UNIVERSITY OF LIVERPOOL

Deep learning optimal molecular scintillators for dark matter detection

Cameron Cook, Carlos Blanco, Juri Smirnov

2501.00091

Overview

1. Motivation
- Why molecular scintillators?
- Why machine learning?

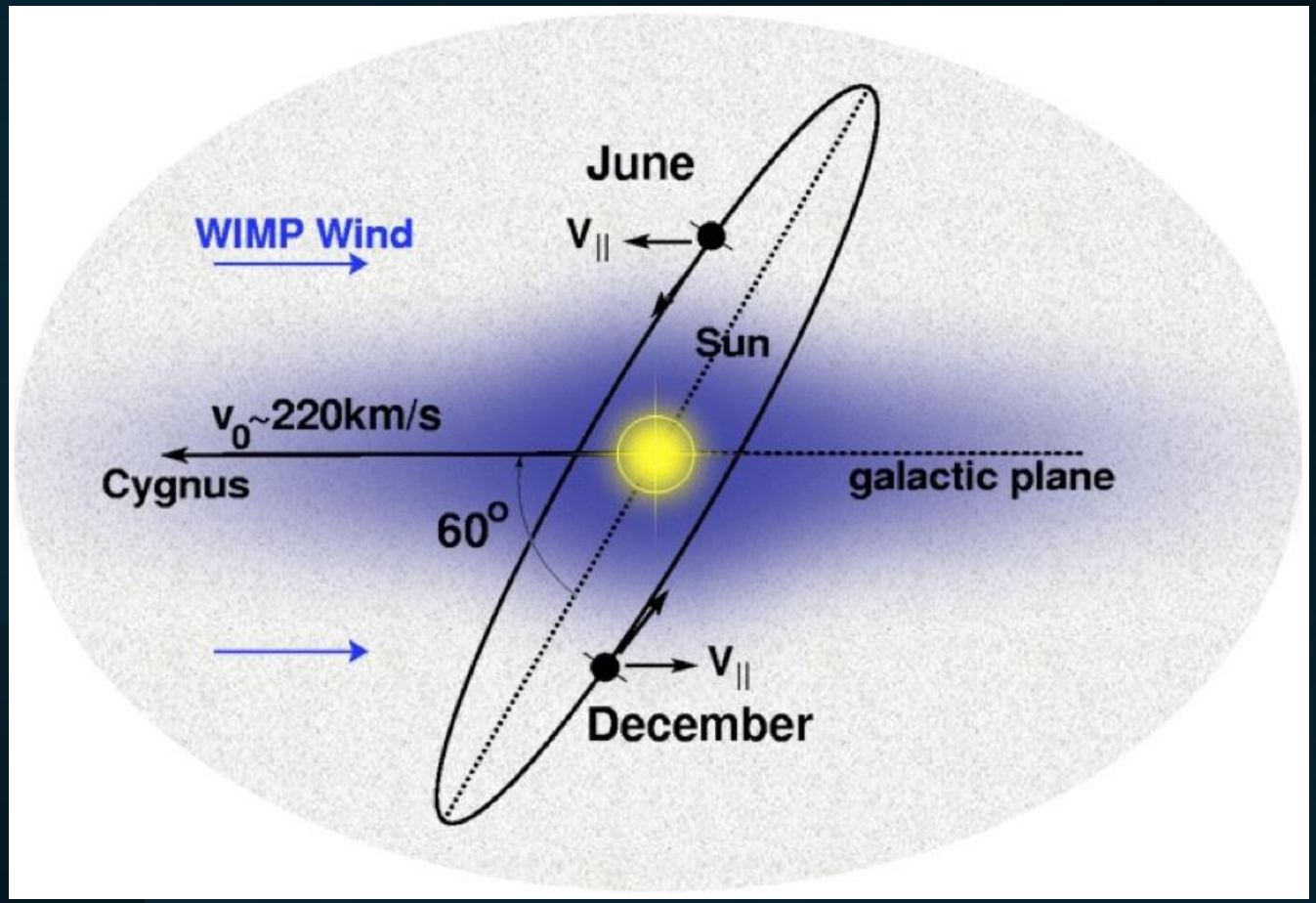
2. Machine learning
- What is a neural network?
- Variational autoencoders, multi – layered perceptrons

3. Results
- Clustering
- Backbones / motifs

4. The future

DM wind

- Dark matter halo formed around the galaxy (spherical)
- We have some peculiar velocity with respect to the dark matter halo
- ~220km/s
- Daily/annual modulation



https://www.quantumdiaries.org/2013/07/01/getting-our-hands-on-dark-matter/

DM kinetic energy

- Relative velocity: 220km/s (~10^{-3}c)
- KE ~ (mass) 10^{-6}c
- MeV scale DM has KE of ~1eV

Molecular scintillators

- Molecular transitions are in the eV range!
- Organic molecules can be crystallised!
- MeV scale mass means we should ‘see’ more DM than we would with WIMPs!

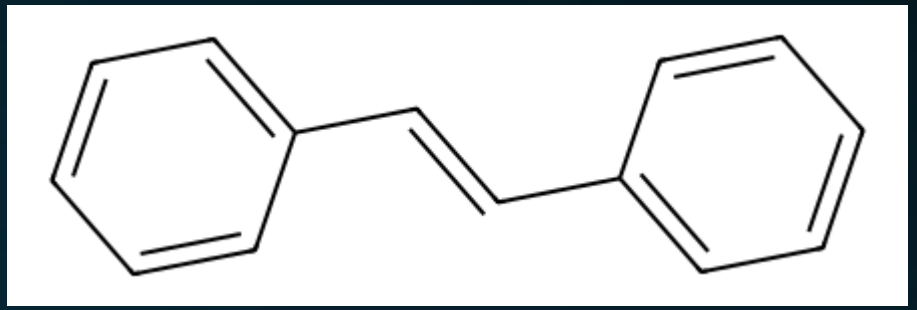
- Our aim:

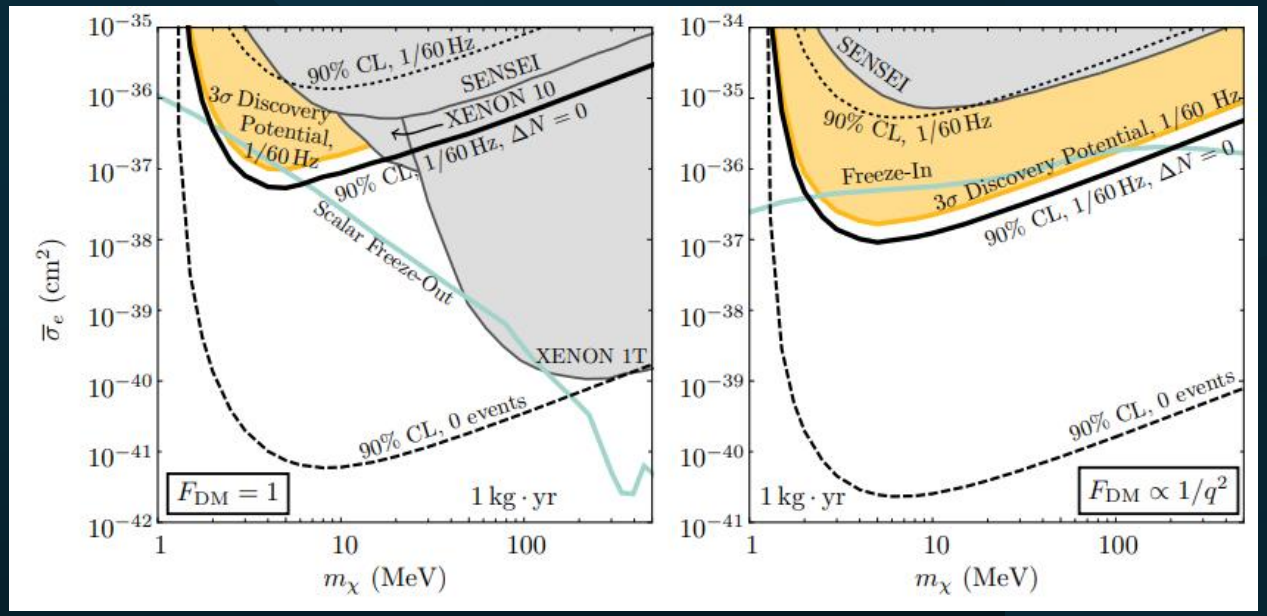
1. High transition probability between states (expressed as oscillator strength -> analogous to coupling strength)
2. Low transition energies
3. Crystallisable

A first go (Trans-Stilbene)

2103.08601

- Carlos Blanco, Yonatan Kahn, Benjamin Lillard, Samuel McDermott
- Experimentally calibrated trans-stilbene
- Produced hypothetical sensitivity plots





Trans-stilbene

- Transition energy: ~few eV
- Oscillator strength: ~0.8

- Pretty good. The best-known crystallisable molecules are only slightly better
- We can do better -> machine learning

Overview

1. Motivation
- Why molecular scintillators?
- Why machine learning?

2. Machine learning
- What is a neural network?
- Variational autoencoders, multi – layered perceptrons

3. Results
- Clustering
- Backbones / motifs

4. The future

Overall plan

- Feed an enormous database of 3 million molecules into a variational autoencoder, generate a bunch of new molecules, calculate the transition energies and the oscillator strengths and then cluster the top molecules to find desirable motifs / backbones.
- 3 main components:

1. Generation (variational auto-encoder)
2. Regression (multi-layered perceptron)
3. Clustering

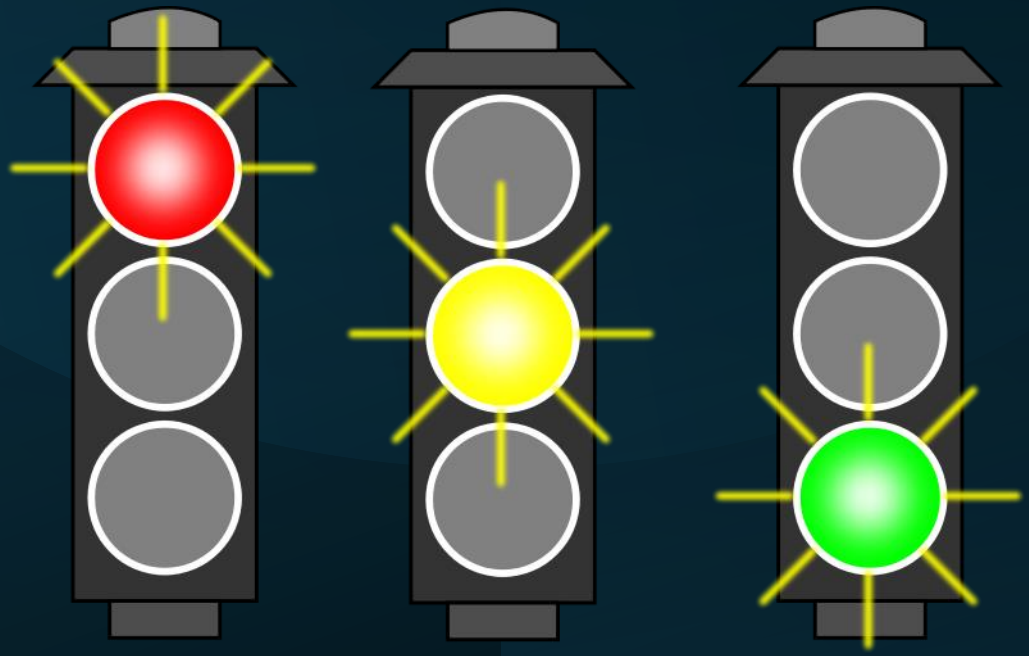
Neural networks

- Two main algorithms

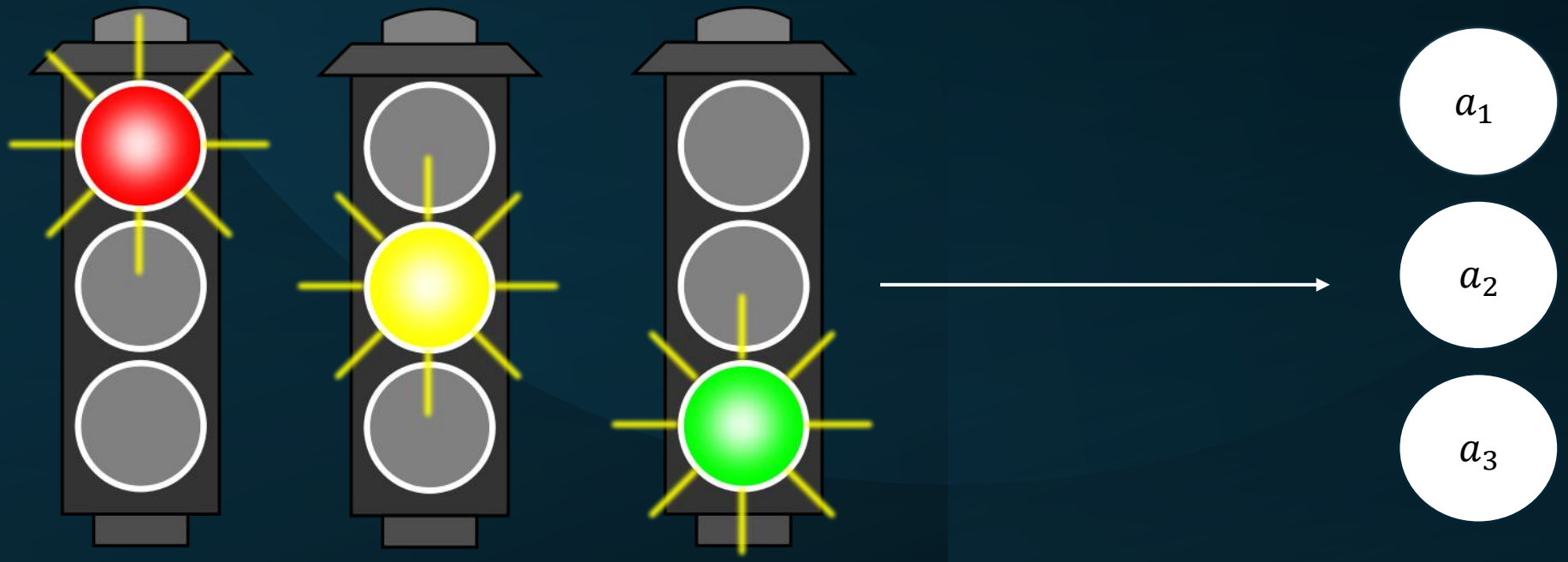
1. Forward propagation: a series of weights, biases, activation functions and maybe more
2. Backward propagation: gradient descent algorithm against a loss/cost function.

Traffic lights

- Task: Predict what signal is active in a traffic light



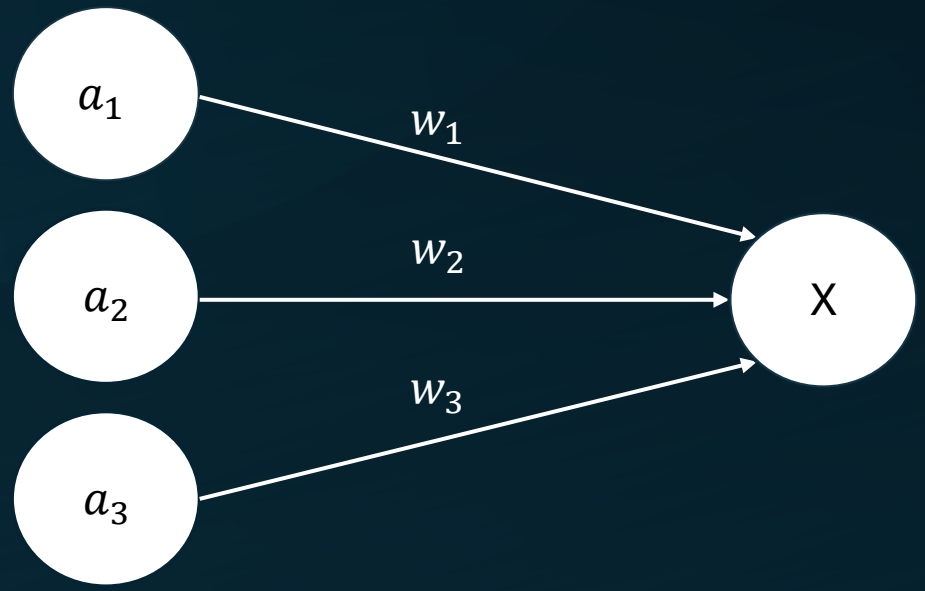
Traffic lights



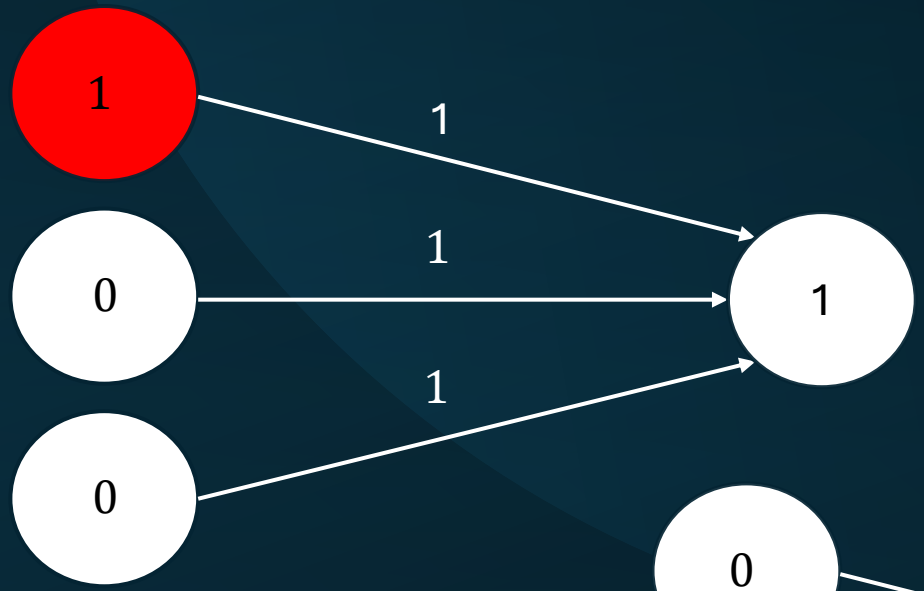
Traffic lights

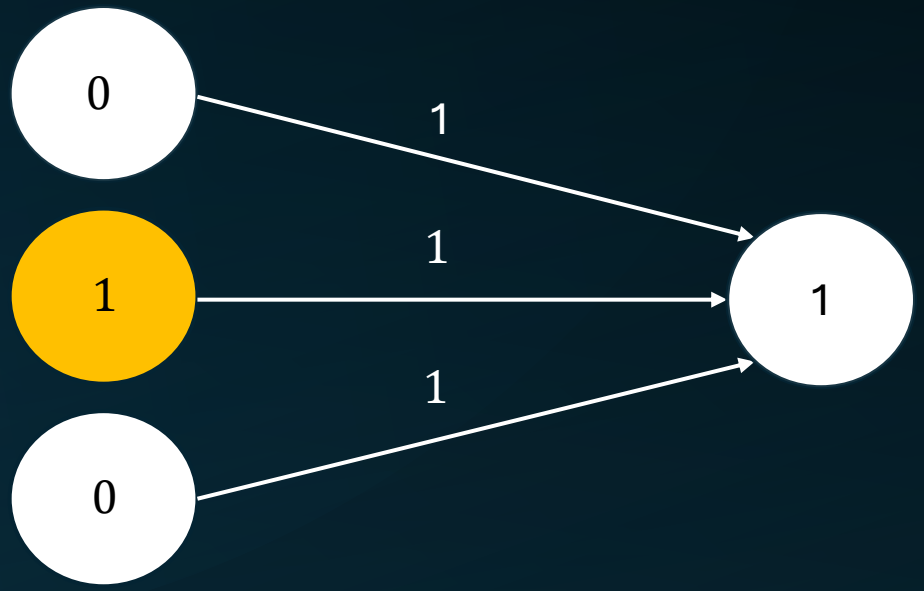
- Let’s say:

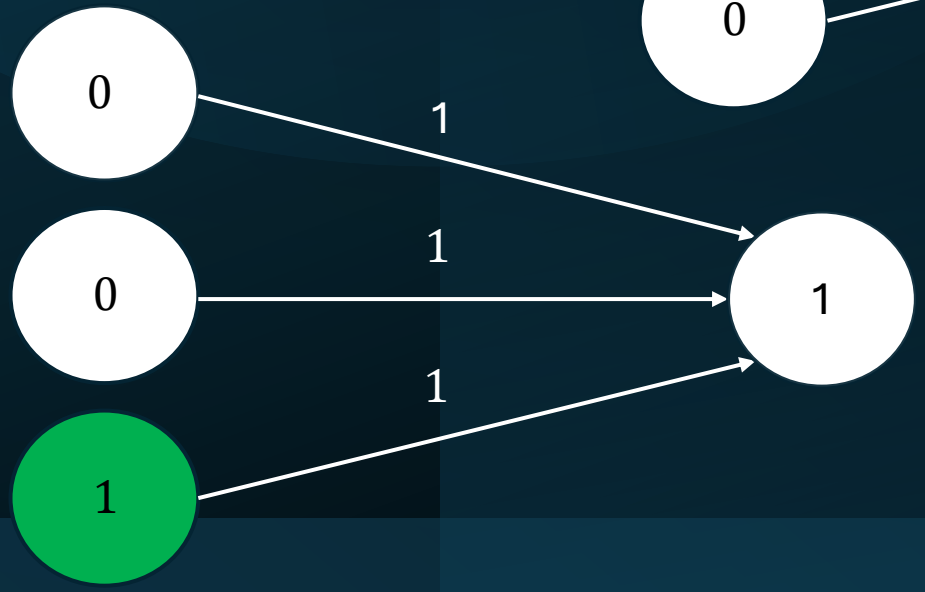
1. When red is active, X=1
2. When amber is active, X=2
3. When green is active, X=3



Traffic lights

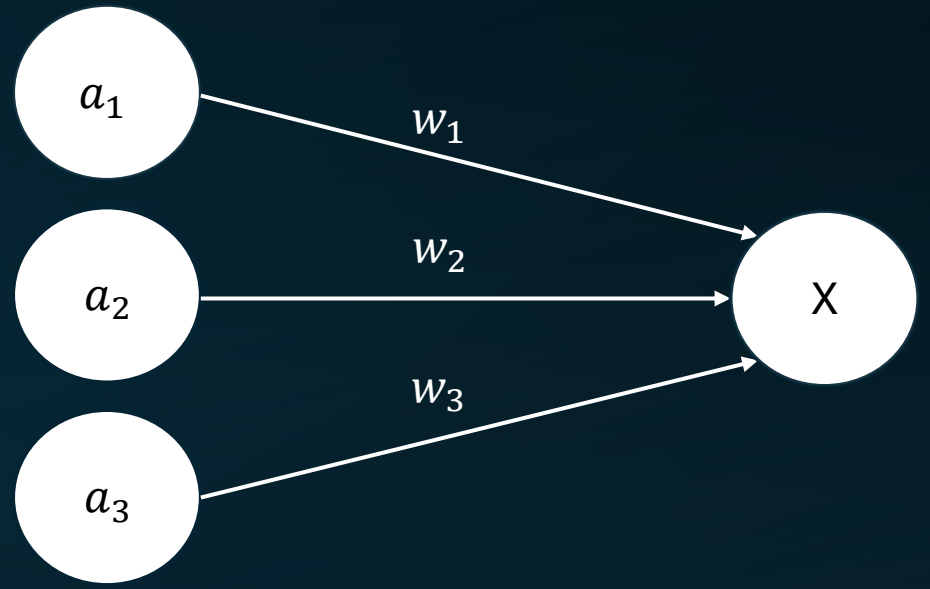






Traffic lights

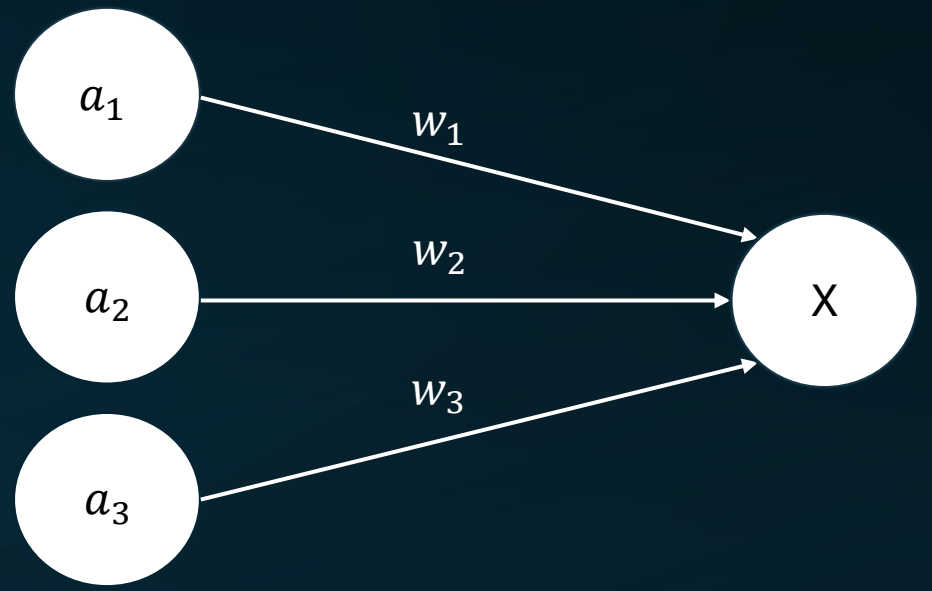
- Predictions / truth:
1. Red: 1 / 1
2. Amber: 1 / 2
3. Green: 1 / 3
- Let’s compute losses:
1. Red: 1-1 = 0
2. Amber: 2-1 = 1
3. Green: 3-1 = 2



Traffic lights

- Weight update: $w'_x = w_x - \alpha \frac{\partial L}{\partial w_x}$
- $X_x = w_x \cdot a_x$
- $L_x = |X_x - t_x| = |w_x \cdot a_x - t_x|$

- $\frac{\partial L_x}{\partial w_x} = \frac{\partial L_x}{\partial X_x} \cdot \frac{\partial X_x}{\partial w_x} = -\operatorname{sgn}(w_x \cdot a_x - t_x) \cdot a_x$
- $\frac{\partial L_1}{\partial w_1} = \frac{\partial L_1}{\partial X_x} \cdot \frac{\partial X_x}{\partial w_1} = -\operatorname{sgn}(w_1 \cdot a_1 - t_1) \cdot a_1$
- $\frac{\partial L_2}{\partial w_2} = \frac{\partial L_2}{\partial X_2} \cdot \frac{\partial X_2}{\partial w_2} = -\operatorname{sgn}(w_2 \cdot a_2 - t_2) \cdot a_2$
- $\frac{\partial L_3}{\partial w_3} = \frac{\partial L_3}{\partial X_x} \cdot \frac{\partial X_x}{\partial w_1} = -\operatorname{sgn}(w_3 \cdot a_3 - t_3) \cdot a_3$



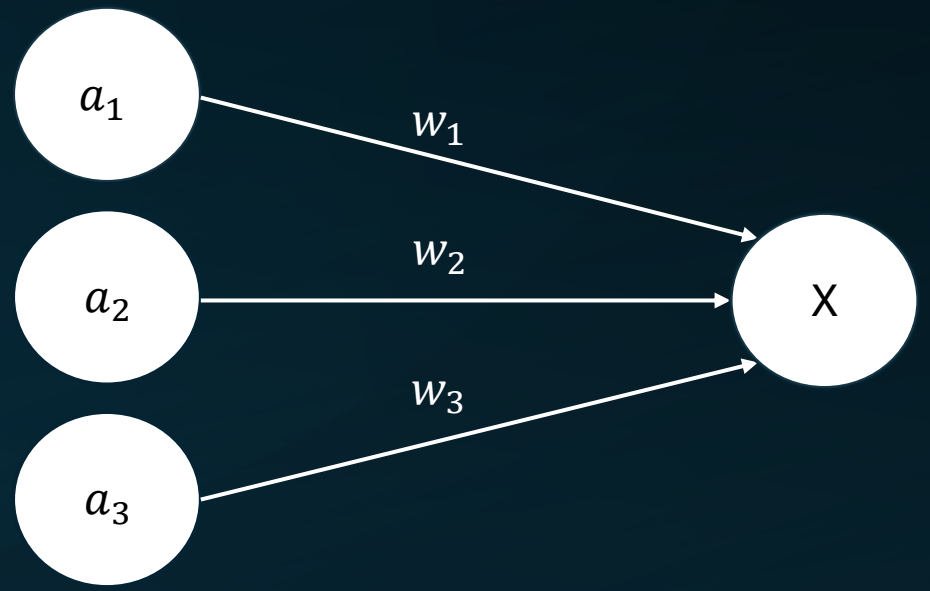
Backward propagation

- Let's compute our gradients:

1. $\frac{\partial L}{\partial w_1} = \text{sgn}(1 \cdot 1 - 1) \cdot 1 = 0$
2. $\frac{\partial L}{\partial w_2} = \text{sgn}(1 \cdot 1 - 2) \cdot 1 = -1$
3. $\frac{\partial L}{\partial w_3} = \text{sgn}(1 \cdot 1 - 3) \cdot 1 = -1$

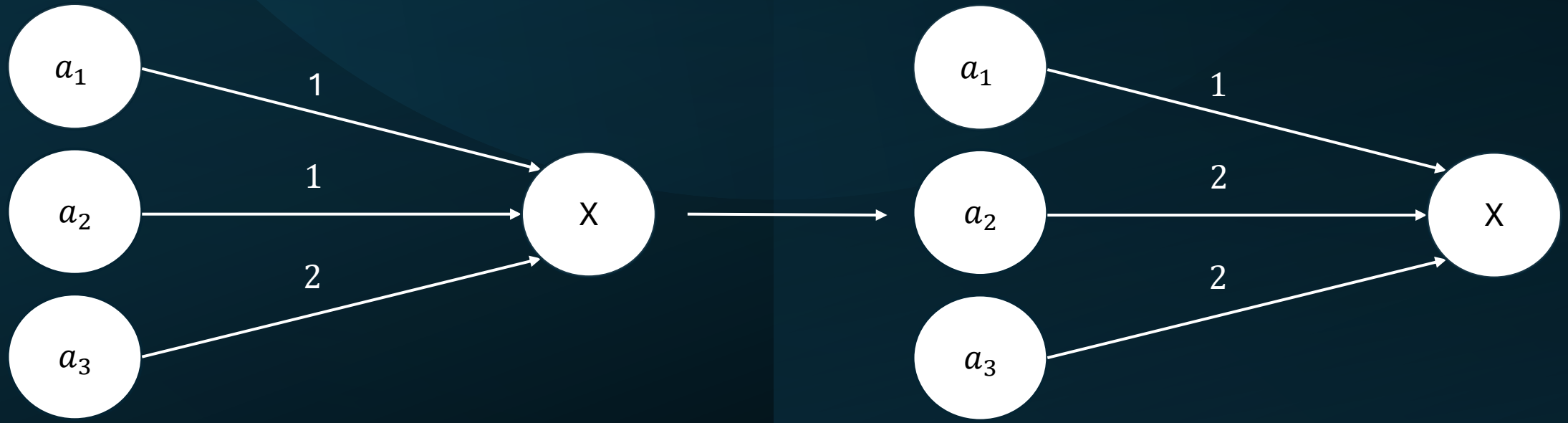
- Update the weights: ($\alpha = 1$)

1. $w_1' = w_1 - \alpha \frac{\partial L}{\partial w_1} = 1 - 1$
2. $w_2' = w_2 - \alpha \frac{\partial L}{\partial w_2} = 1 - 1 \cdot -1 = 2$
3. $w_3' = w_3 - \alpha \frac{\partial L}{\partial w_3} = 1 - 1 \cdot -1 = 2$



Backward propagation

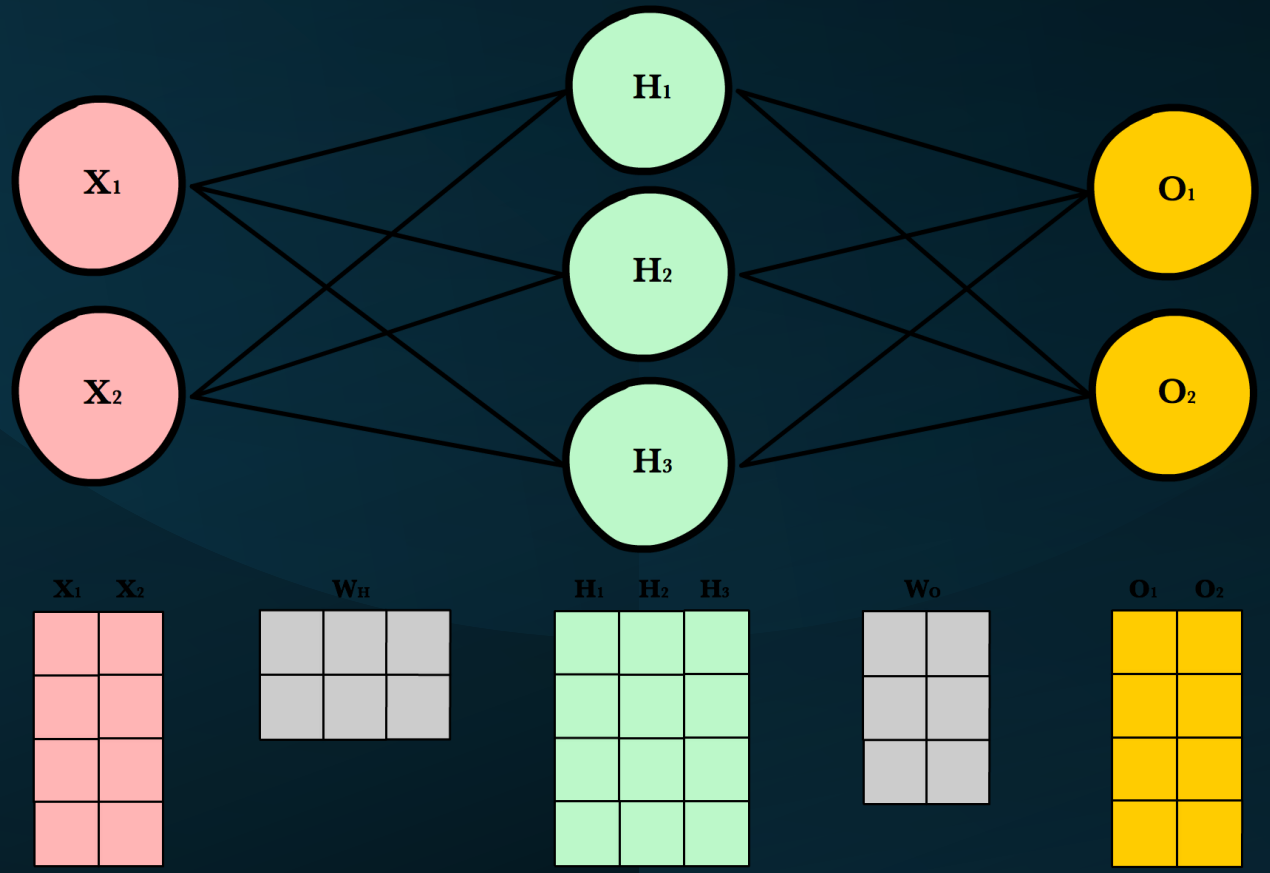
1. $w_1' = 1$
2. $w_2' = 2$
3. $w_3' = 2$



Backward propagation

REPEAT

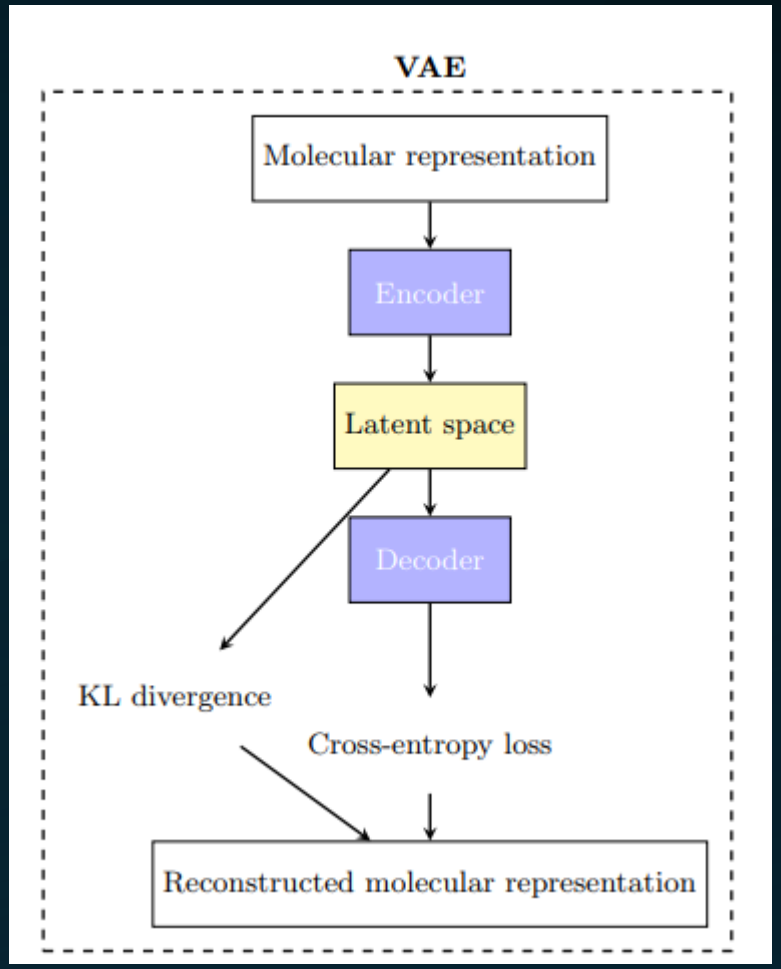
Neural networks



https://ml-cheatsheet.readthedocs.io/en/latest/forwardpropagation.html

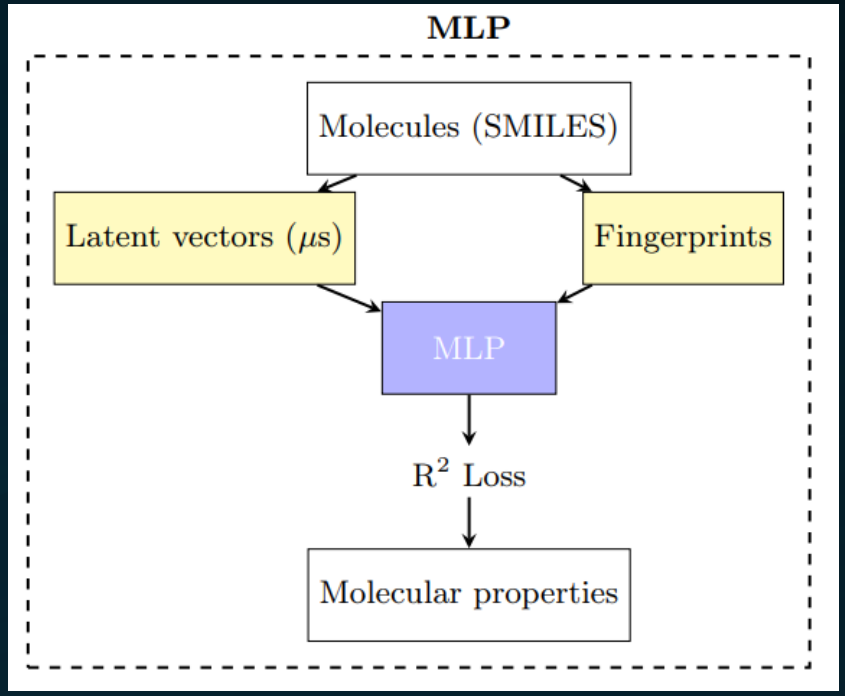
Variational autoencoder (generative)

- Variational autoencoders learn mappings from inputs to N-dimensional vectors which optimally represent information related to the input but also retain the input distribution
- It then learns to decode these N-dimensional vectors
- This N-dimensional encoding space is called the latent space
- Varying within the latent space allows for the generation of new molecules



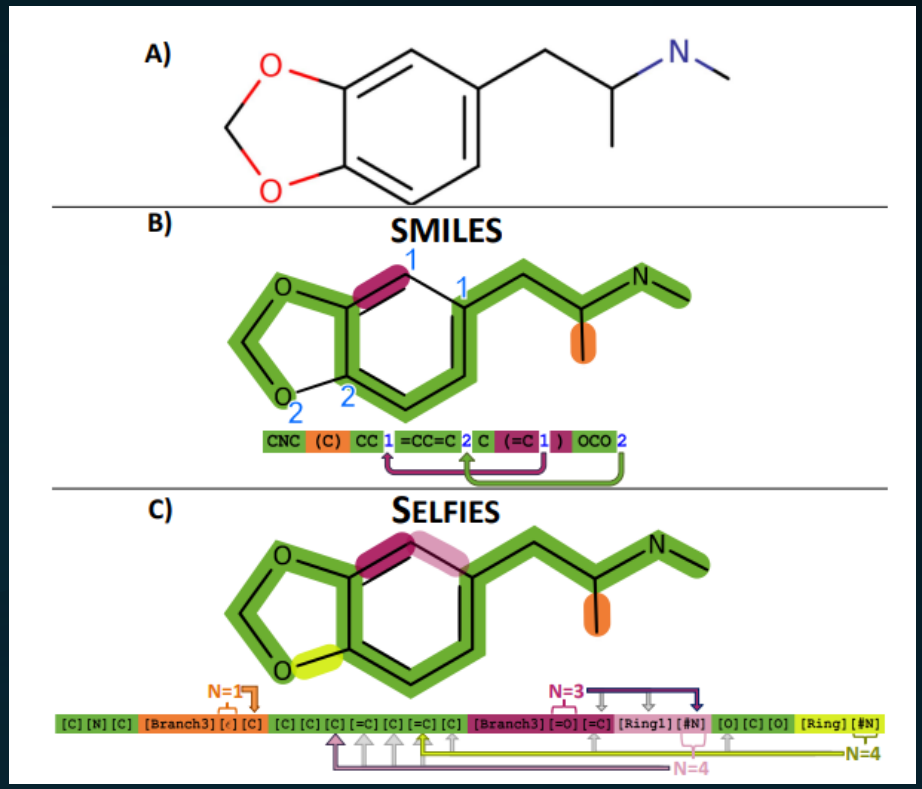
Multi-layered perceptron (regression)

- Much simpler.
- The multi-layered perceptron operates like the previous neural network example.
- It takes inputs and then predicts molecular properties (oscillator strength and transition energies).
- Heavily depends on what sort of inputs you give.



Inputs?

- How do we represent molecules?
- Graphs, chemical languages, or somewhere in-between.
- We need representations which are robust to permutations since this is exactly what the variational autoencoder does.
- Typical representations (SMILES) won't cut it
- Need SELFIES. SELFIES are permutation invariant!



1905.13741

Molecular representations (inputs)

- A typical SELFIES looks like this: ‘[C][=C][C][=C][C][=C][Ring1][=Branch1]’
- Can’t just throw this at the VAE.
- Need to convert to a representation which is readable to a computer.

One-hot encodings

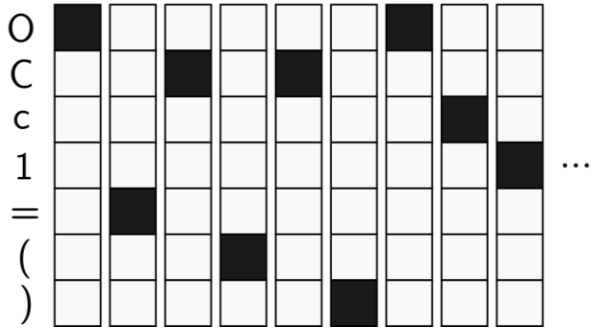
- Black: 1
- White: 0
- Matrix of 1s and 0s.

SMILES

string:

O=C(C)Oc1ccccc1C(=O)O

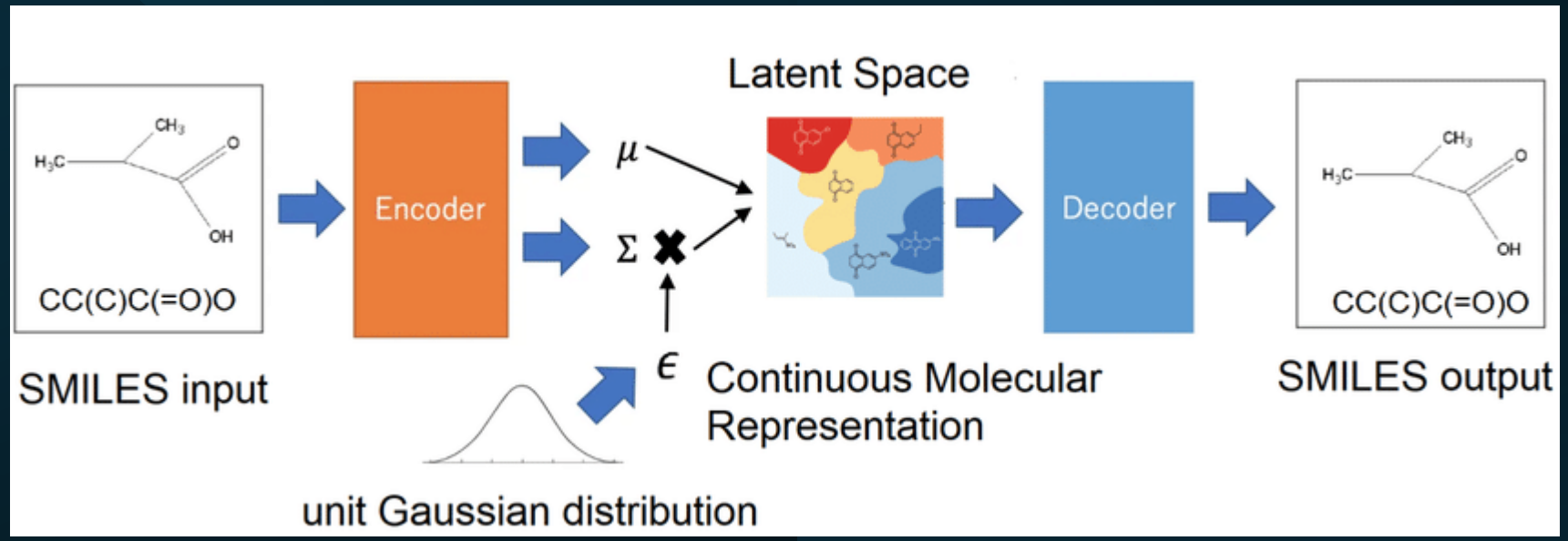
one-hot encoding:



arxiv:1907.01632

Variational autoencoders

- Take one-hot inputs and transform into N-dimensional vectors in the ‘latent space’
- Decode items inside of the latent space into ‘logit’ matrices, i.e., matrices corresponding to unnormalised probabilities



10.1109/ACCESS.2020.3048309

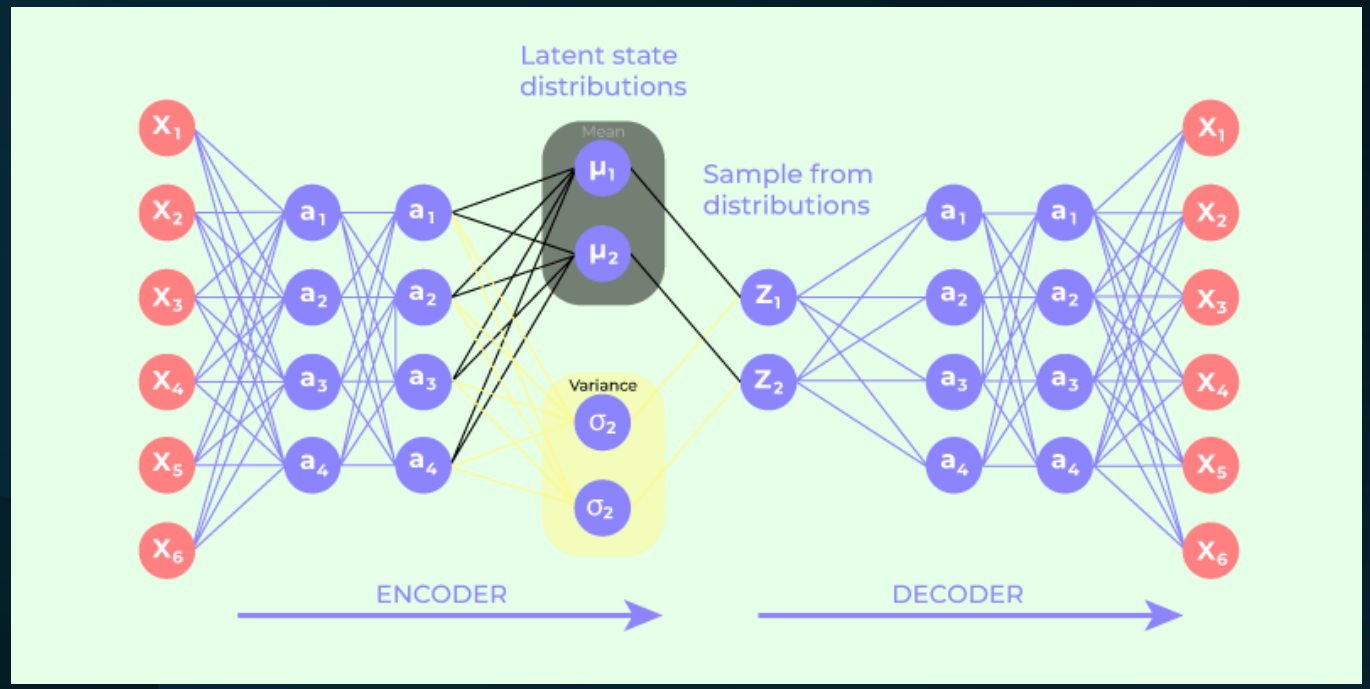
Latent space

- N-dimensional space where indices correspond to properties associated with the inputs
- Each vector in the latent space ought to correspond to some molecule

https://medium.com/vitrox-publication/generative-modeling-with-variational-auto-encoder-vae-fc449be9890e

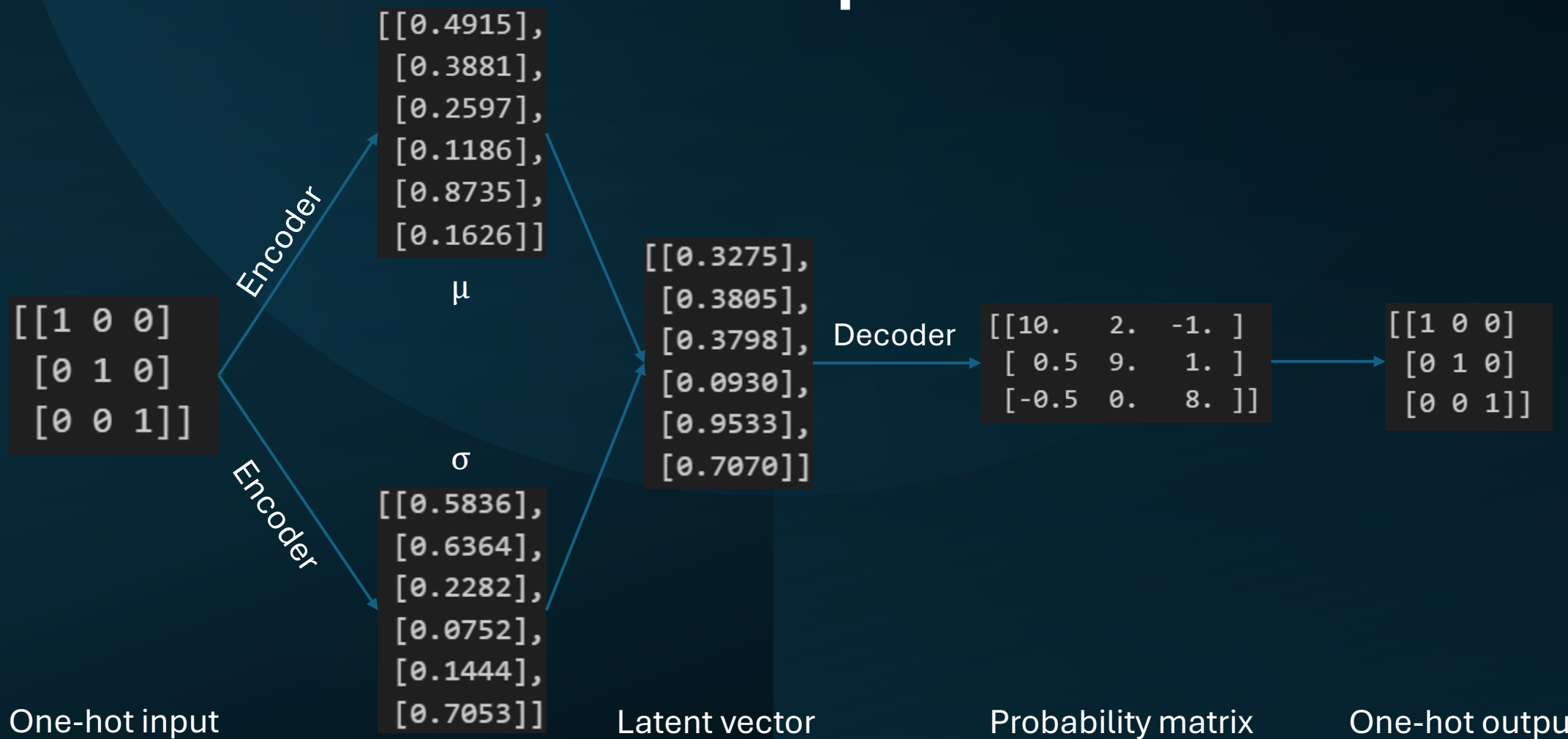
Latent vectors

- The encoder outputs two vectors: μ & σ
- By sampling from the distribution defined by μ & σ, we can generate new latent vectors.
- These new latent vectors ought to correspond to new molecules



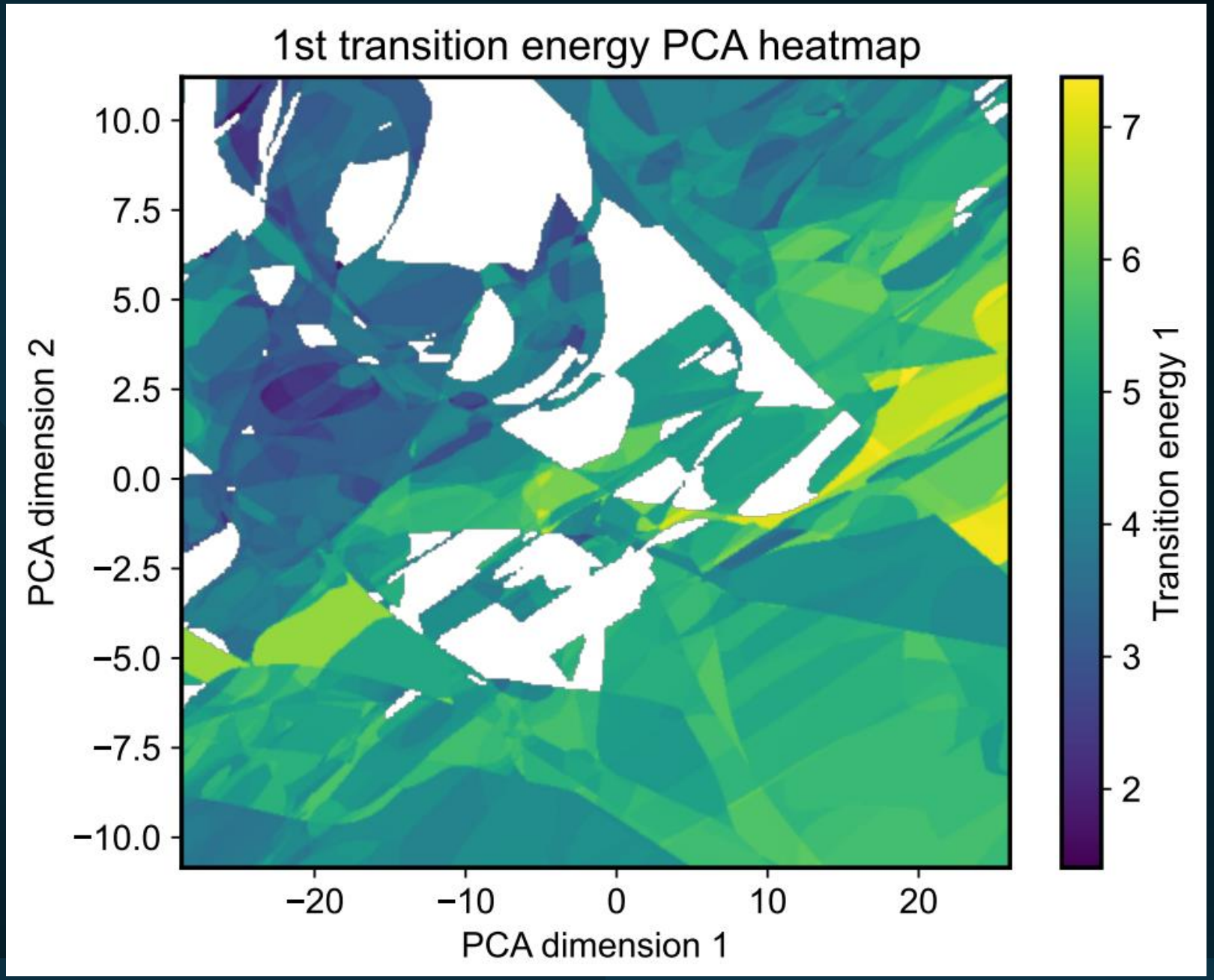
https://www.geeksforgeeks.org/variational-autoencoders/

Example


[[1 0 0]
[0 1 0]
[0 0 1]]
Encoder
[[0.4915],
[0.3881],
[0.2597],
[0.1186],
[0.8735],
[0.1626]]
μ
Encoder
σ
[[0.5836],
[0.6364],
[0.2282],
[0.0752],
[0.1444],
[0.7053]]
[[0.3275],
[0.3805],
[0.3798],
[0.0930],
[0.9533],
[0.7070]]
Decoder
[[10. 2. -1.]
[0.5 9. 1.]
[-0.5 0. 8.]]
[[1 0 0]
[0 1 0]
[0 0 1]]
One-hot input
Latent vector
Probability matrix
One-hot output

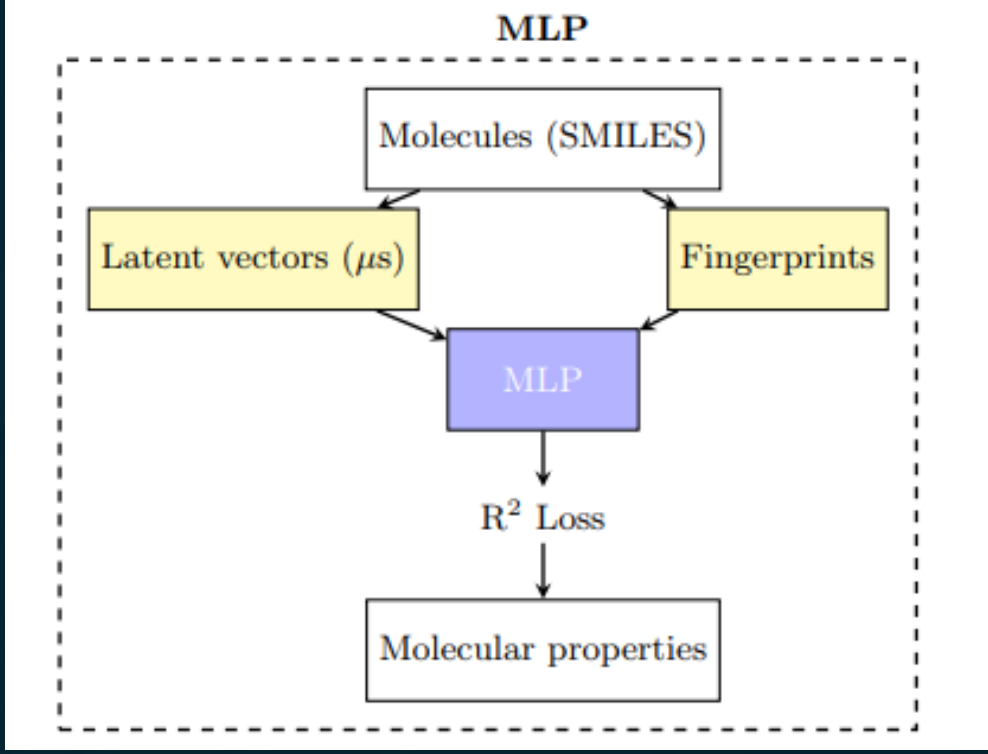
Loss functions

- Reconstruction loss: this loss checks that you are actually reconstructing your inputs properly
- KLD loss: enforces that the distribution of latent vectors ‘looks’ like the distribution of molecular one-hots


1st transition energy PCA heatmap
PCA dimension 2
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−20
−10
0
10
20
PCA dimension 1
Transition energy 1
7
6
5
4
3
2

Multi-layered perceptron (MLP)

- One can imagine this like the encoder but with just two dimensions
- We transform the one-hots into 2-D outputs: the oscillator strengths and transition energies.



Generation strategy

- Two options:

1. Gradient descent within the latent space, using the MLP to predict how ‘good’ a certain position is. Very elegant and, in theory, would give you the absolute minimum
2. Brute force.

Gradient descent

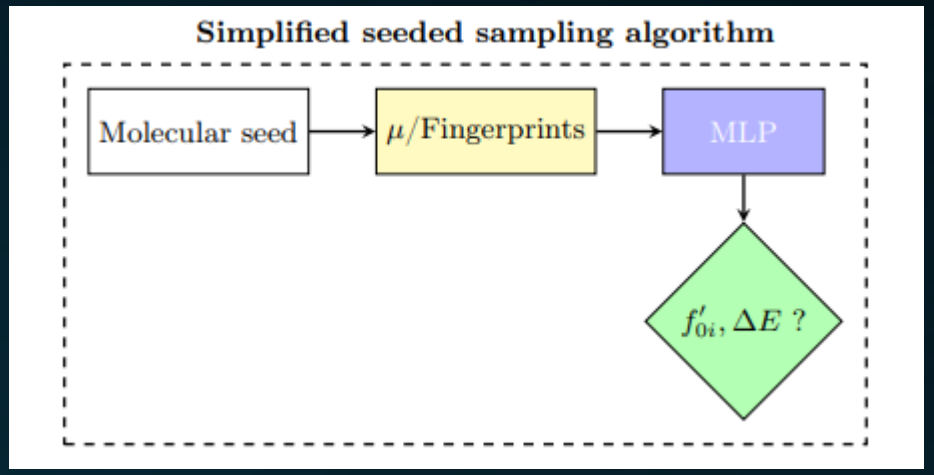
- Two options:

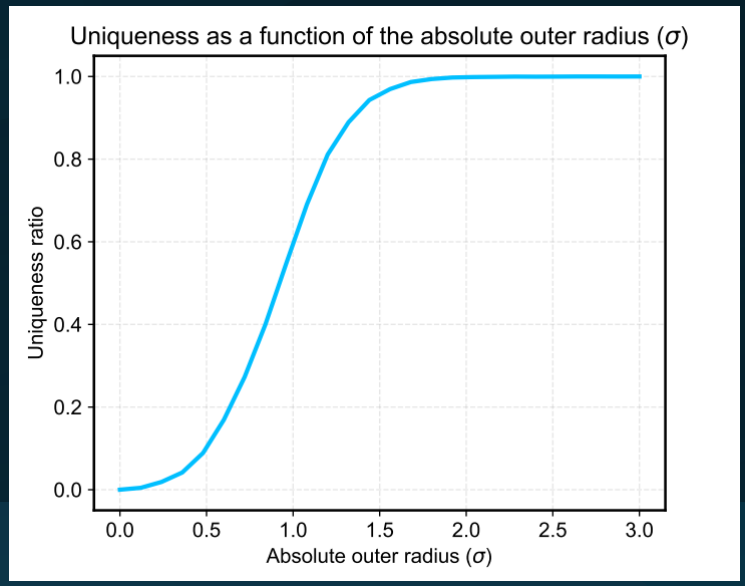
1. Gradient descent within the latent space, using the MLP to predict how ‘good’ a certain position is. Very elegant and, in theory, would give you the absolute minimum
2. Brute force.

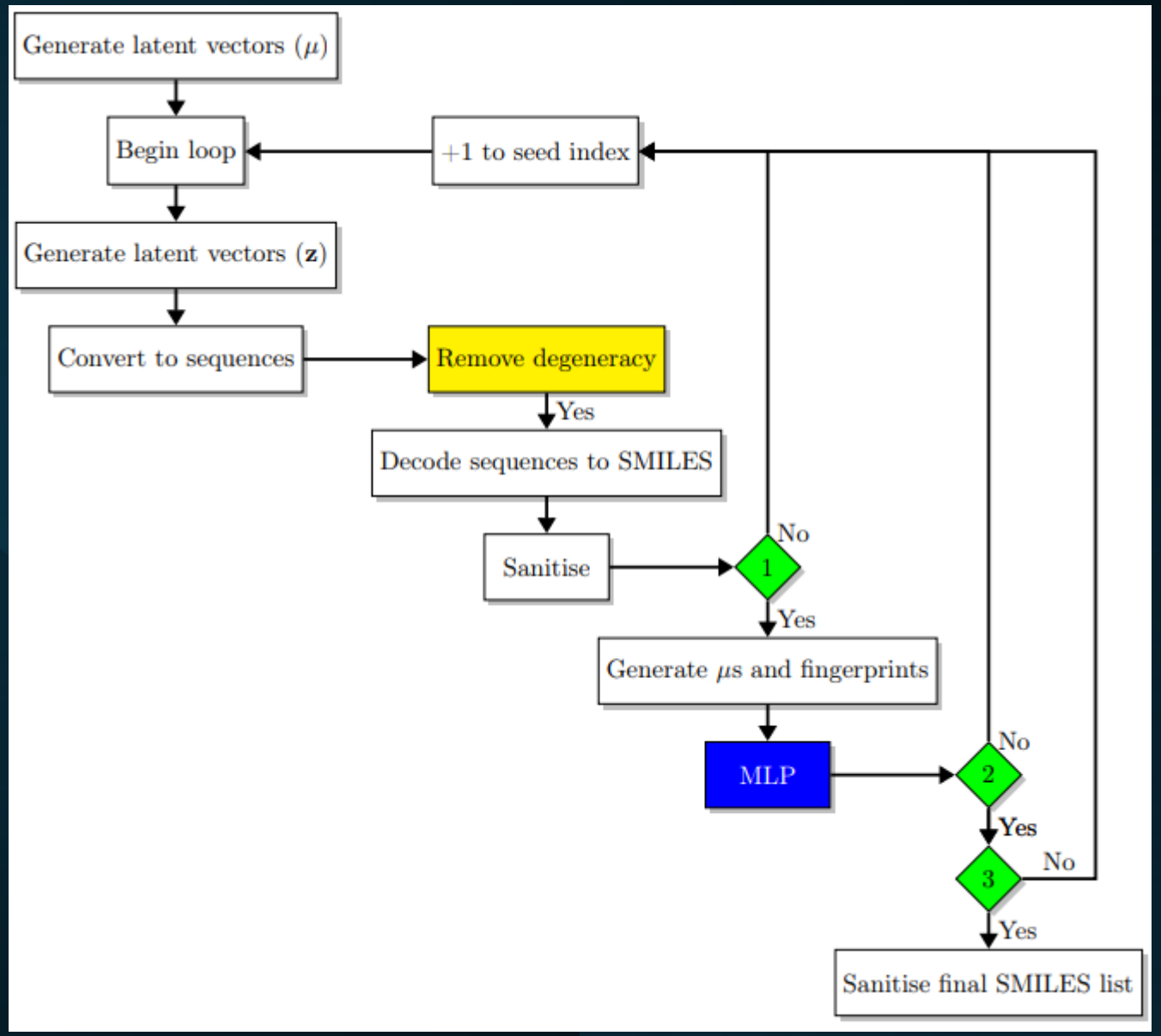
- Gradient descent suffers from too much success. The latent space is too large to be mapped well. Only sampling via the normal process leads to genuine mappings by the decoder.
- Ultimately get nonsense out.

Brute force algorithm

1. Take entire input database and sample around each molecule (~1000 samples per molecule)
2. Tag outputs by how well they match our criteria.
3. Condense down such that we retain 90% of the seeds which account for our final outputs
4. Use this seed list to generate (~0.3% of dataset) but generate ~500,000 samples per molecule






Generate latent vectors (μ)
Begin loop
+1 to seed index
Generate latent vectors (z)
Convert to sequences
Remove degeneracy
Yes
Decode sequences to SMILES
Sanitise
No
1
Yes
Generate μs and fingerprints
MLP
No
2
Yes
No
3
Yes
Sanitise final SMILES list

Overview

1. Motivation
- Why molecular scintillators?
- Why machine learning?
2. Machine learning
- What is a neural network?
- Variational autoencoders, multi – layered perceptrons
3. Results
- Clustering
- Backbones / motifs
4. The future

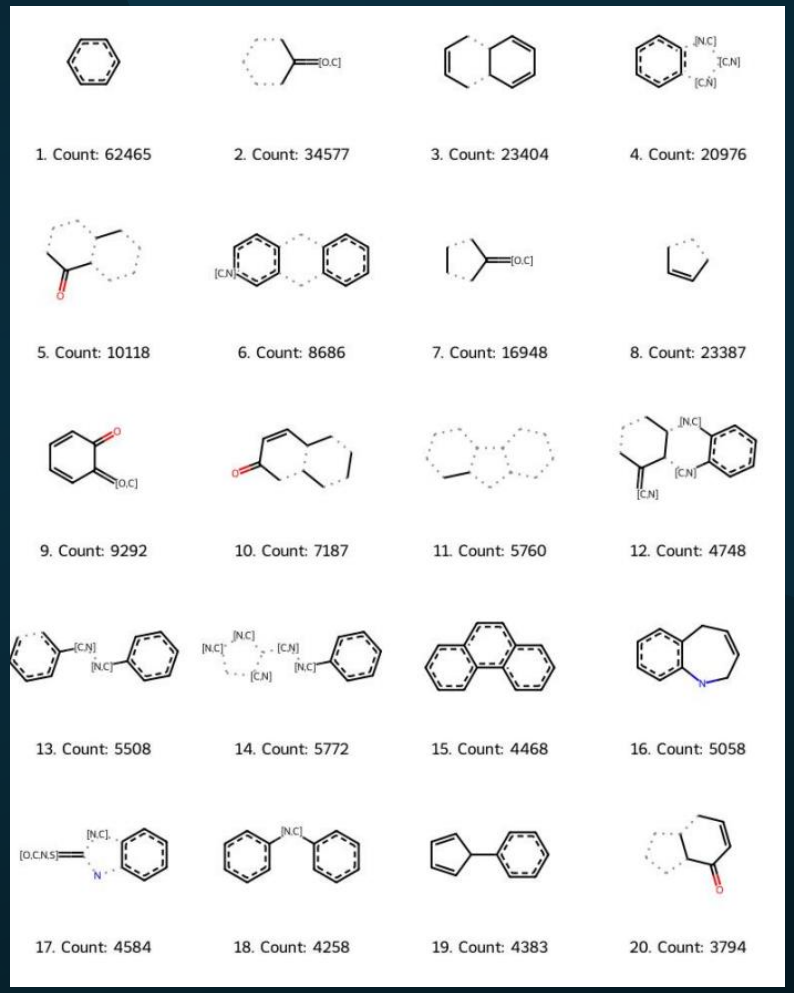
Clustering

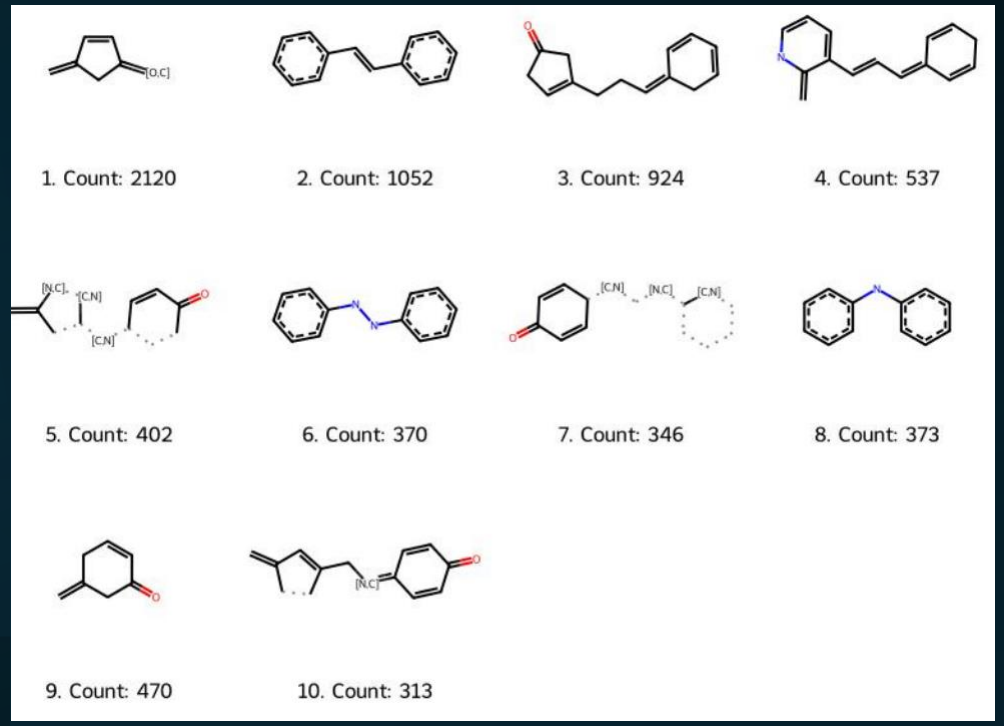
- In the end, we are left with around 8.4 million molecules outputted by the brute force algorithm.
- If we cluster them, we can find common motifs and specific backbones which we find to be promising.
- Also, we can match these common motifs to known organic molecules known to form crystals

- Clustering techniques:

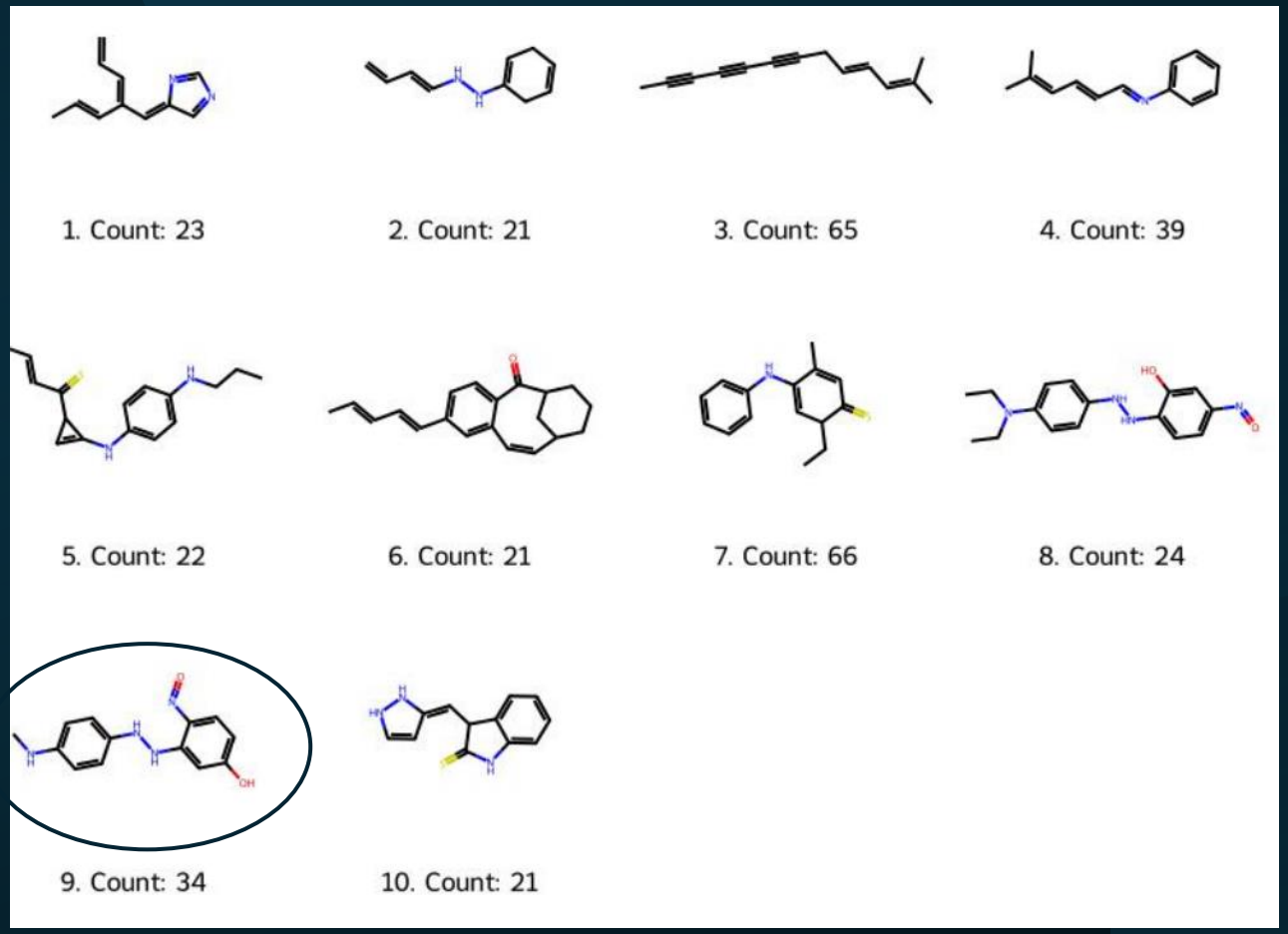
1. Motifs: BitBirch
2. Backbones: Single-linkage clustering

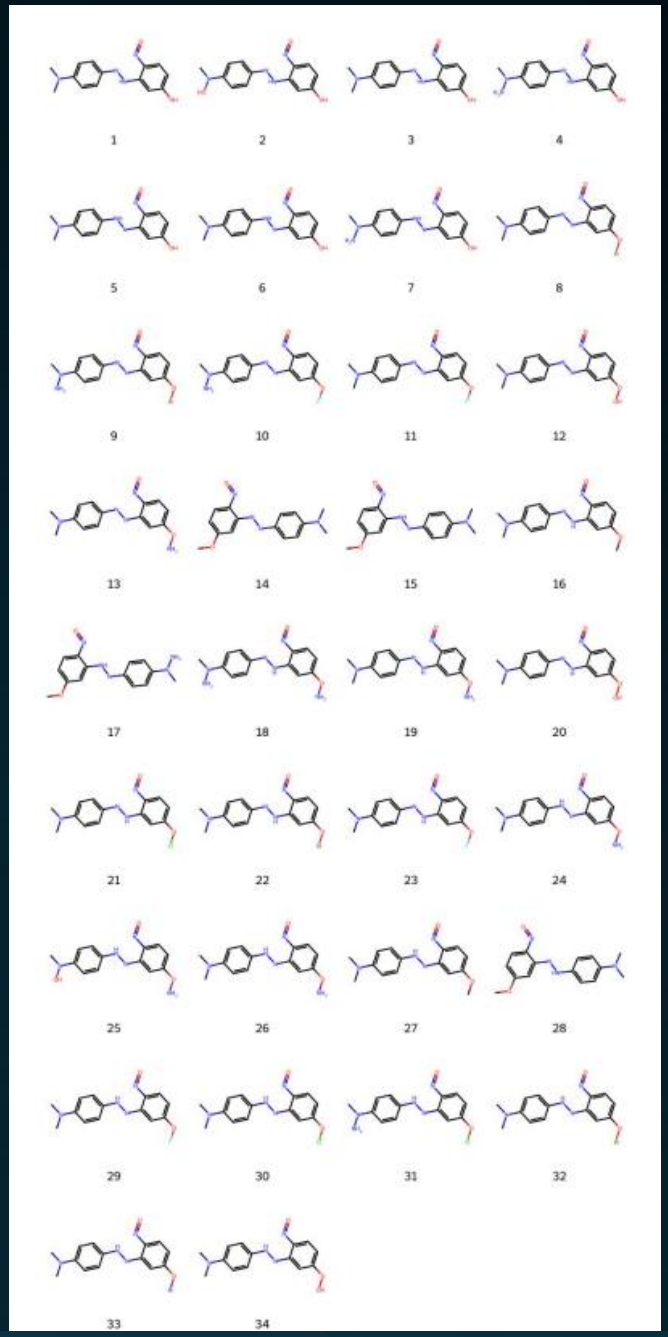
Motifs


1. Count: 62465
2. Count: 34577
3. Count: 23404
4. Count: 20976
5. Count: 10118
6. Count: 8686
7. Count: 16948
8. Count: 23387
9. Count: 9292
10. Count: 7187
11. Count: 5760
12. Count: 4748
13. Count: 5508
14. Count: 5772
15. Count: 4468
16. Count: 5058
17. Count: 4584
18. Count: 4258
19. Count: 4383
20. Count: 3794

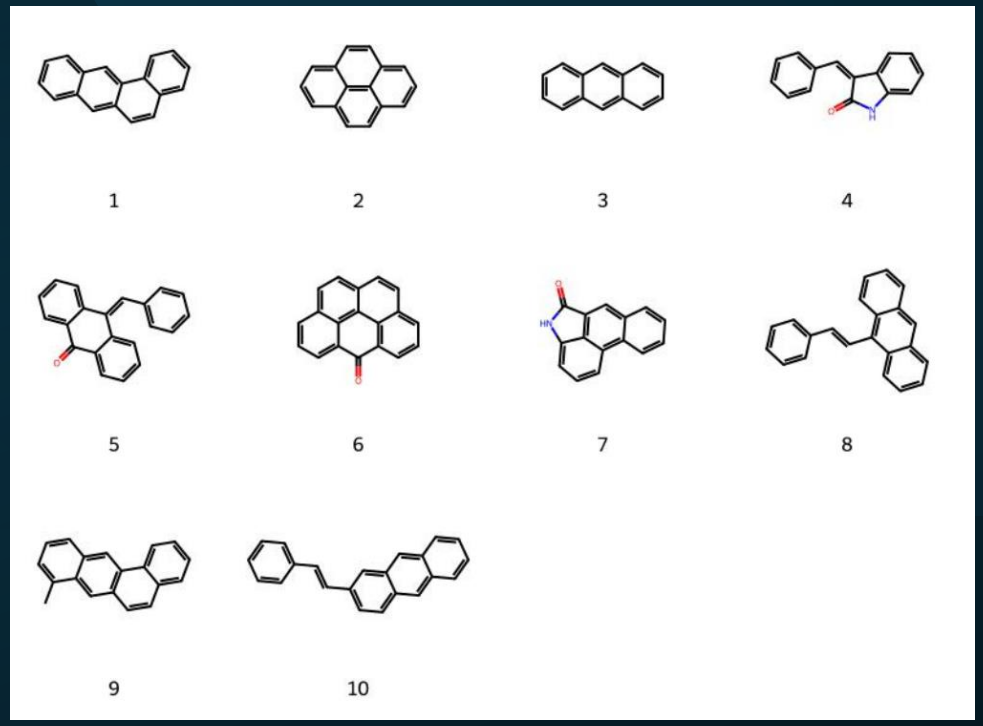

1. Count: 2120
2. Count: 1052
3. Count: 924
4. Count: 537
5. Count: 402
6. Count: 370
7. Count: 346
8. Count: 373
9. Count: 470
10. Count: 313

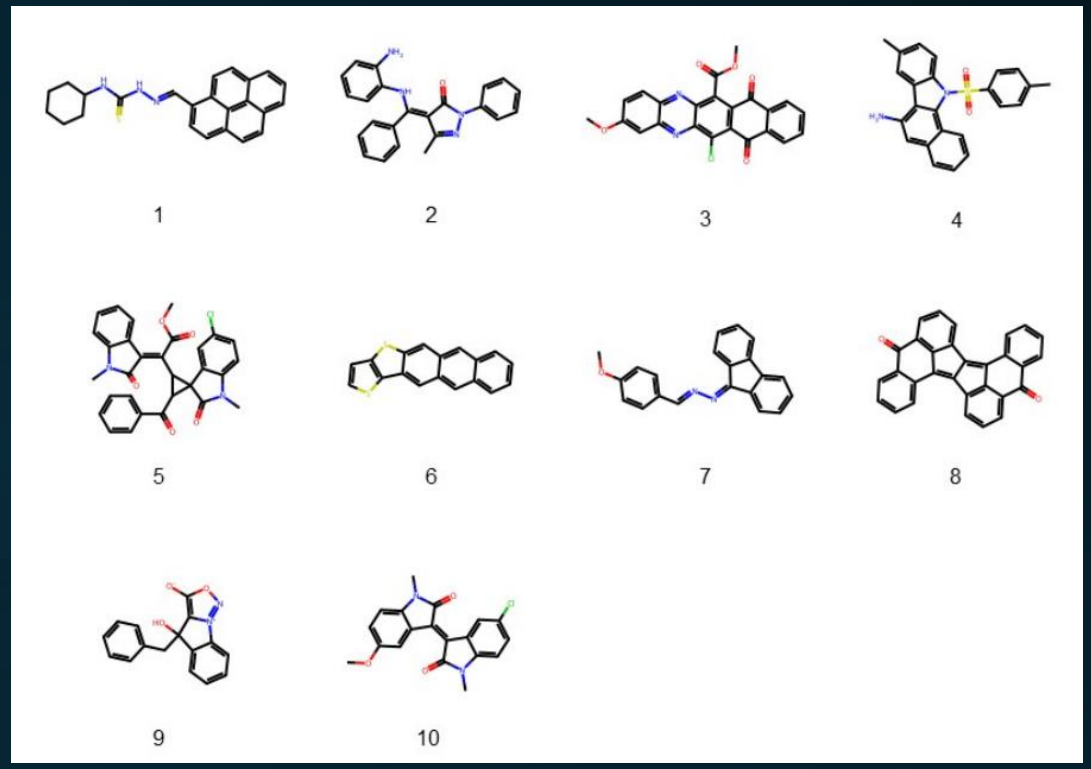
Backbones


1. Count: 23
2. Count: 21
3. Count: 65
4. Count: 39
5. Count: 22
6. Count: 21
7. Count: 66
8. Count: 24
9. Count: 34
10. Count: 21



Crystal structures





Overview

1. Motivation
- Why molecular scintillators?
- Why machine learning?
2. Machine learning
- What is a neural network?
- Variational autoencoders, multi – layered perceptrons
3. Results
- Clustering
- Backbones / motifs
4. The future

The future

- Machine learning side:

1. Better VAE -> swap out GRUs for LSTMs.
2. Graph neural network predictions -> use graphs instead of SELFIES to do predictions.

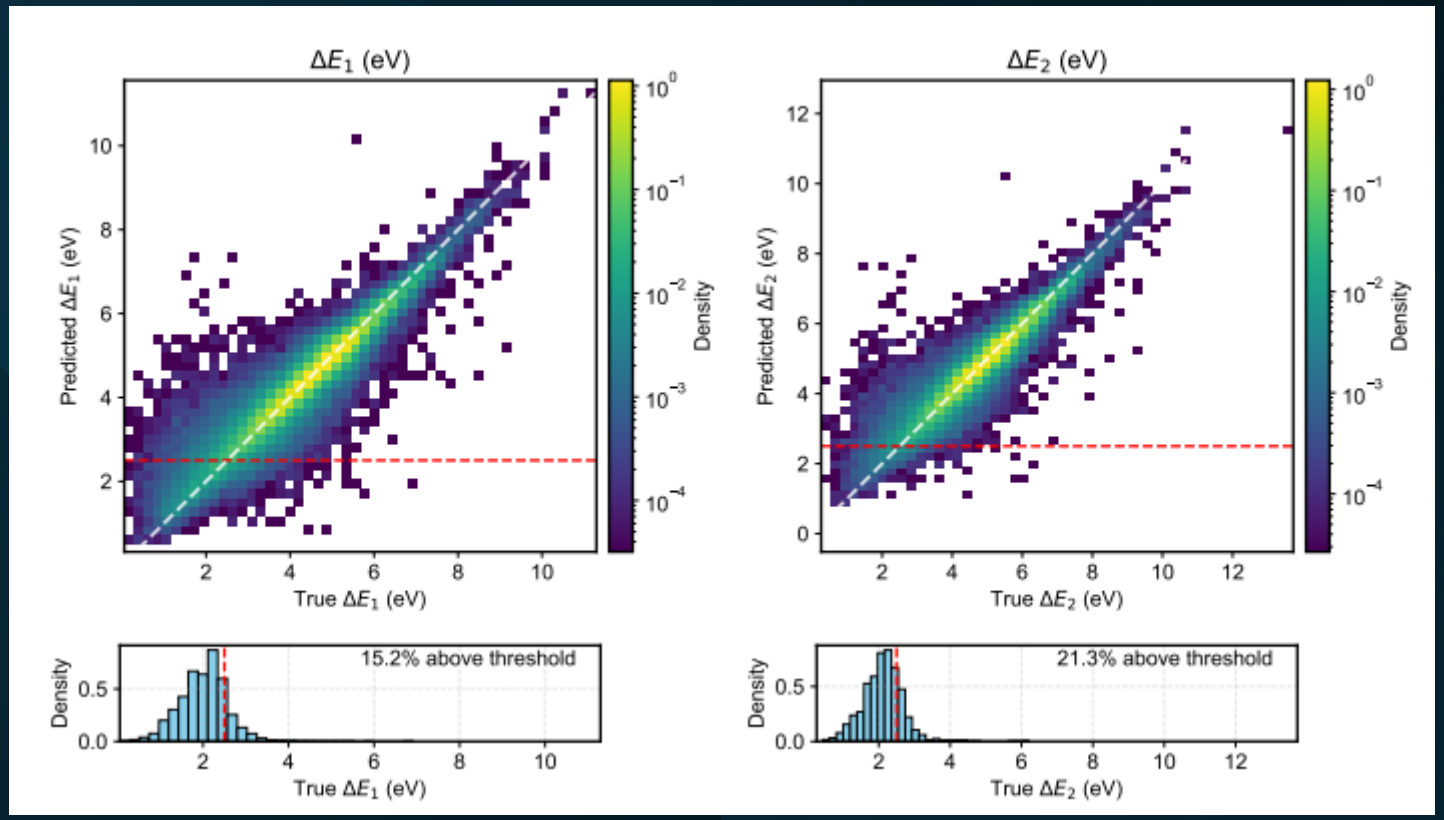
- Chemistry side:

1. Set up a pipeline to calculate interaction rates.
2. Work with chemists to make these crystals / molecules.

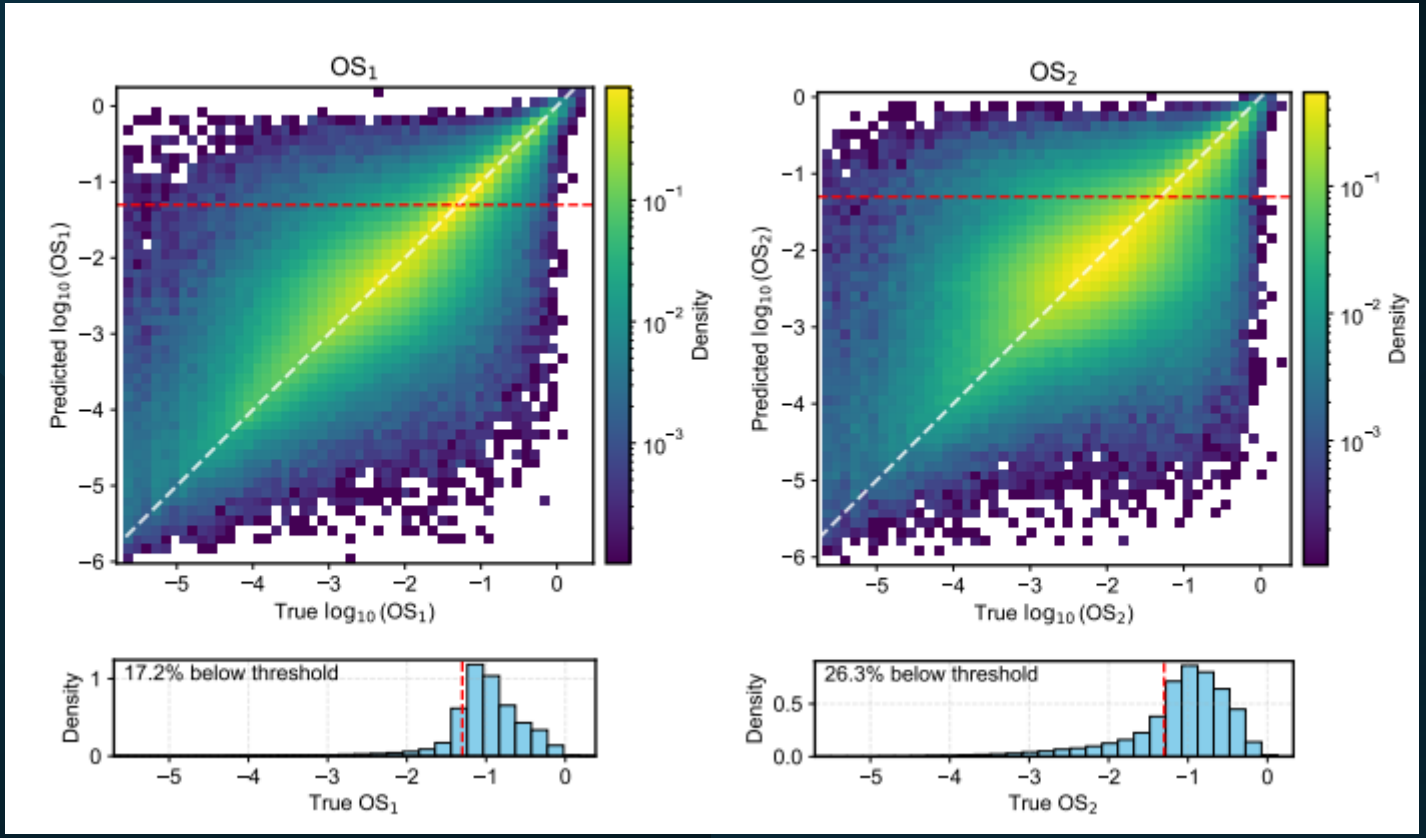
- Other applications

1. Medicine

MLP performance



MLP performance



MLP performance