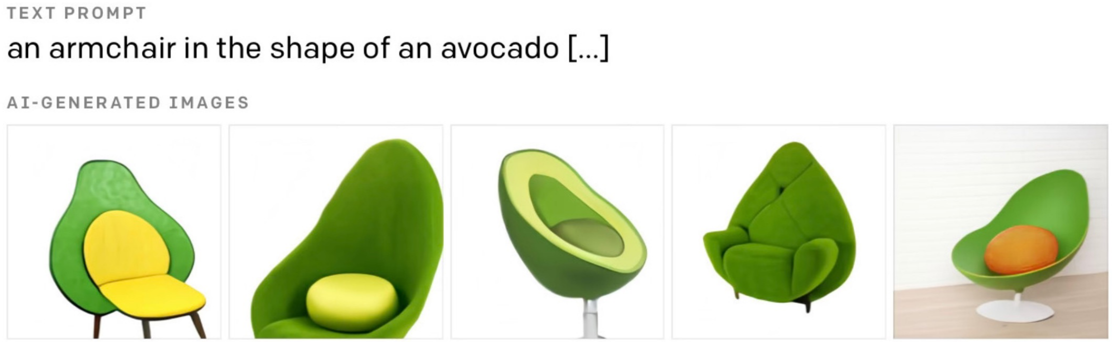
# Intelligent Agents
## Language-Vision-Models: DALL-E

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

# Generate Images from Text – Naïve Approach



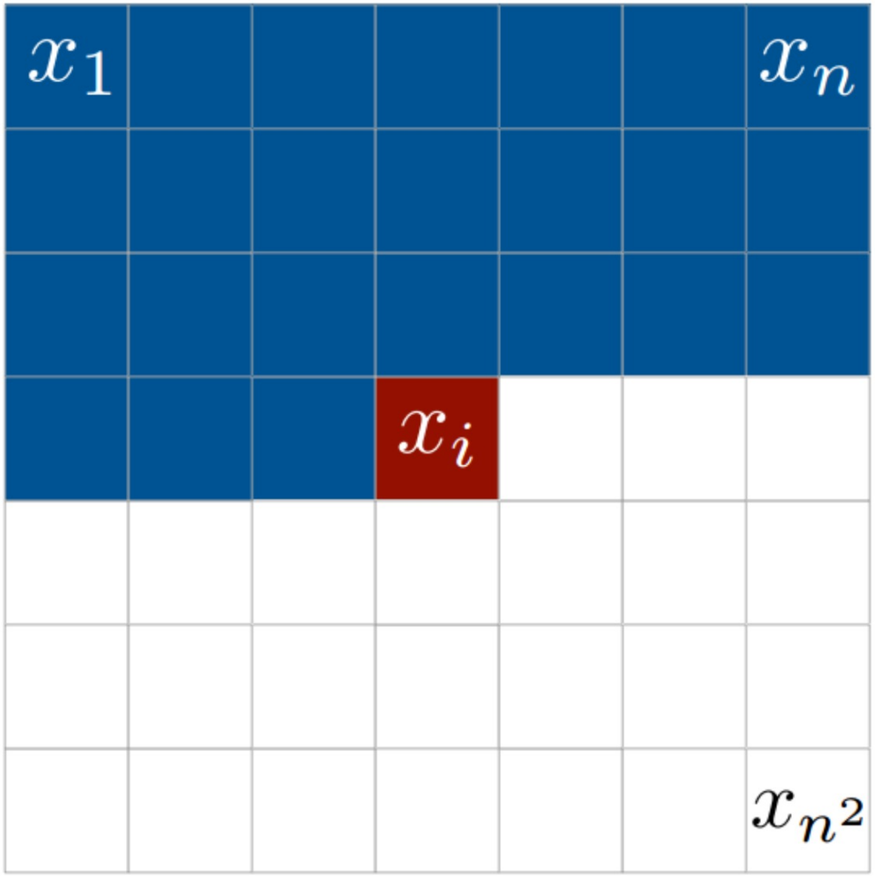

1. Concatenate the set of text tokens with the unrolled set of pixel values in a corresponding image (typically unrolled top left to bottom right).
2. Given this sequence of text and pixel values, we can factor the distribution $p(x|y)$ autoregressively:
   $$p(x|y) = p(x1, x2, x3, ...|y) = p(x1|y)p(x2|x1, y)p(x3|x1, x2, y)...$$
   Here $xi$ is the $i$th pixel value in the unrolled image.
3. We now estimate $p(x|y)$ by running maximum likelihood estimation on any autoregressive sequence model (e.g. LSTM or Transformer) over each of these $p(xi|xi{-}1, xi{-}2, ..., x2, x1, y)$ factors.
   That is to say, we want to train a model to predict the next pixel value in an image, given some text and all previous pixel values.



How is it so good ? (DALL-E Explained Pt. 2). Charlie Snell

# Use RNN decoder to generate images??

An armchair in the shape of […]



# Generate Images from Text – Naïve Approach


1. Concatenate the set of text tokens with the unrolled set of pixel values in a corresponding image (typically unrolled top left to bottom right).
2. Given this sequence of text and pixel values, we can factor the distribution $p(x|y)$ autoregressively:

   $$p(x|y) = p(x1, x2, x3, \ldots|y) = p(x1|y)p(x2|x1, y)p(x3|x1, x2, y)\ldots$$

   Here $xi$ is the $i$th pixel value in the unrolled image.
3. We now estimate $p(x|y)$ by running maximum likelihood estimation on any autoregressive sequence model (e.g. LSTM or Transformer) over each of these $p(xi|xi{-}1, xi{-}2, \ldots, x2, x1, y)$ factors.

   That is to say, we want to train a model to predict the next pixel value in an image, given some text and all previous pixel values.



How is it so good ? (DALL-E Explained Pt. 2). Charlie Snell

# Acknowledgements

## Zero-Shot Text-to-Image Generation

Authors: Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever

Open AI (ICML 2021)

Presentation from: Adam Kutchak, George Lu, Fernando Treviño, and Sarah Wilson



Zero-Shot Text-to-Image Generation. Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever Proceedings of the 38th International Conference on Machine Learning, PMLR 139:8821-8831, **2021**.

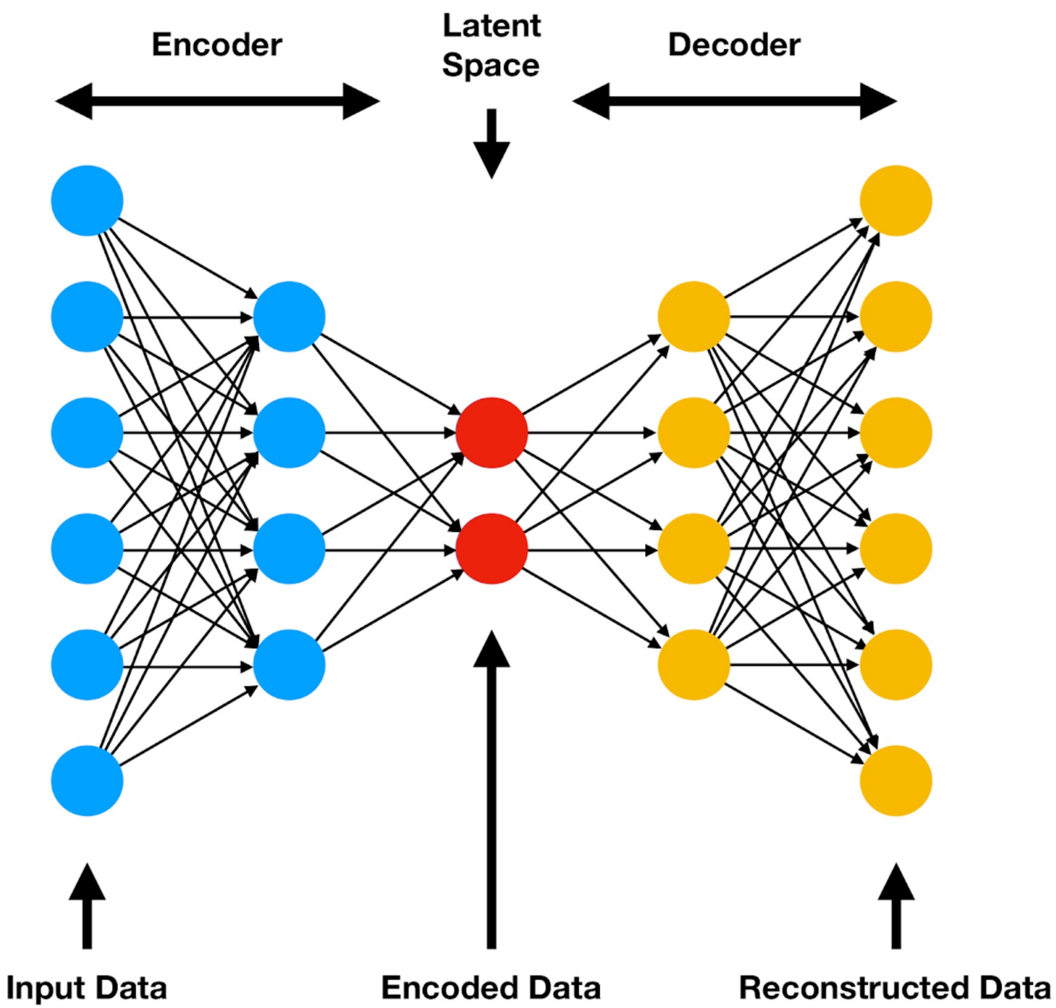
# Introduction

- Generate Images from text captions
- 12 billion parameters version of GPT-3
- Dataset comprised of 3.3 million text - image pairs
- Combine unrelated concepts

# Related Works

- Autoencoder - (encoder - decoder)
- Variational Autoencoders (continuous state space)
- Vector Quantized-Variational AutoEncoder VQ-VAE (discrete quantized state space)

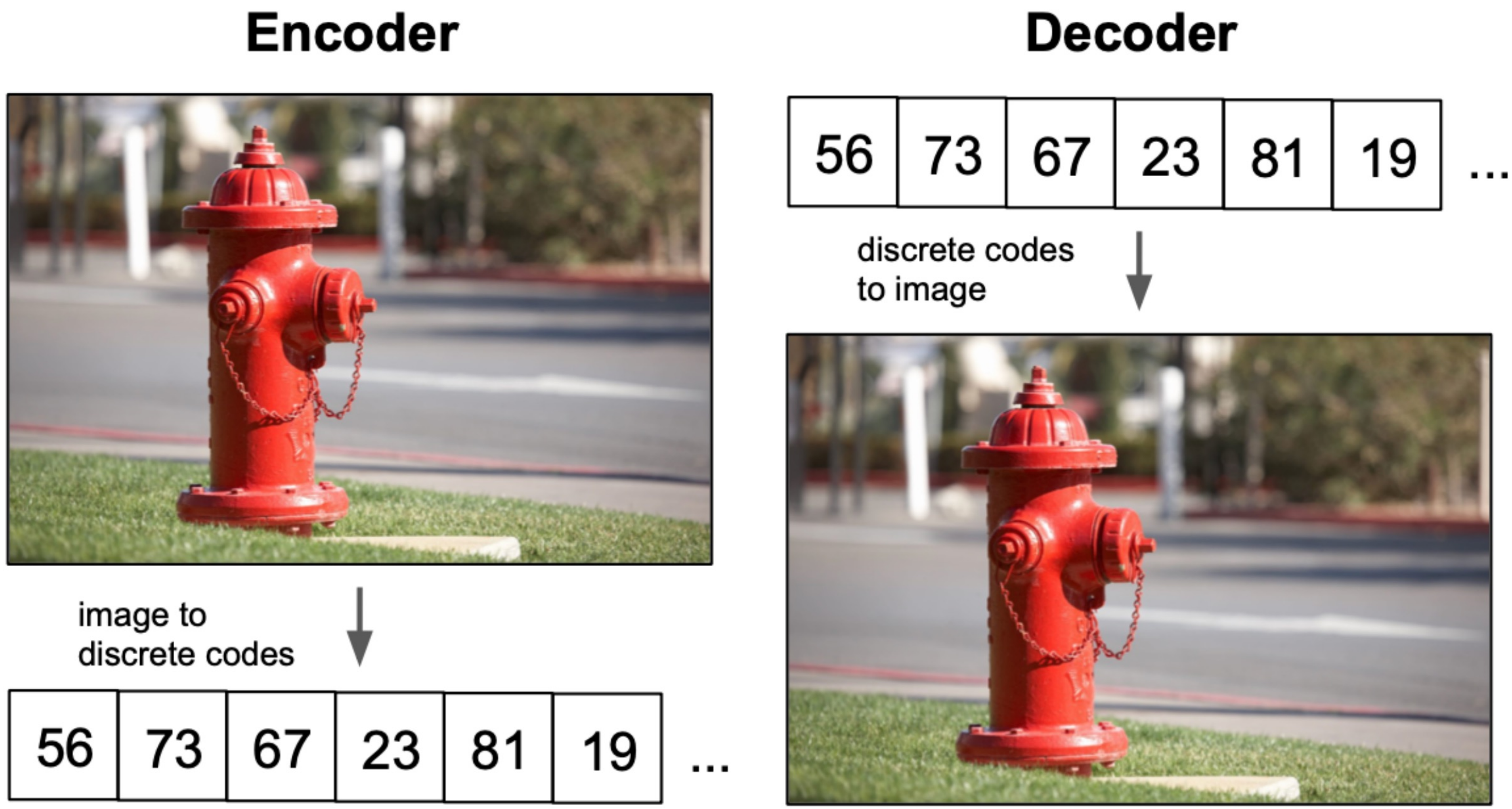
# Related Work - Autoencoder






Understanding VQ-VAE (DALL-E Explained Pt. 1)). Charlie Snell

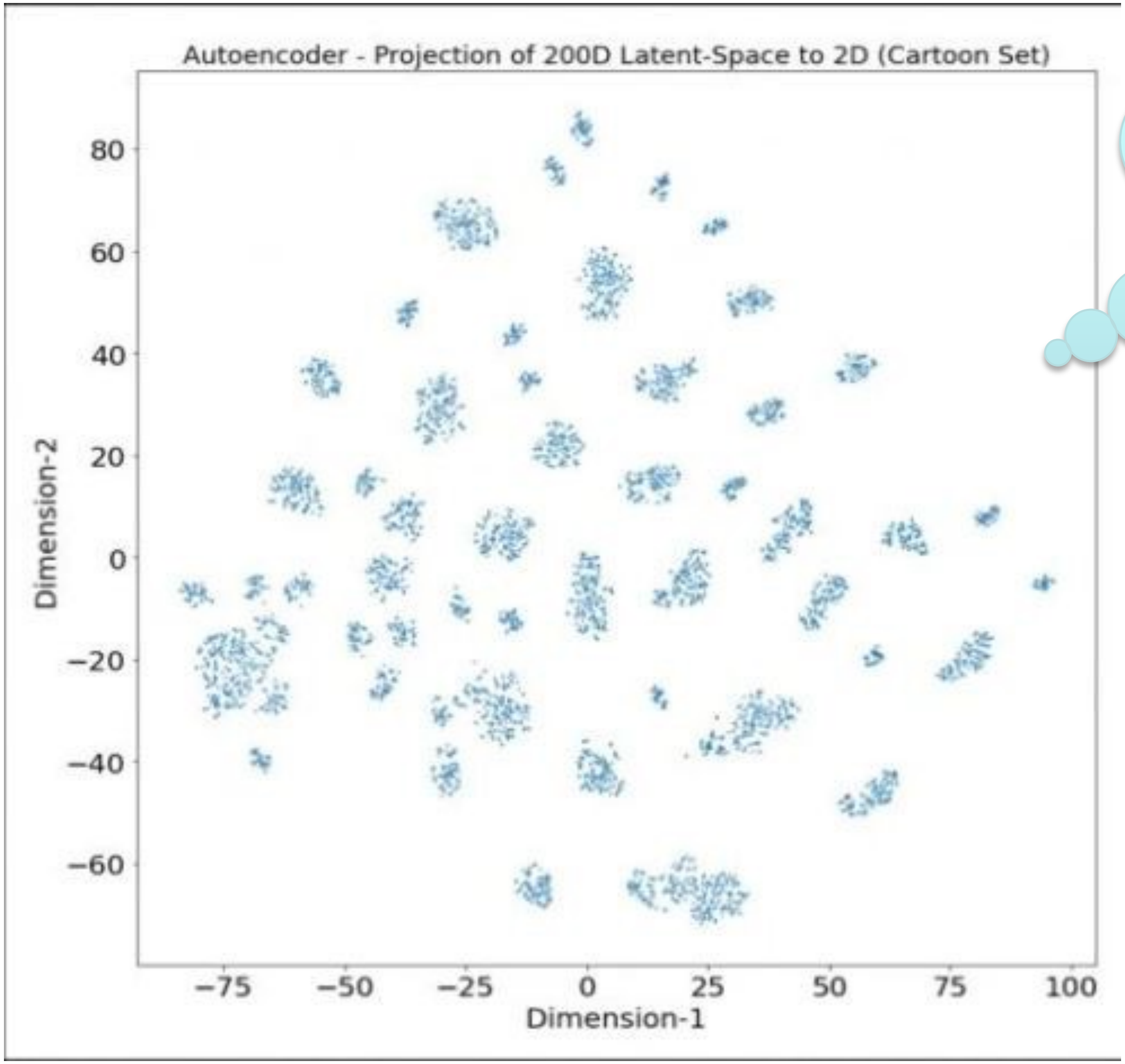
# Related Work - Autoencoder






Understanding VQ-VAE (DALL-E Explained Pt. 1)). Charlie Snell

# Related Work - Autoencoder problem




Continuous latent space, but…

# Related Work - Variational Autoencoder

- Consider our latent space $z$ as a random variable
- First let's enforce a **prior** $p(z)$ on our latents, in most VAEs this is typically just a standard gaussian distribution $\mathcal{N}(0, 1)$
- Given a raw datapoint $x$, we also define a **posterior** for the latent space as $p(z \mid x)$
- The goal is to compute this posterior for the data, which we can express using Bayes' rule as

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

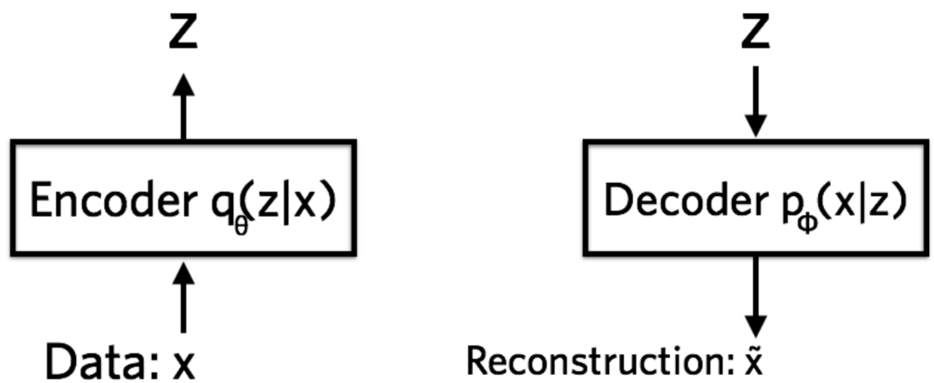
- But… $p(x)$ is intractable
- Need approximation


Understanding VQ-VAE (DALL-E Explained Pt. 1)). Charlie Snell

# Related Work - Variational Autoencoder

- Restrict approximation of the posterior to a specific family of distributions: independent gaussians. Call this approximated distribution $q(z|x)$

VAE: Add a prior to the autoencoder latent space: Approximate $p(z)$ with $q(z|x)$



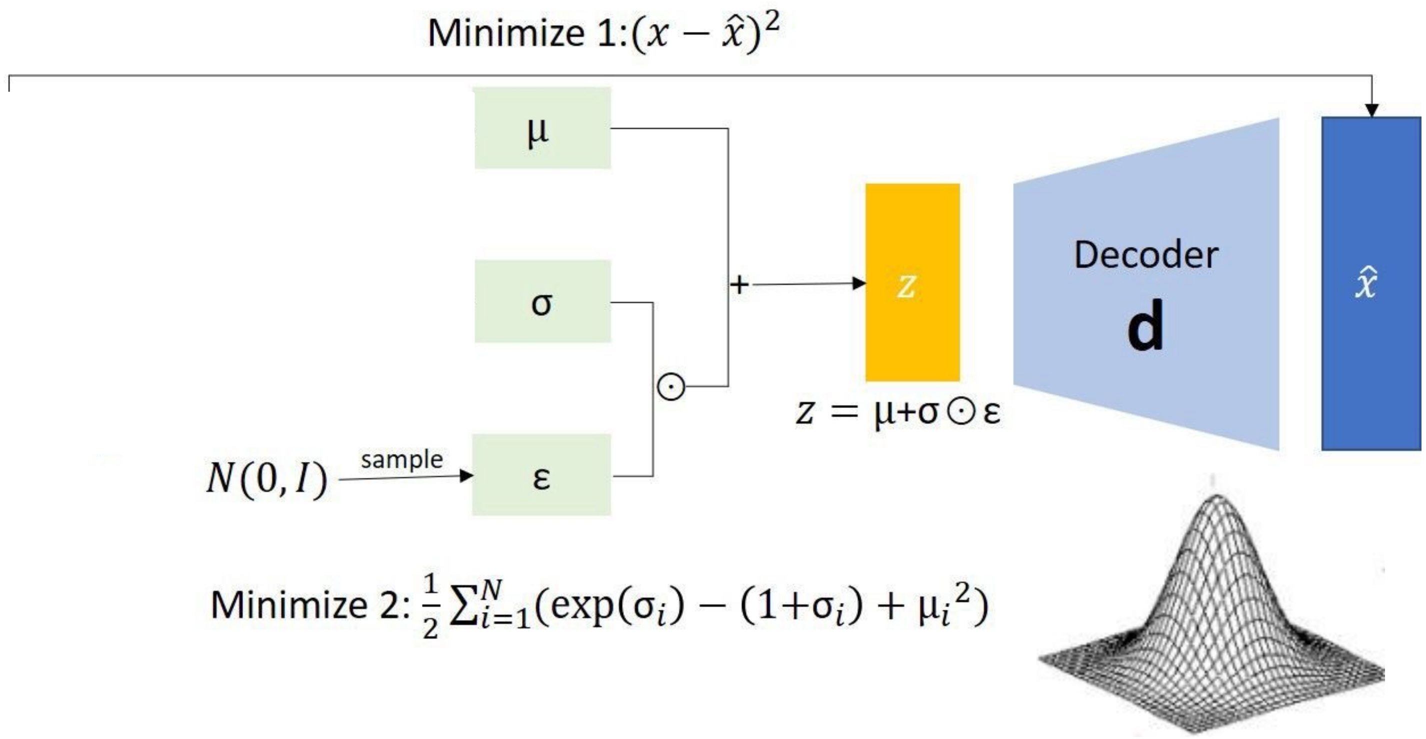

Derive loss function: $-E_{z\sim q(z|x)}[log(p(x|z))] + KL(q(z|x)||p(z))$

ELBO: See course on Probabilistic and Differential Programming

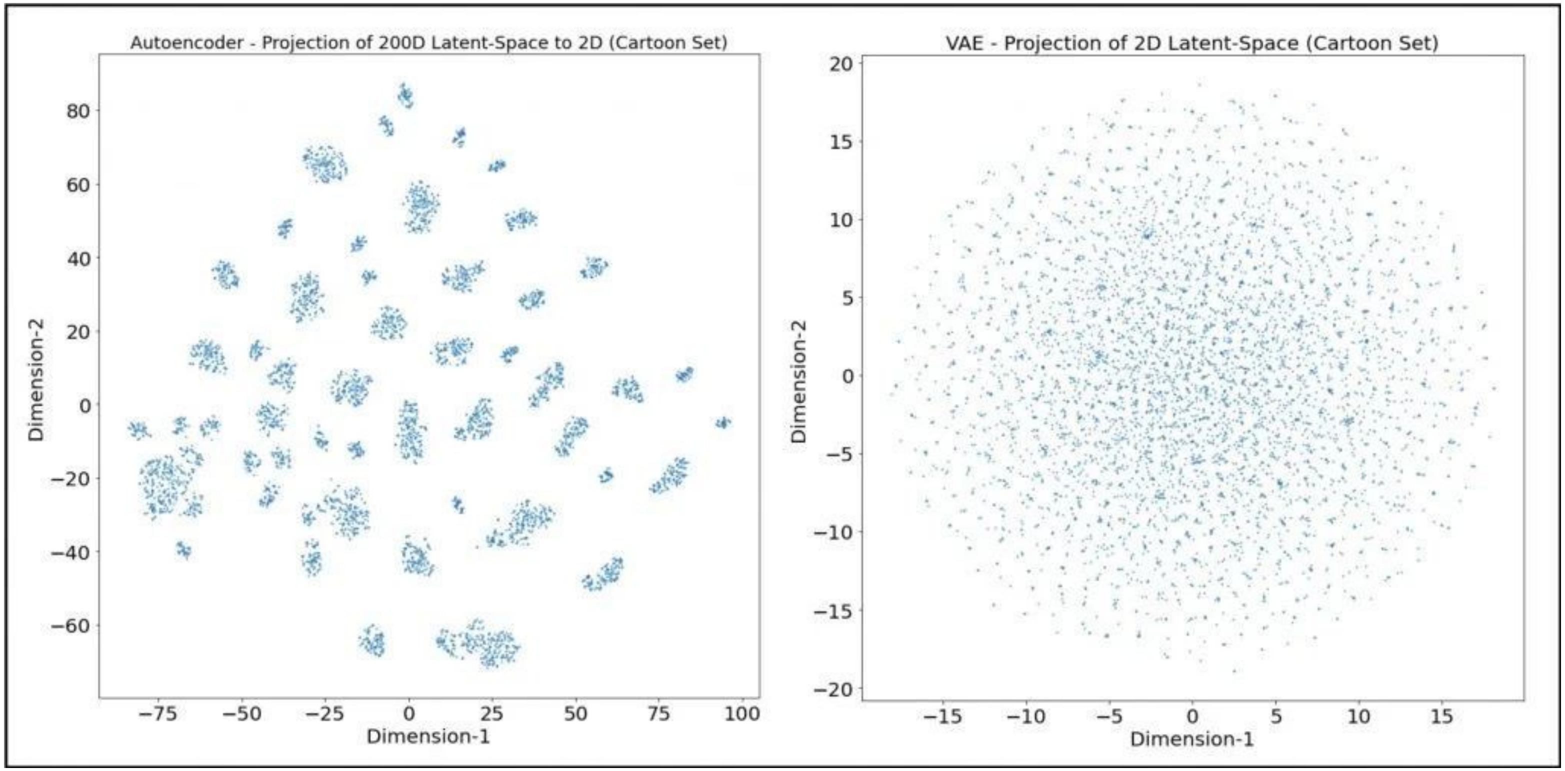


Understanding VQ-VAE (DALL-E Explained Pt. 1)). Charlie Snell

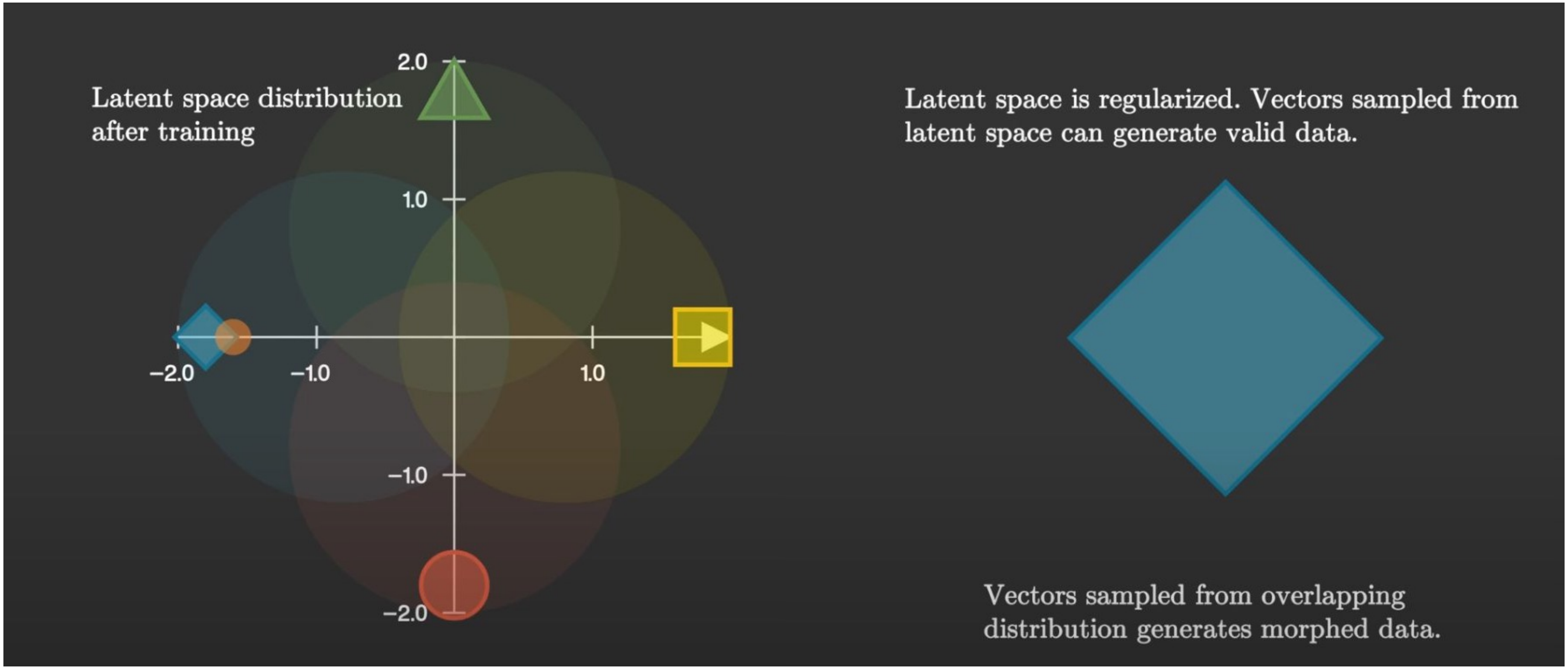
# Variational Autoencoder as a Generator



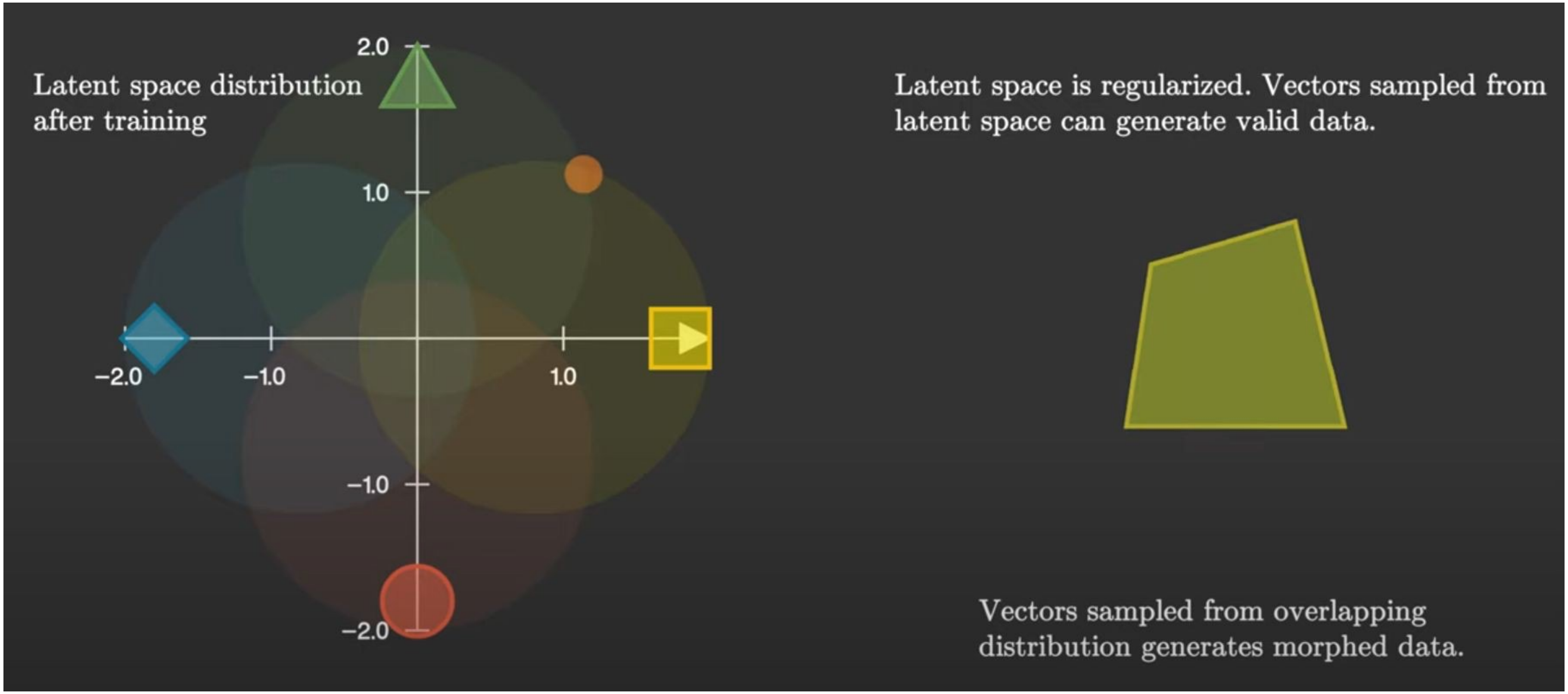
# Related Work - Autoencoder vs. VAE



# Variational Autoencoder as a Generator

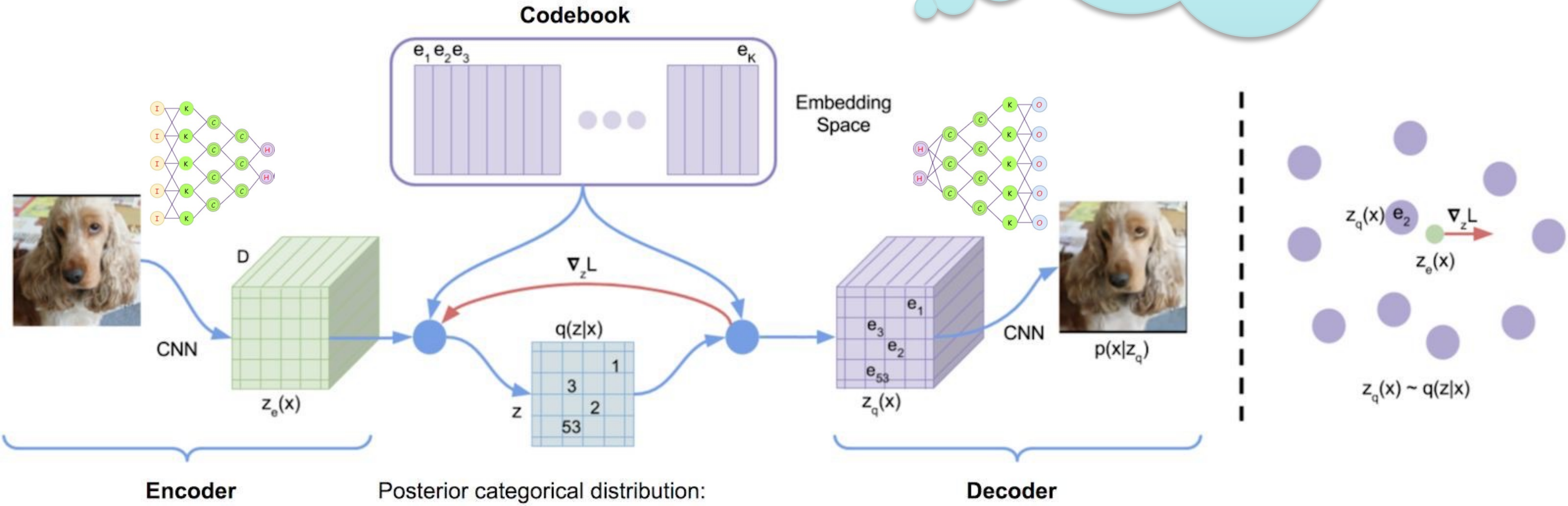


# Variational Autoencoder as a Generator

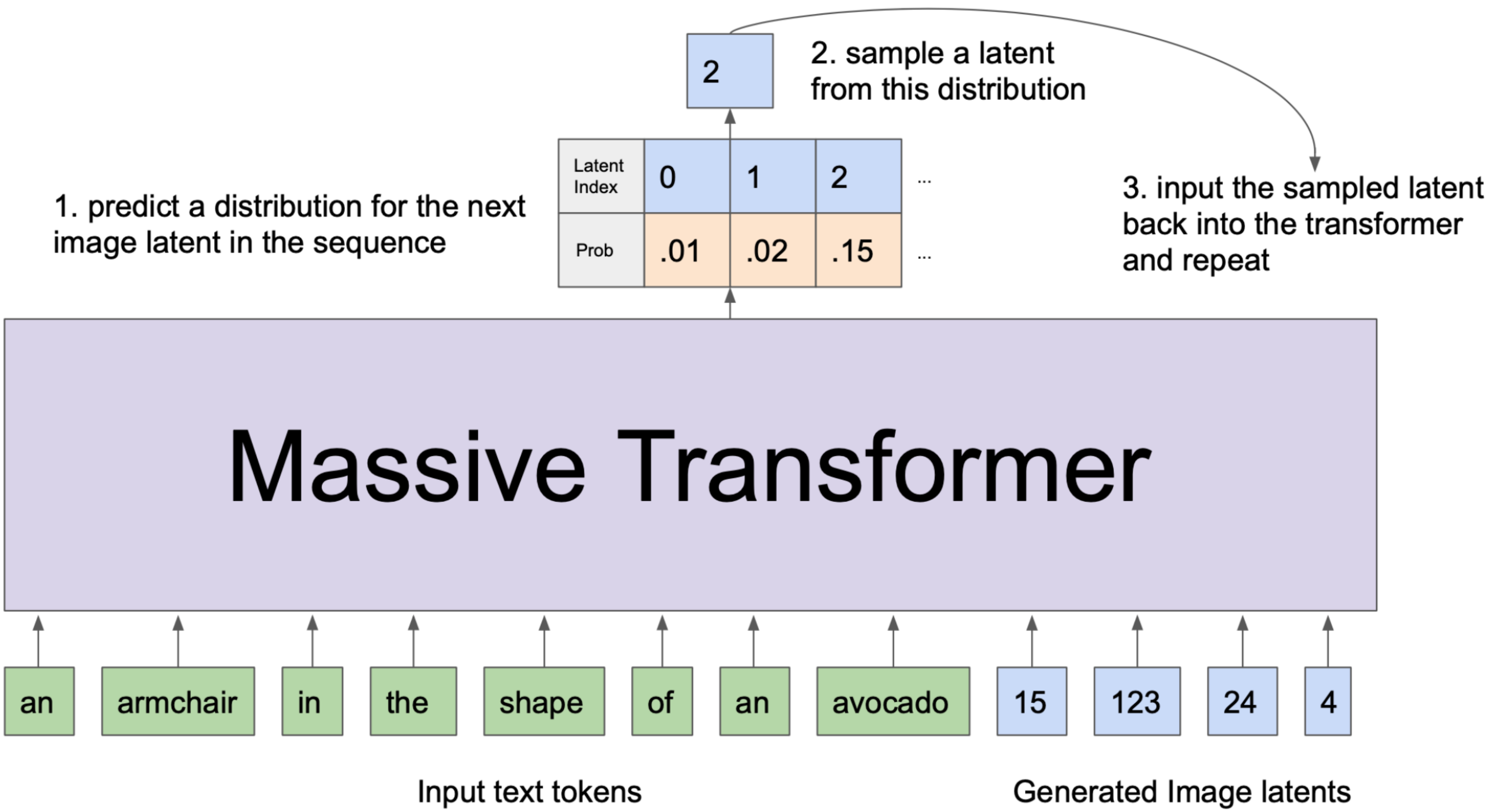


# Related Work - VQ-VAE

Want discrete
latent space?
Vector Quantized
VAE



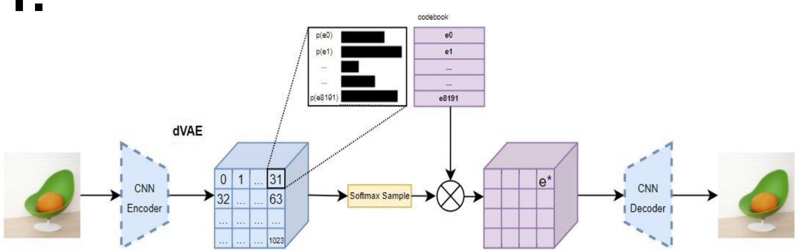
# DALL-E – Central Idea



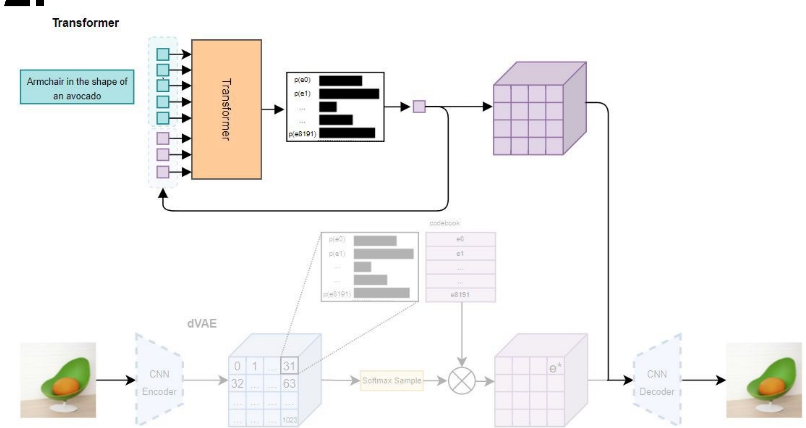
# Model

- Transformer to model text and image tokens as single stream of data
  - Pixels as image tokens takes up too much memory
  - Likelihood objectives prioritize short range dependencies between pixels
  - Solution: 2 stage training!

**1.**

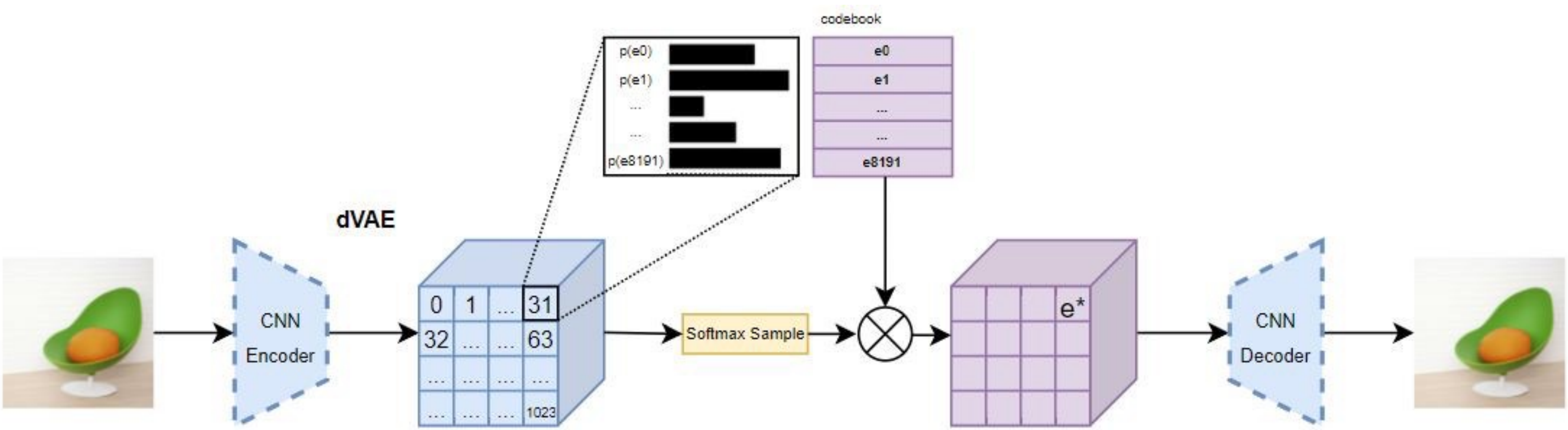



**2.**



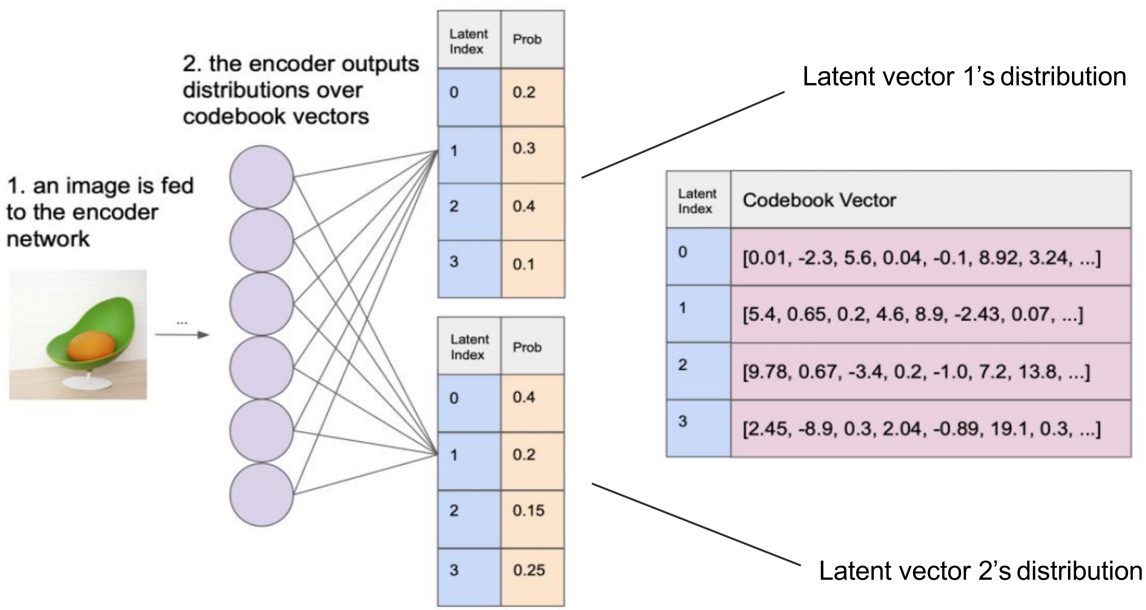
# Stage One: Learning the Visual Codebook

- Discrete Variational Autoencoder (dVAE)
  - Similar to VQ-VAE (in VQ-GAN) but uses distribution instead of nearest neighbor



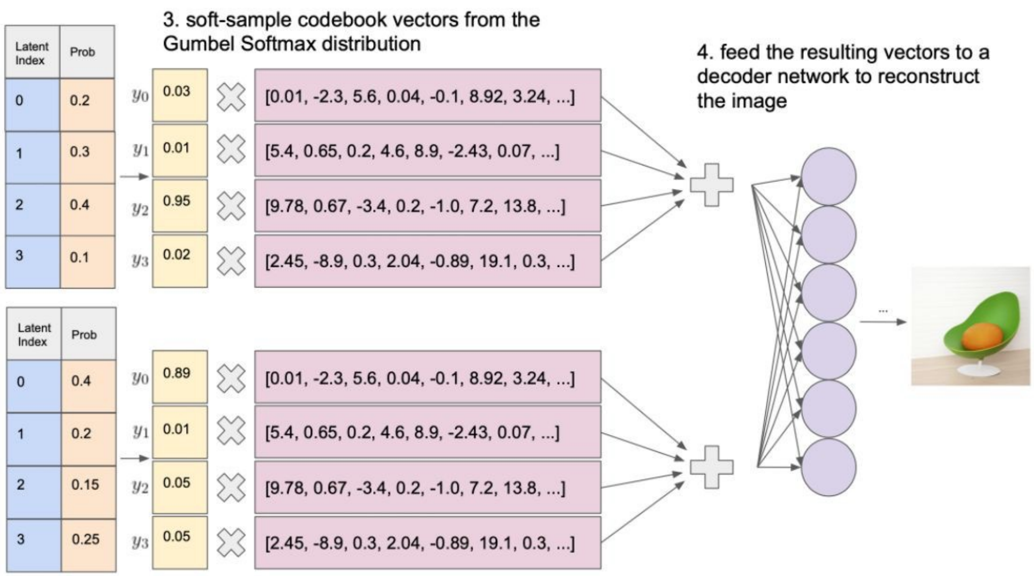
# Stage One: Learning the Visual Codebook

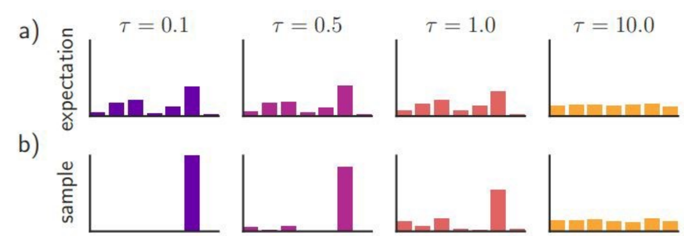
- Discrete Variational Autoencoder (dVAE) encoder



# Stage One: Learning the Visual Codebook

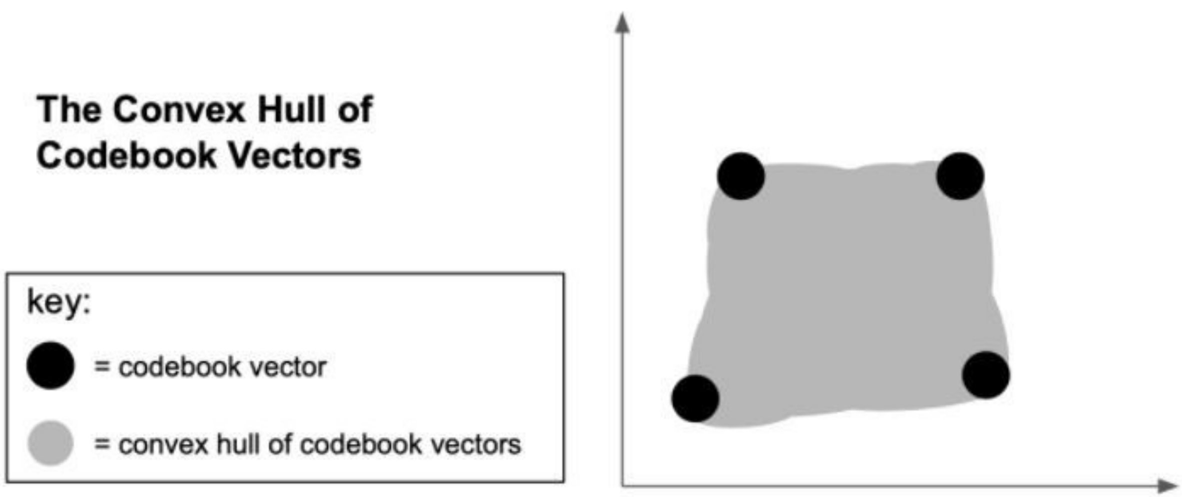
- Discrete Variational Autoencoder (dVAE) decoder
  - Gumbel softmax distribution becomes categorical over training schedule



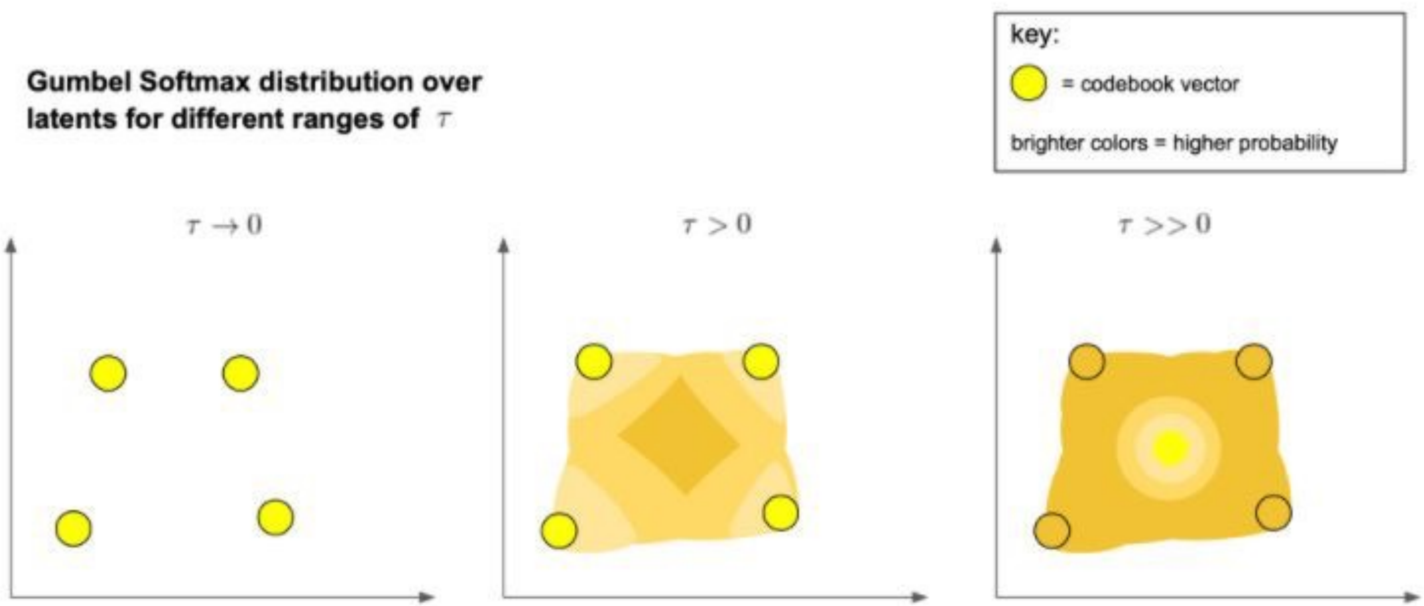
# Stage One: Learning the Visual Codebook

- Discrete Variational Autoencoder (dVAE) encoder
  - Issue: Can't differentiate backprop through category distribution of the bottleneck
  - Solution: Relax the bottleneck to include vectors from convex hull of set of codebook vectors



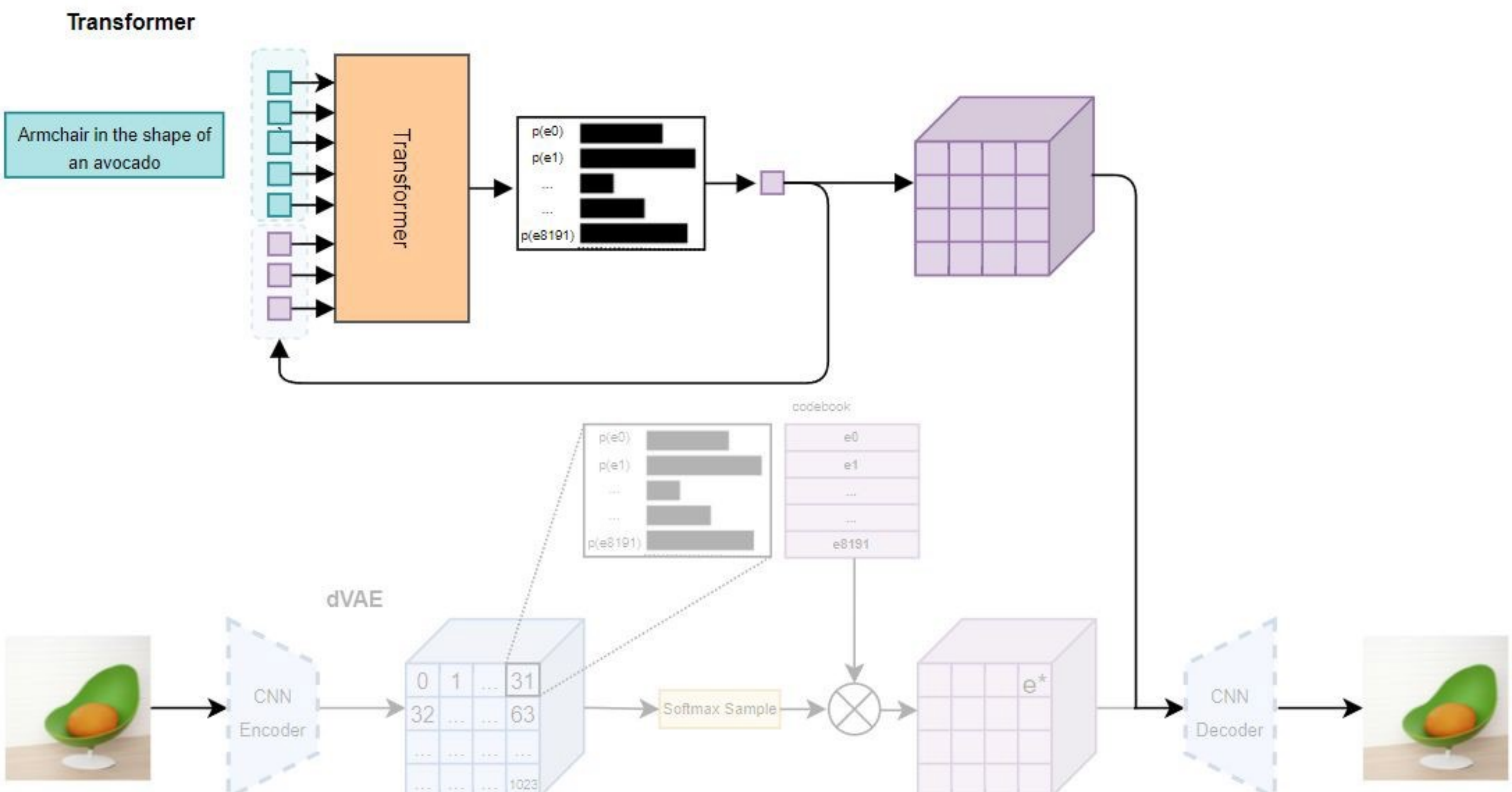
# Stage One: Learning the Visual Codebook

- Gumbel Softmax Relaxation
  - Sample: $z = \text{codebook}[\text{argmax}_i[g_i + \log(q(e_i|x))$
    - Gives weights $y_i$
    - Sampled latent vector is the sum of the weighted codebook vectors
  - Differentiable
  - Relaxation temperature annealing schedule for hyperparameter $\boldsymbol{\tau}$



# Stage Two: Learning Prior Distribution

- Transformer
  - Predict distribution for next token
  - Sample distribution and repeat until 1024 image tokens



# Google's Approaches

- Pegasus (Google, LLM, text summarization)
  - https://ai.googleblog.com/2020/06/pegasus-state-of-art-model-for.html
- Flamingo (Google's GPT-3 and CLIP)
  - https://www.deepmind.com/blog/tackling-multiple-tasks-with-a-single-visual-language-model
- Imagen (Google's DALL-E)
  - https://imagen.research.google
- Bard (Google's answer to ChatGPT, soon)
  - https://blog.google/technology/ai/bard-google-ai-search-updates/

# Meta's Approaches

- OPT (Meta, LLM)
  - https://ai.facebook.com/blog/democratizing-access-to-large-scale-language-models-with-opt-175b/
- Galactica (Meta, LLM chatbot for science)
  - https://galactica.org/