# AUTOENCODERS FOR MOLECULAR DESIGN

---

Logan Ward
Asst. Computational Scientist
Argonne National Laboratory

17 February 2022

# Chemical space is enormous

*Searching through even small fraction of "all molecules" is impractical*

1.6 x $10^{11}$ molecules

Wide variety in functionalities

"Known molecules" small fraction

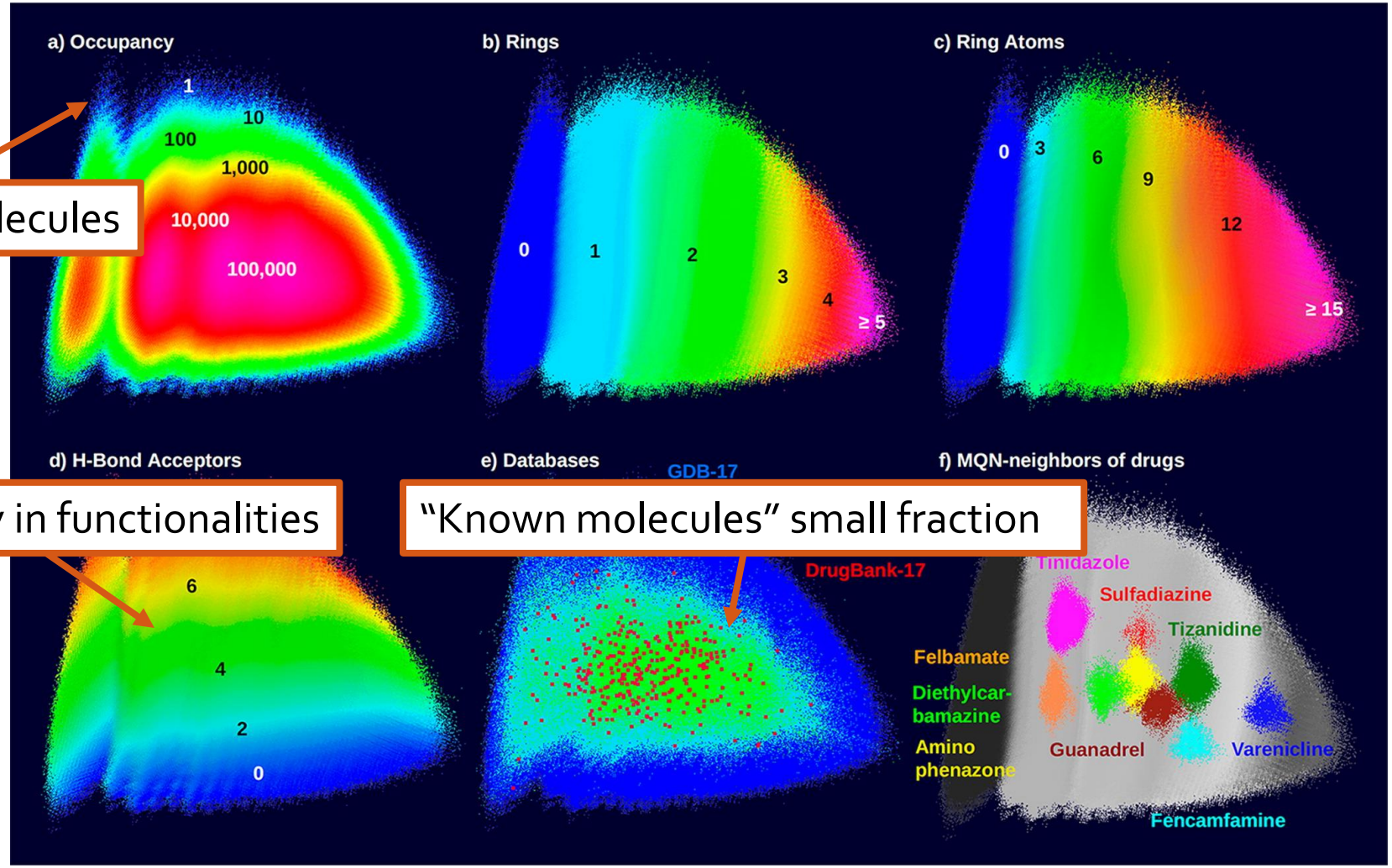
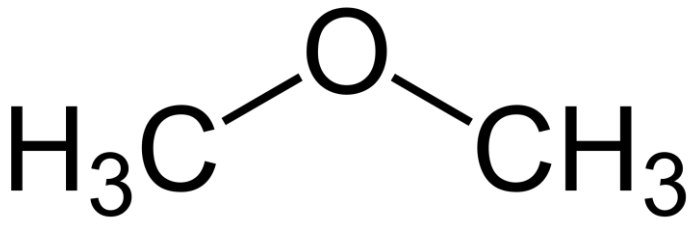



Fig: Ruddigkeit et al. JCIM. (2013)

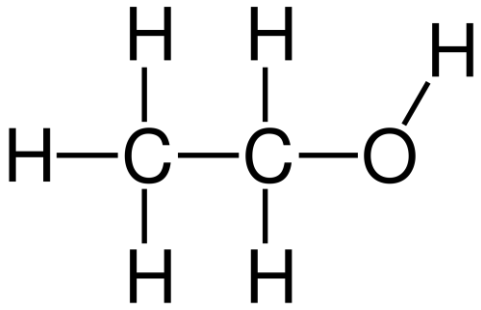
*We need a way to sample through space efficiently*

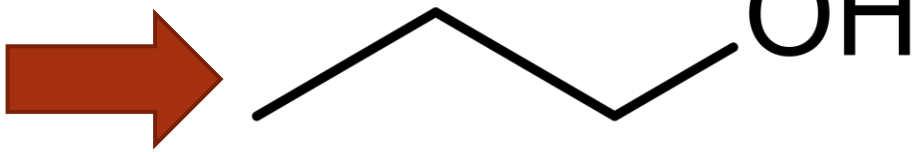
# What if I just want to search nearest neighbors?

Pick a solvent that could replace ethanol!




*Same composition,*
*very different behavior*




*Different composition,*
*similar behavior*

We need complicated rules to tell us how to navigate chemical space

**Today's topic:** What if machine learning could just "invent" new molecules for us?

# Generating data with machine learning

**Autoencoders:** Data to coordinates

**GANs:** Noise to meaningful data

**Autoregression:** Fill in missing bits

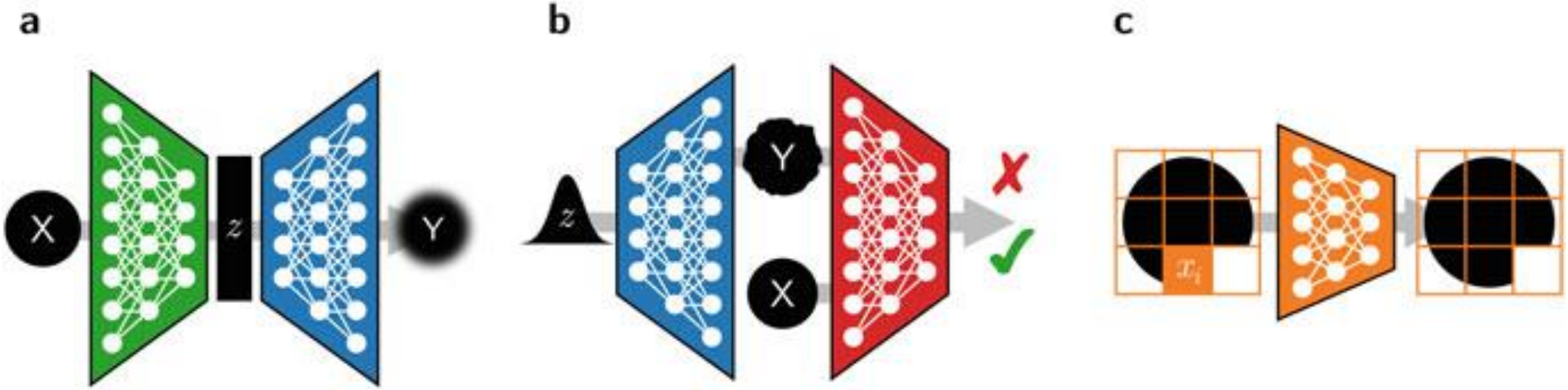
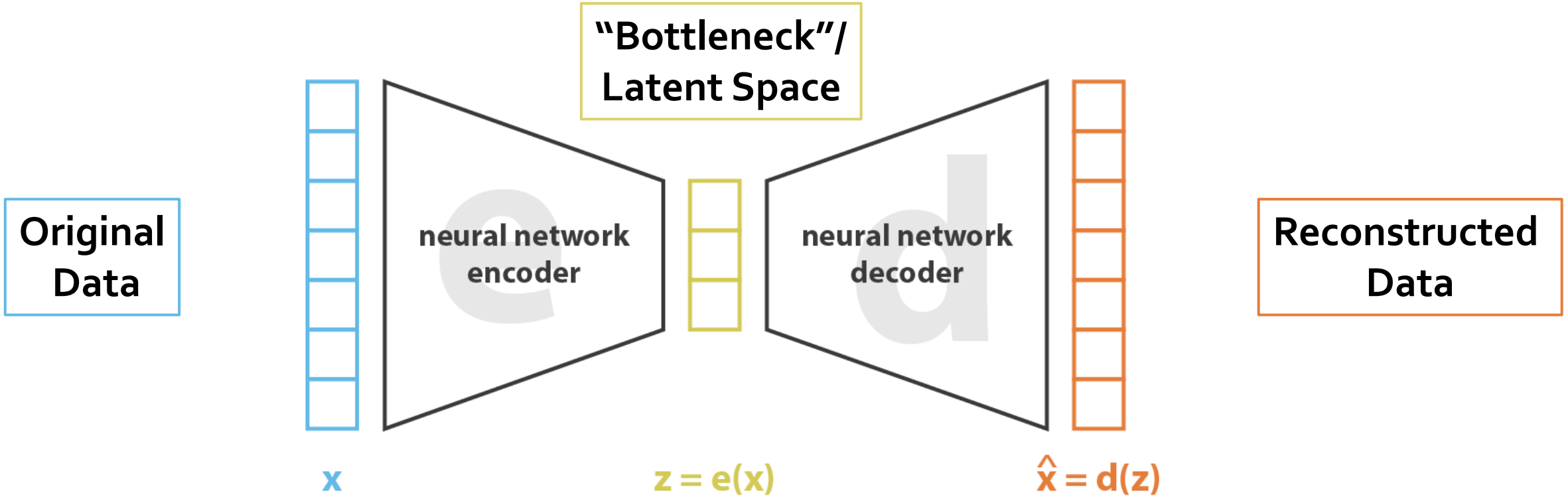



**Fig:** Schwalbe-Koda and Gómez-Bombarelli (2019) – Part of an excellent textbook

# AUTOENCODERS

(Because we are doing GANs later)

# Simple autoencoder



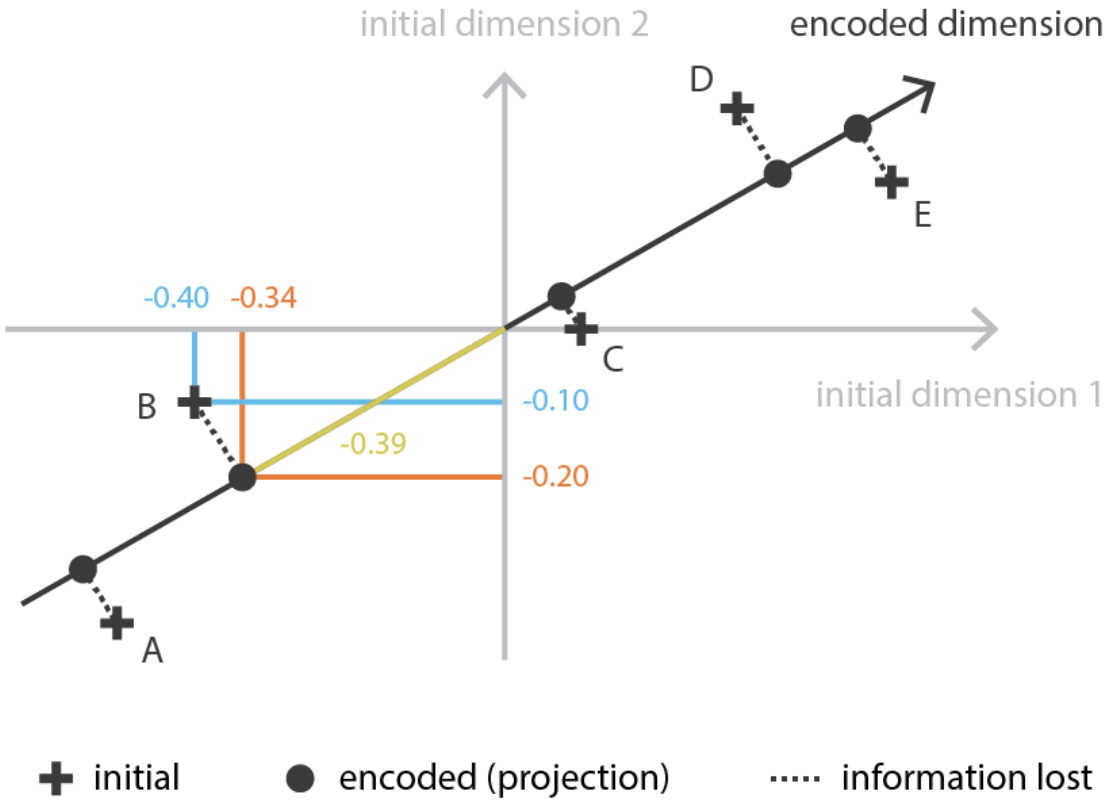

$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

Autoencoders learn a simple, *continuous* representation that captures higher-dimensional data

Ref: Joseph Rocca's excellent blog

# PCA: Autoencoder without neural networks

PCA learns a simple, *continuous* representation that captures higher-dimensional data

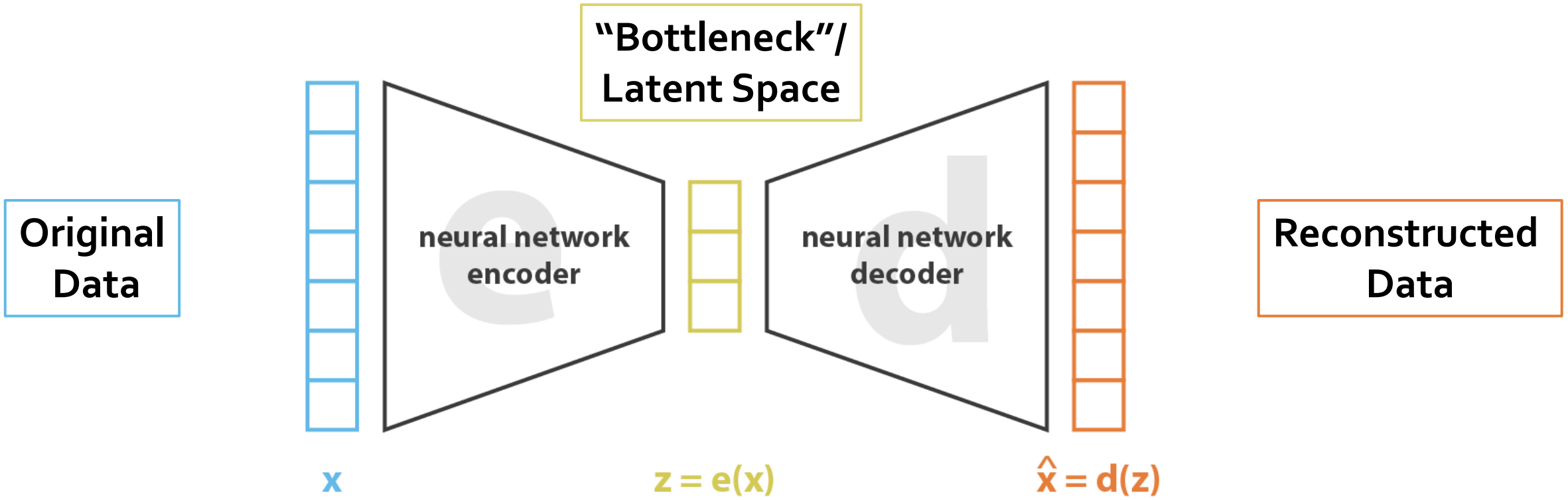



| Point | Initial | Encoded | Decoded |
|---|---|---|---|
| A | (-0.50, -0.40) | -0.63 | (-0.54, -0.33) |
| B | (-0.40, -0.10) | -0.39 | (-0.34, -0.20) |
| C | (0.10, 0.00) | 0.09 | (0.07 0.04) |
| D | (0.30, 0.30) | 0.41 | (0.35, 0.21) |
| E | (0.50, 0.20) | 0.53 | (0.46, 0.27) |

Autoencoders can learn this for even more complex data

Ref: Joseph Rocca's excellent blog

# Simple autoencoder

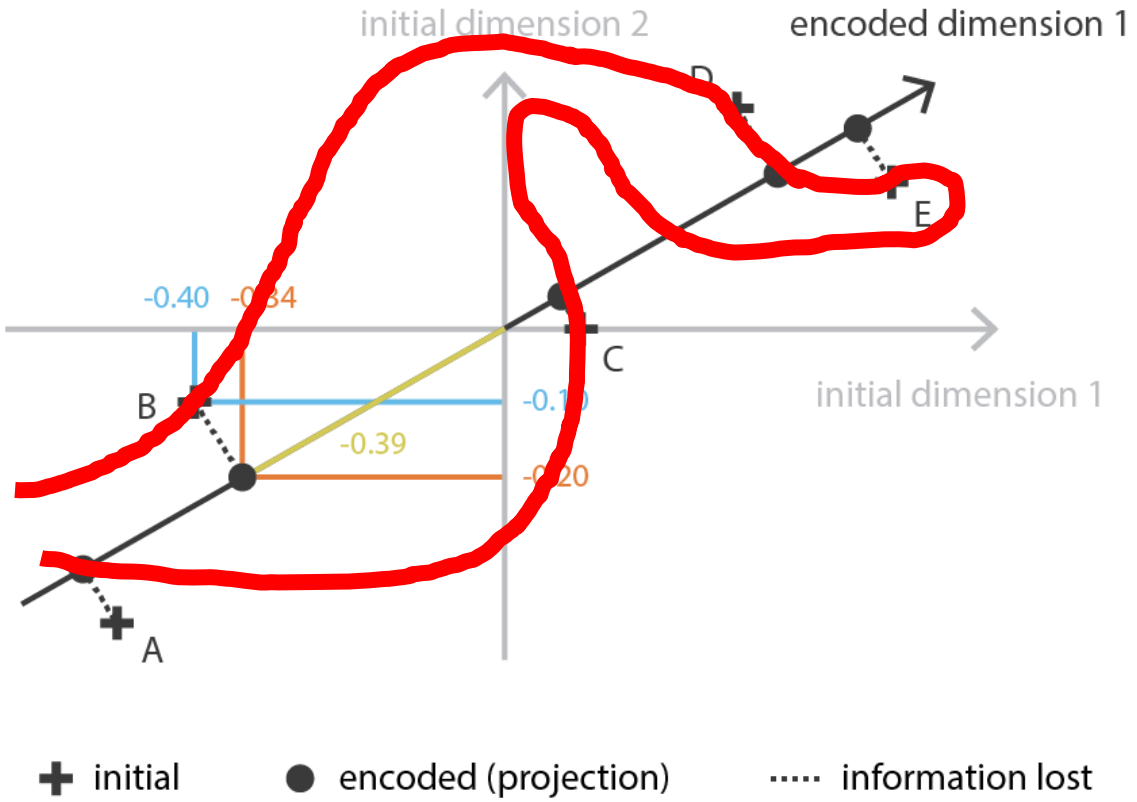



$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

What makes sure this compressed space ($z$) makes sense?

Ref: Joseph Rocca's excellent blog

# Given enough complexity, I can learn any decoding

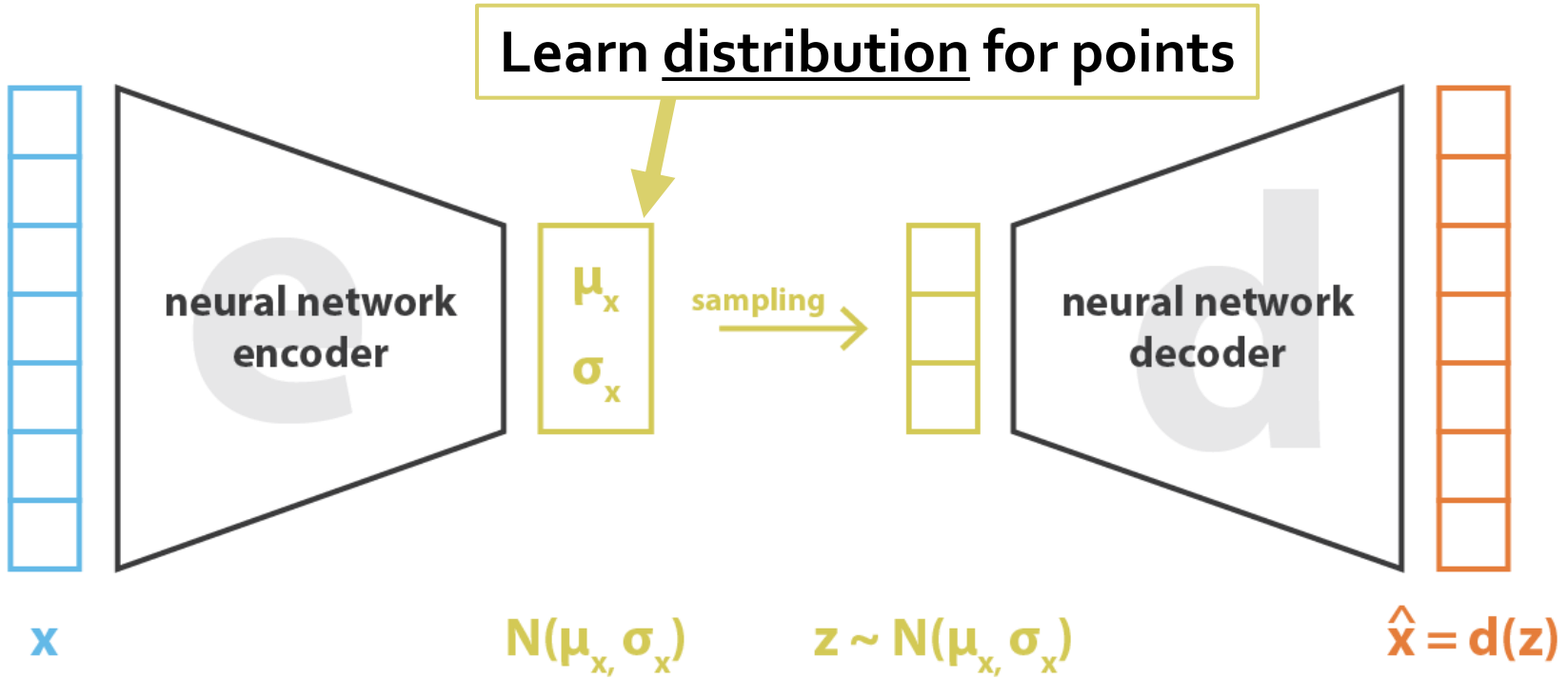



… but I don't want *any* latent space.

I want one with meaning, where:
similar points have similar *encodings*

Ref: Joseph Rocca's excellent blog

# Variational autoencoder

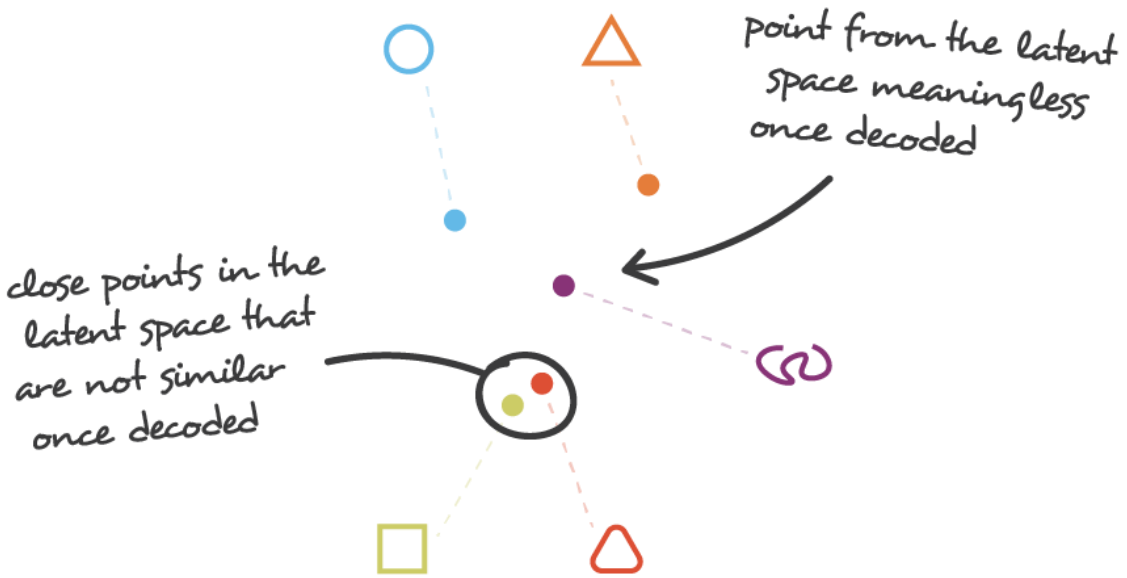



$$\text{loss} = \|x - \hat{x}\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)]$$

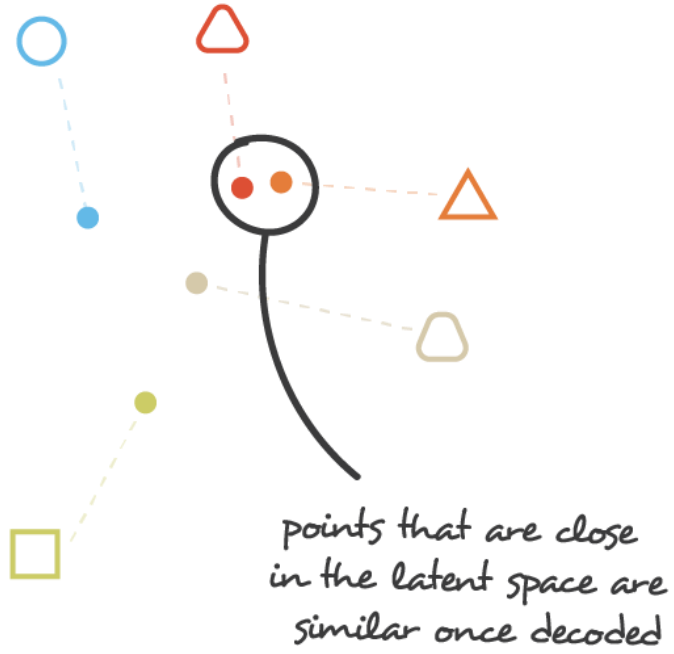
Train distribution to be like a normal distribution (KL is distance for distributions)

# Using point distributions

Predicting distributions helps ensure nearby encoded points decode to ~same data




irregular latent space

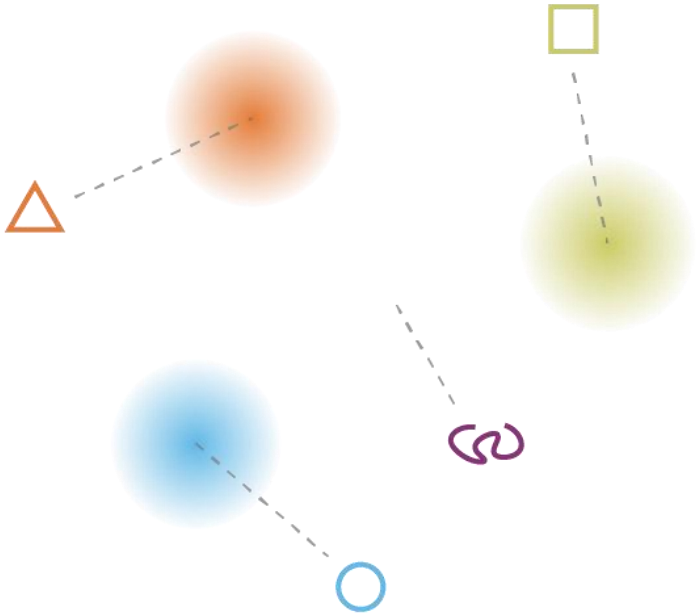



regular latent space

Ref: Joseph Rocca's excellent blog

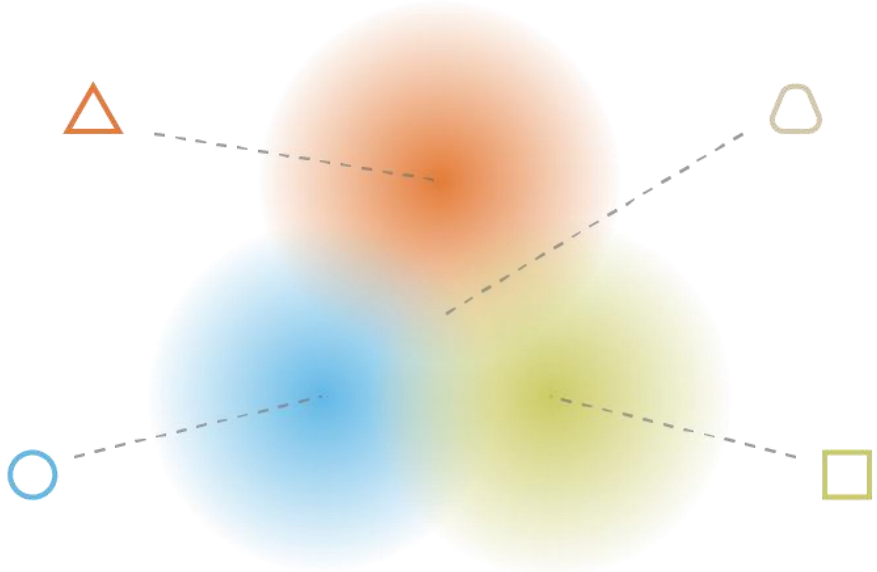
# Benefits of regularization (KL Divergence)

Without regularization, encodings can be anywhere…



what can happen without regularisation

… KL-divergence forces them to be near



what we want to obtain with regularisation

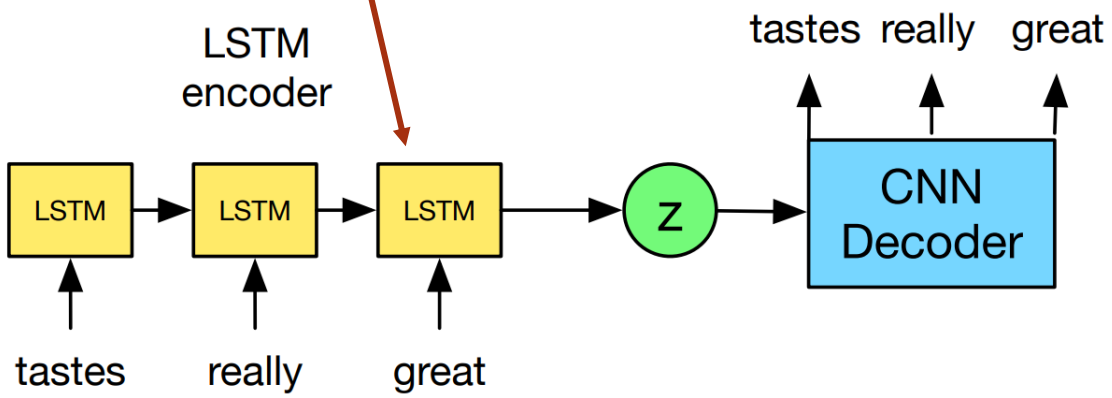
KL encourages center to be close to zero

$$KL[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

Ref: Joseph Rocca's excellent blog

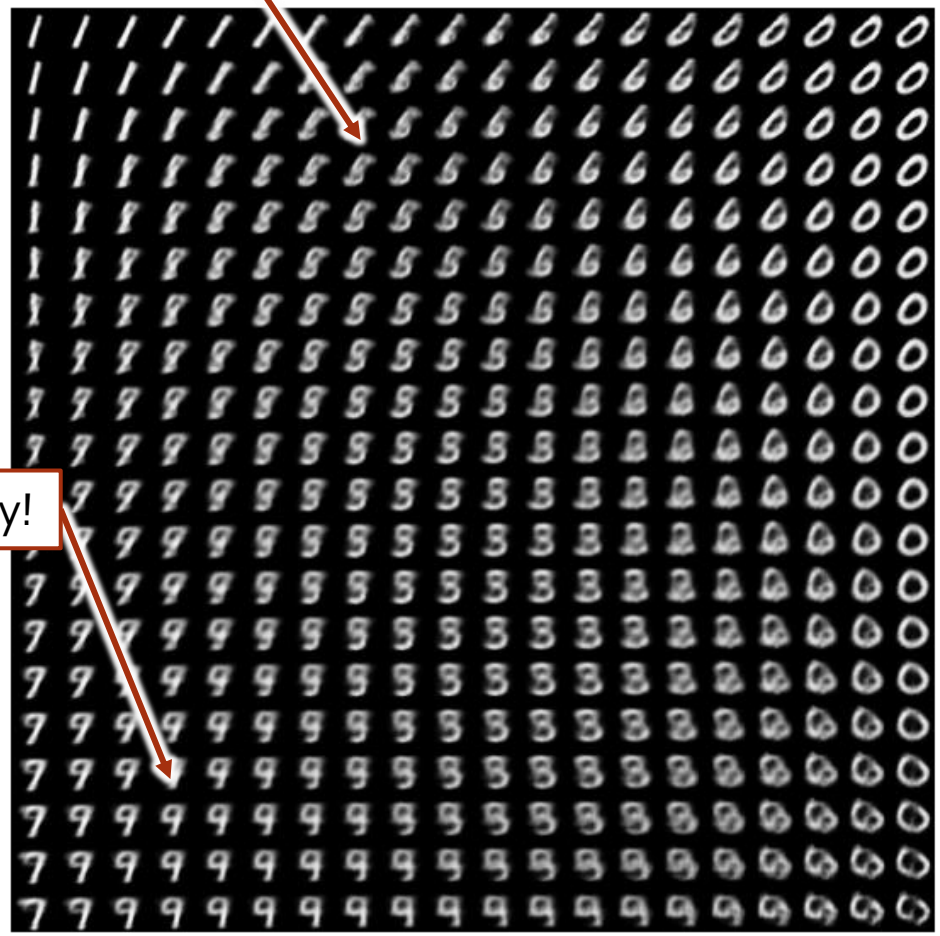
# Autoencoders for anything!

LSTMs deal with sequences, like text!




Ref: Yang et al. ICML (2017)

Use convolutions for image data



Note how similar digits are nearby!

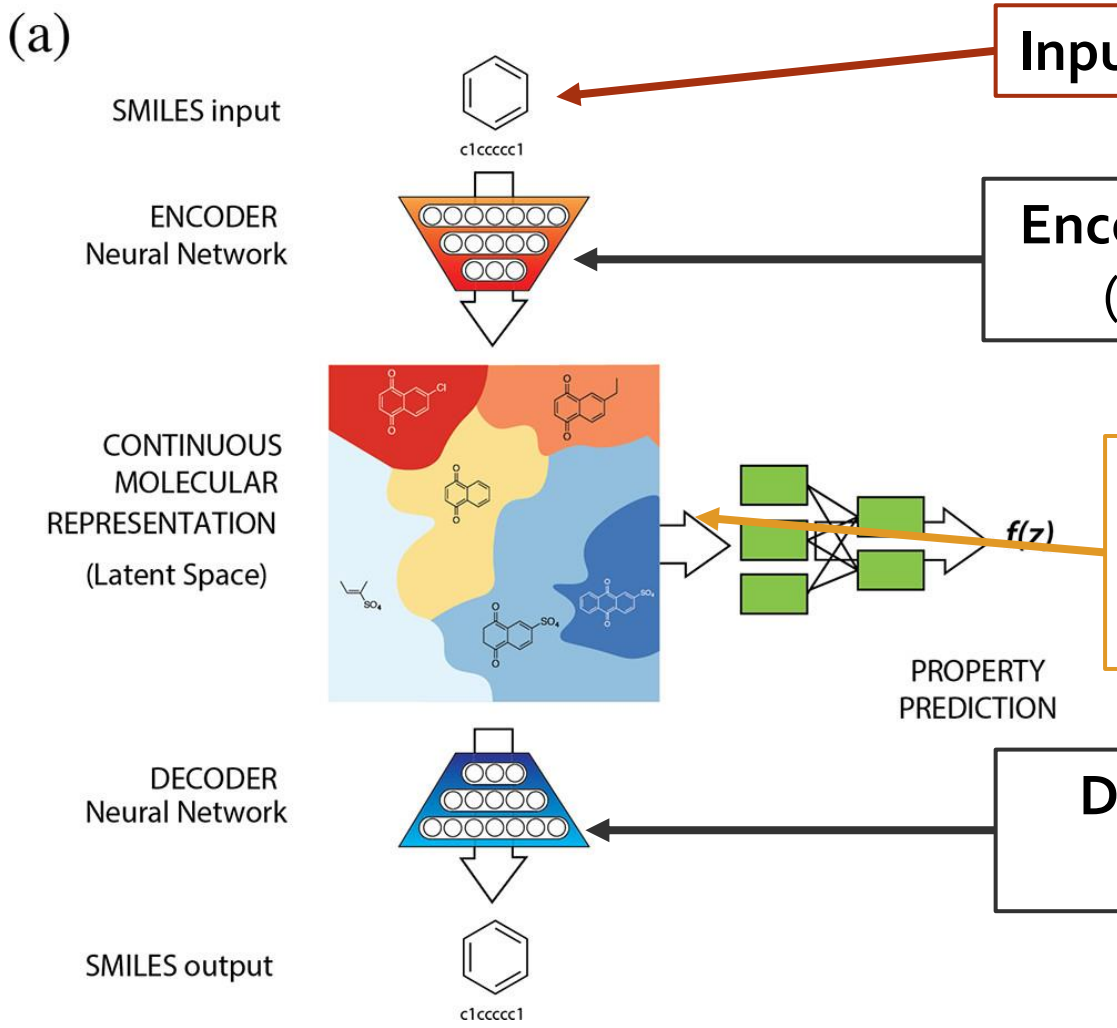
Ref: TF Documentation

**Your challenge is finding a good encoder/decoder function**

More fun with generative models: https://aiweirdness.com/

# EXAMPLES OF AUTOENCODERS FOR CHEMISTRY

# Encoders and decoders of molecules




**Input/Output:** Smiles String

**Encoder:** Convolutional layers (account for adjacency)

**Latent Space:** Trained using

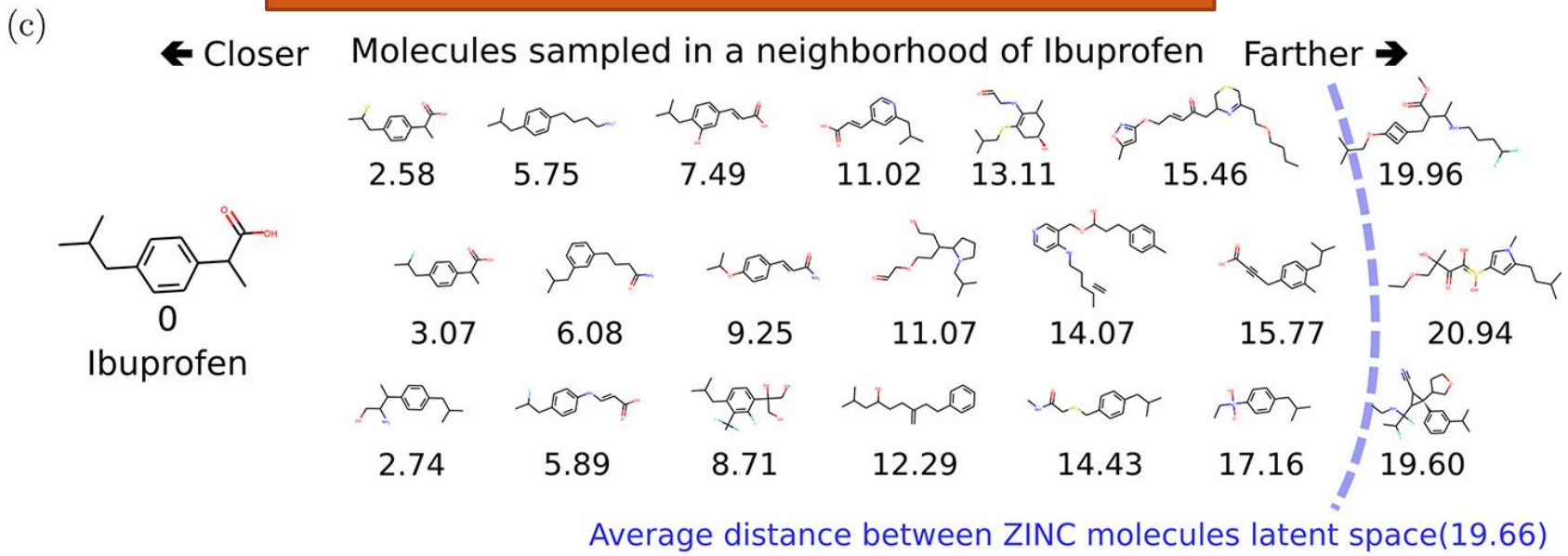
1. Variational losses (good spatial properties)
2. Supervised losses (space should *have meaning*)

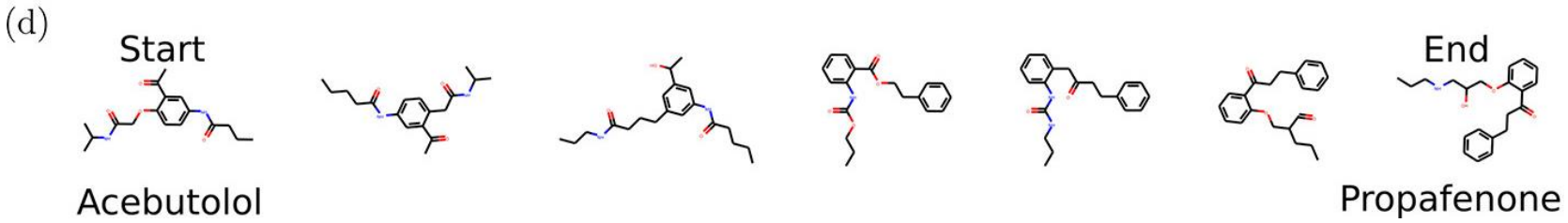
**Decoder:** Recurrent layers (generate sequences)

Ref: Gómez-Bombarelli et al. ACS Cent. Sci. (2018)

# And it works nicely!

Nearby molecules have similar functional groups!




Can express interpolations between different molecules

Ref: Gómez-Bombarelli et al. ACS Cent. Sci. (2018)

# Breaking down the encoder

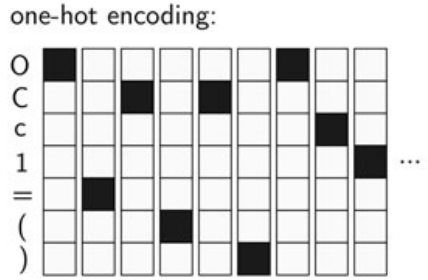
A few good points to know about training this VAE

- Strings are broken into tokens

  *(Vocabulary is limited to observed compounds)*

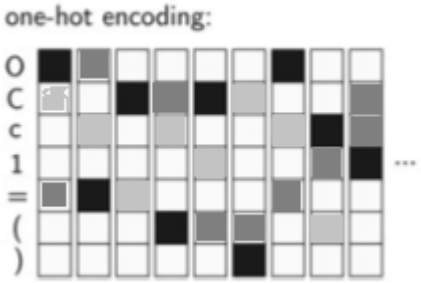
string:

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




- The output is probabilistic

  *(More than one option per decoding)*




*(Trained to maximize log-likelihood of original input)*

$$\mathcal{L}(\theta,\phi) = -D_{KL}(Q_\phi(z|X)||P_z(z)) + \mathbb{E}_{z\sim Q_\phi}[\log P_\theta(X|z)]$$

Variational loss

Reconstruction loss

# The problem: No guarantee of validity

*Table 1.* Reconstruction accuracy and prior validity results. Baseline results are copied from Kusner et al. (2017); Dai et al. (2018); Simonovsky & Komodakis (2018); Li et al. (2018).

| Method | Reconstruction | Validity |
|---|---|---|
| CVAE | 44.6% | 0.7% |
| GVAE | 53.7% | 7.2% |
| SD-VAE | 76.2% | 43.5% |
| GraphVAE | - | 13.5% |
| Atom-by-Atom LSTM | - | 89.2% |
| JT-VAE | **76.7%** | **100.0%** |

**Big problem:** You must decode many times to get a valid molecule

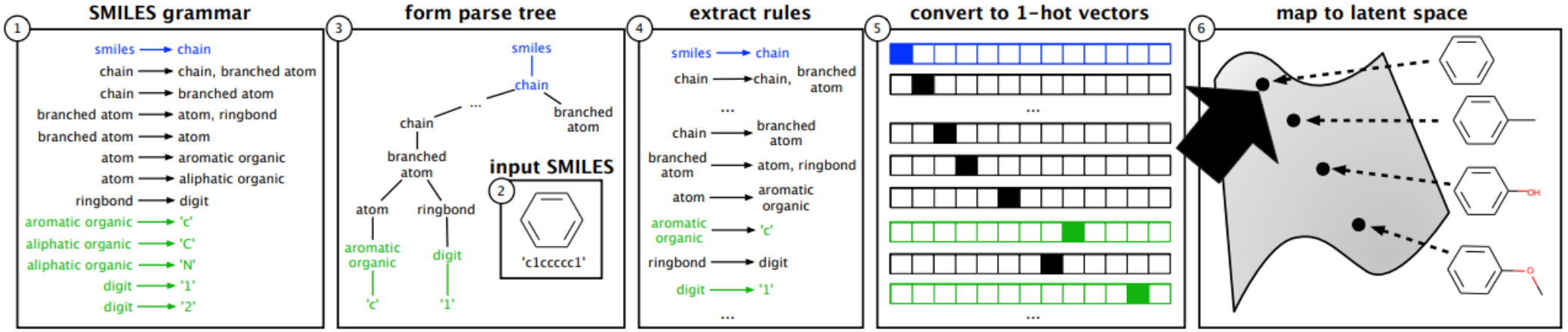
**Ex:** CC=O is valid, CO=O is not

**Newer methods improve reconstruction by encouraging/enforcing validity**

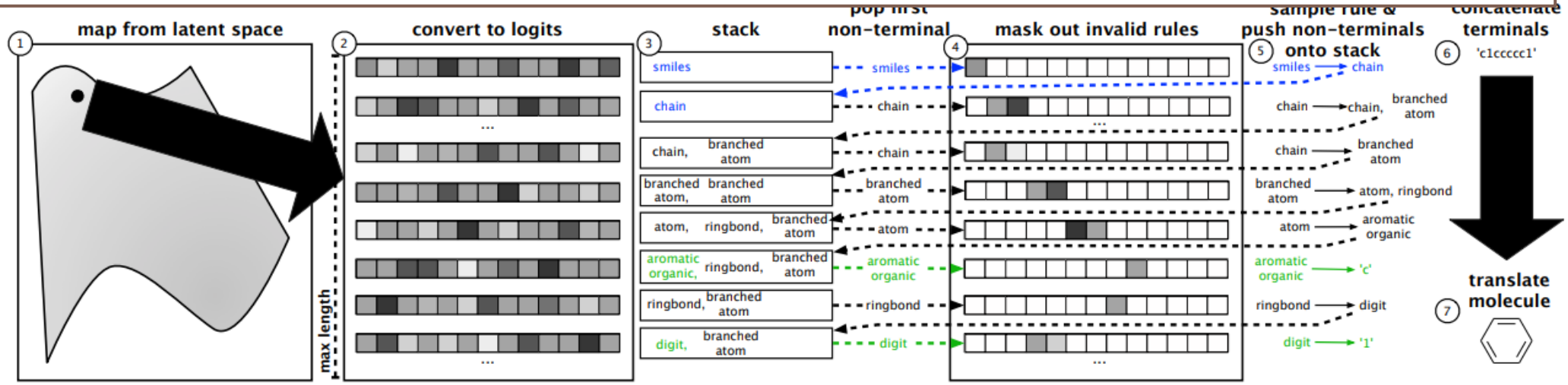
Table from Jin et al. (2019)

# Improve by enforcing grammar (Kusner (2017))

**Encoder:** Given grammar, parse molecule into a series of rules, encode as digits, compress rules to a latent space

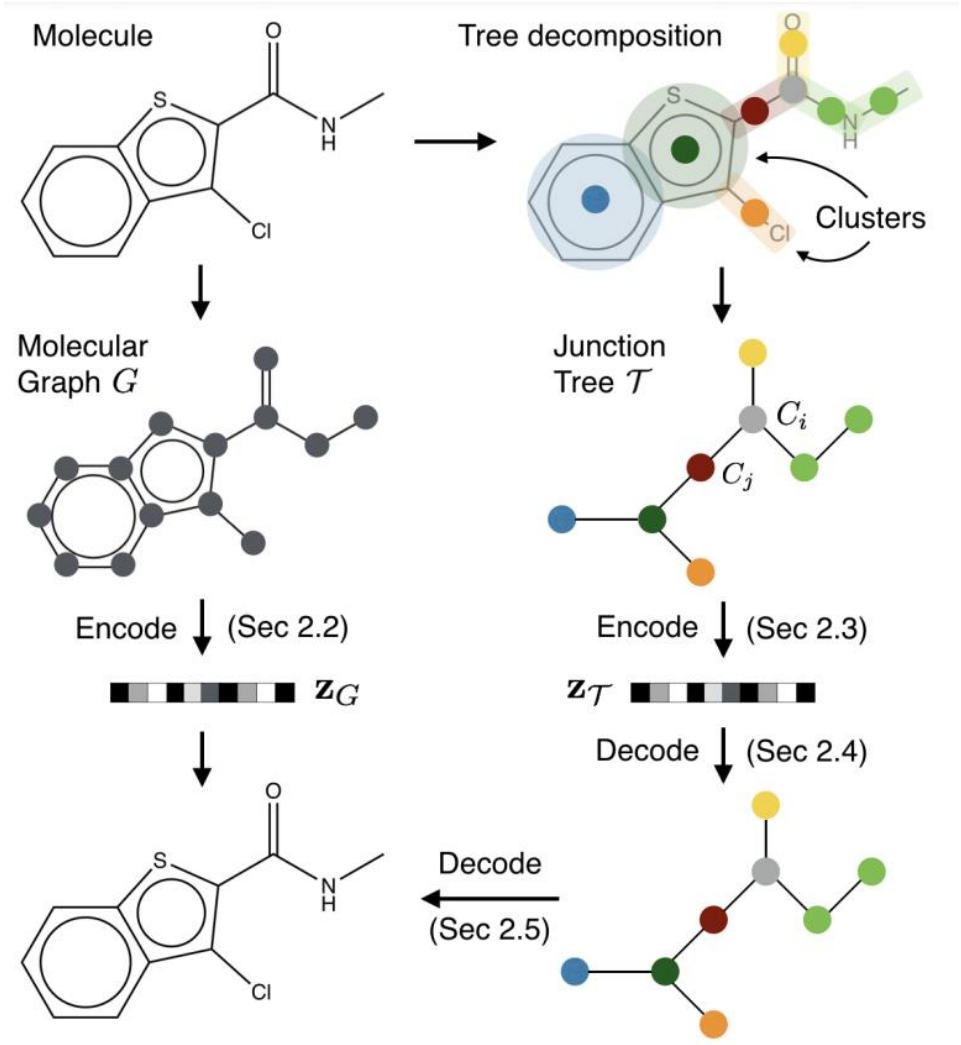



**Decoder:** Given latent space, decompress to rule probabilities, sample rules (enforcing grammar!), rebuild

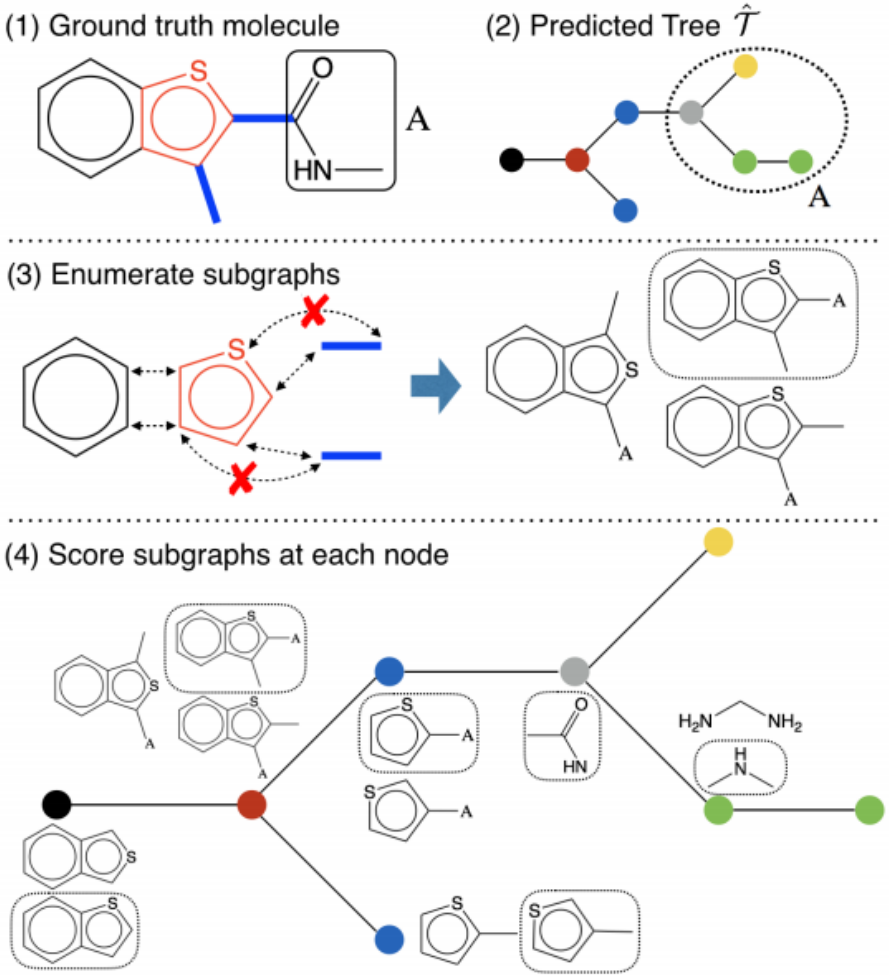


# Building up graphs: Junction-Tree VAE (Jin 2019)

Encode with message-passing networks

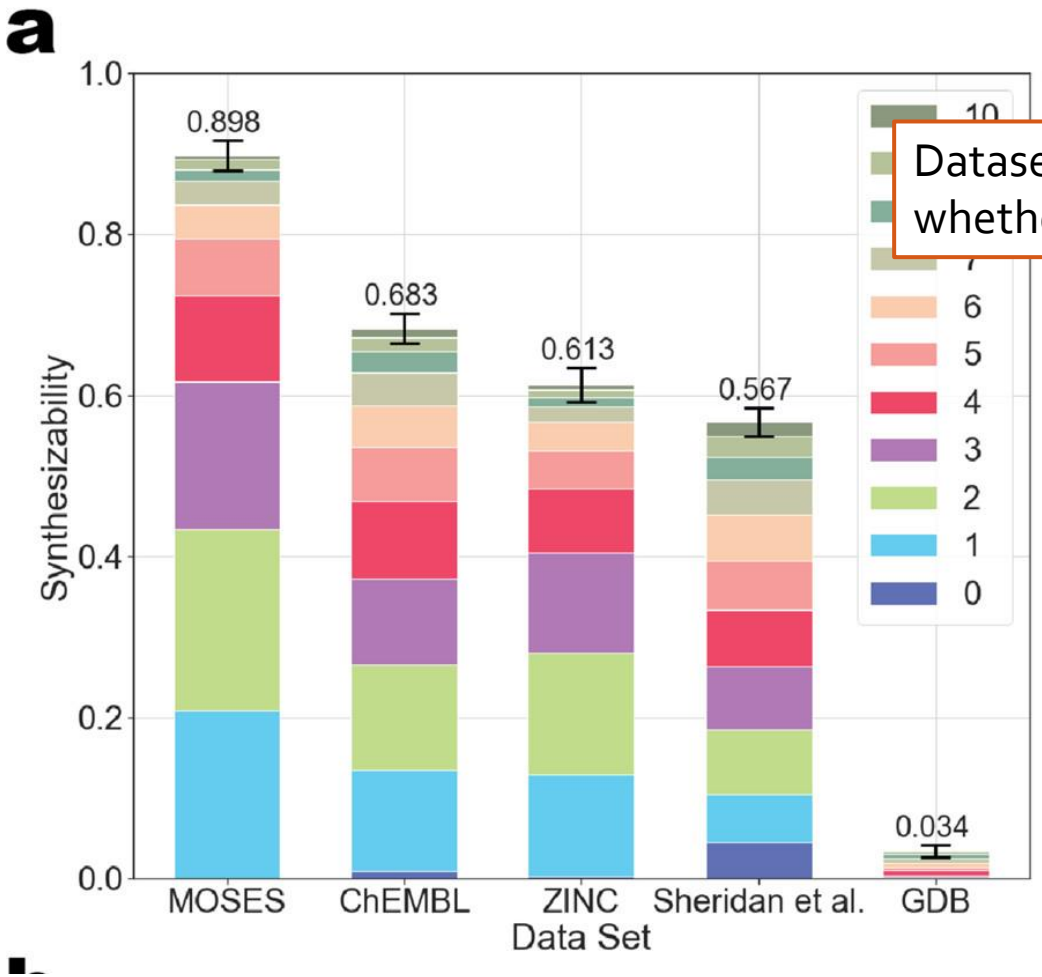



Decoding is complex.
Ex: picking between possible coarse-to-fine



# Last note: "Valid" does not mean "synthesizable"

**On-going issue:**
How to predict synthesizability?

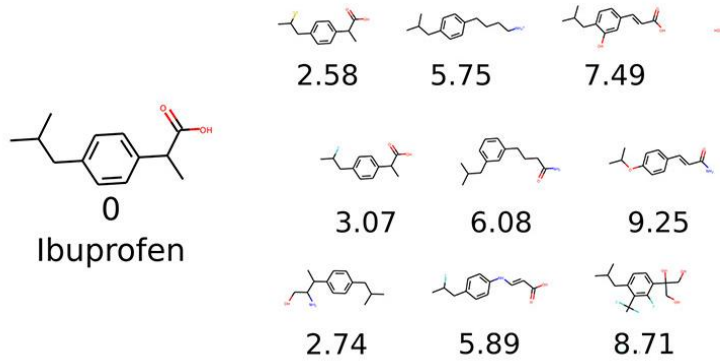



Dataset sizes can have big effects whether molecules are synthesizable

**Fig:** Gao et al. JCIM (2020)

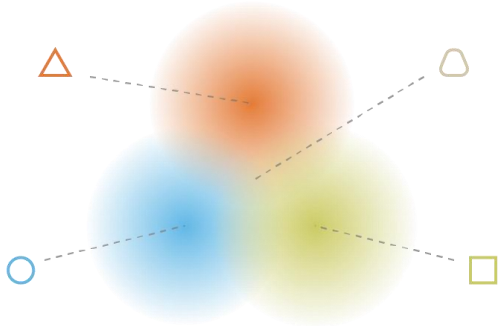
# Take Home Points

- **Why choose VAEs?**
  Easily generate new molecules



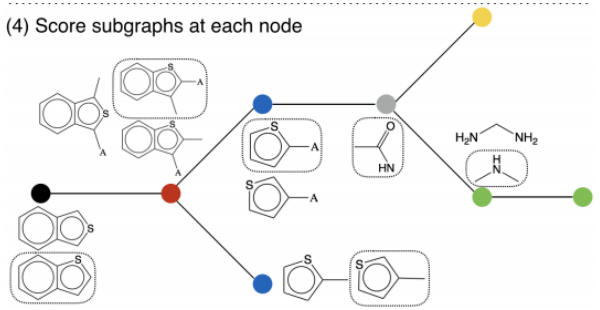

Ref: Gómez-Bombarelli et al. ACS Cent. Sci. (2018)

- **How do VAEs work?**
  Enforcing meaningful distances on latent space



Ref: Joseph Rocca's excellent blog

- **How to make better VAEs?**
  Design better encoders / decoders




Ref: Jin et al. 2019