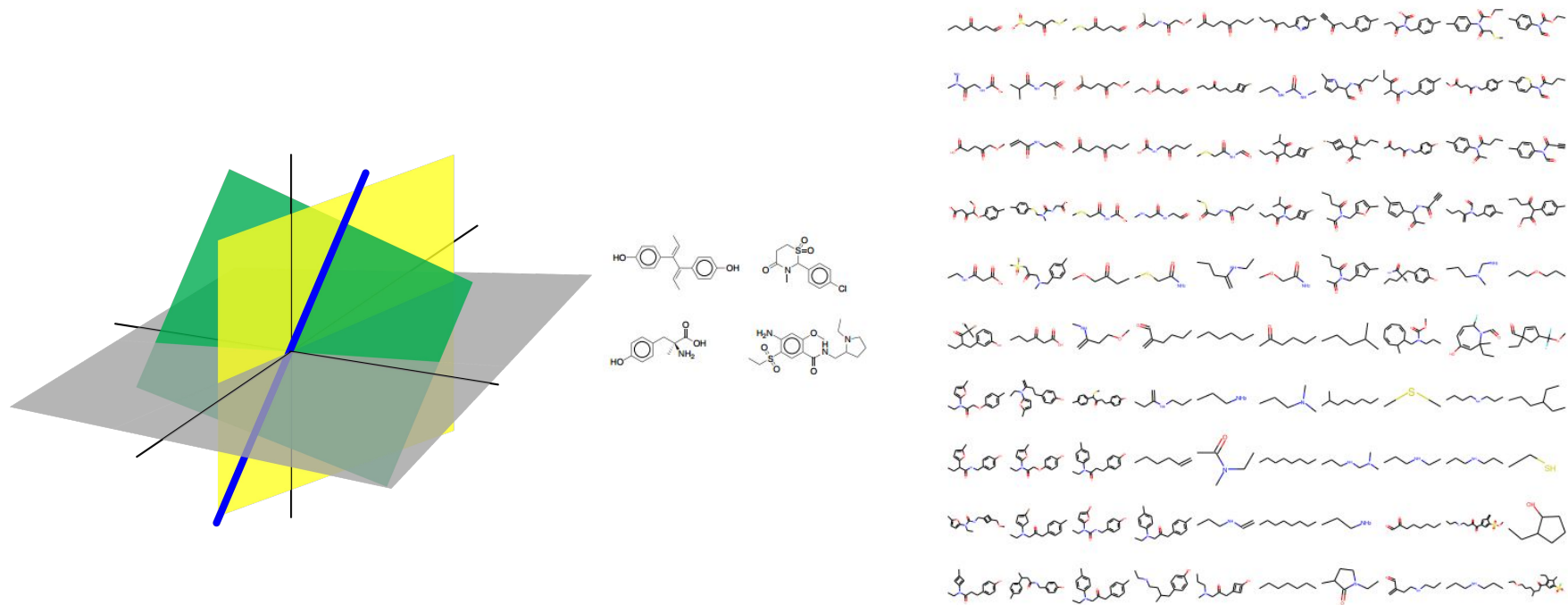


Figure 5: Interpolation. Two-dimensional interpolation between four random drugs. *Left* Starting molecules encoded, whose decodings correspond to the respective four corners of the figure to the right. *Right* Decodings of interpolating linearly between the latent representations of the four molecules to the right.

	Input molecule	Molecule A	Molecule B
Objective values	4.52	5.02	4.68
logP score	8.25	8.07	8.51
The autoencoder can be combined with Bayesian optimization to discover new molecules with better properties than those in the training set.			

## Experiments with OLEDs

Unfortunately, the latent vectors selected by the Bayesian optimization procedure either did not produce valid SMILE strings, or the resulting SMILES were already found in the training data.

We believe the molecule autoencoder failed in this case to learn a generalizable latent representation of chemical space.

Probably because the training molecules were very similar to each other. It could be solved by training the autoencoder using larger amounts of diverse, unlabeled molecules.

# Limitations

# Architecture

- **Variational autoencoder** does not know in advance what properties will be optimized. No guarantee on smoothness of functional values in the latent space.
- **String-based representation** doesn't have desired invariants (rotational and translational). Try a graph-based representation?
  - **Convolutional Networks on Graphs for Learning Molecular Fingerprints**
  - **Molecular Graph Convolutions: Moving Beyond Fingerprints**

# Invalid SMILES Strings

- Some of the strings produced by the model actually do not map to valid molecules
- Notice: The model in the paper is only trained on positive examples

# Evaluation

- Limited comparison between naïve autoencoder and variational autoencoder
- How does this method fare for other properties?
- Found drugs with higher objective values 5.02 and 4.68 vs. 4.52 of the best molecule in the training set.
  - Is this delta significant?
- Verification:
  - The “better” molecules have not been tested experimentally.

**Thank You**