Towards Designing and Exploiting Generative Networks for Neutrino Physics Experiments using Liquid Argon Time Projection Chambers

Nikita Saxena, Paul Lutkus, Shuchin Aeron, Taritree Wongjirad
Tufts University

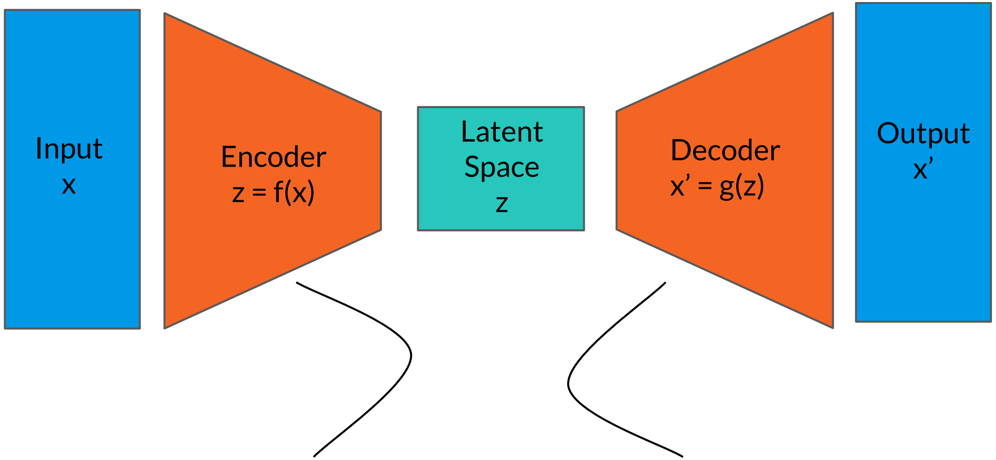
Motivation

Generate LArTPC events given the particle and momentum of the particle.

Why?

- LArTPCs are widely used in neutrino physics.
- Potentially Fast Generation of simulated events (over GEANT4 simulations).

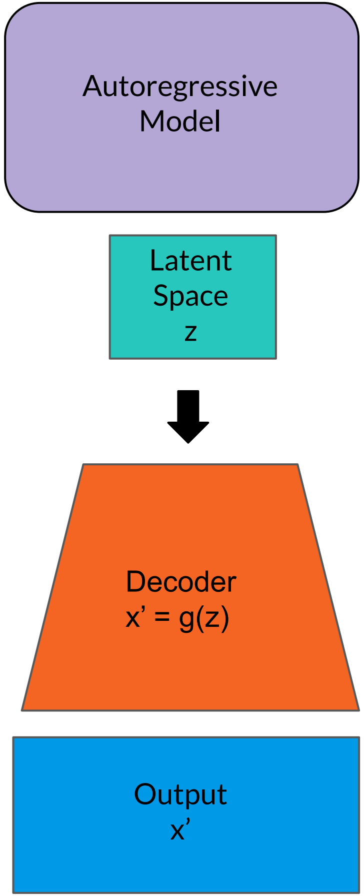
Brief Overview



Learns how the images are represented in latent space.

Learns how to reconstruct image from latent space.

Represents Images



Learns how to generate latent maps.

Generates New Images

Latent Space

- compressed representations of data.
- emphasize the most important and semantically interesting features.

An Oversimplified Example of a Cat/Dog Image Latent Space

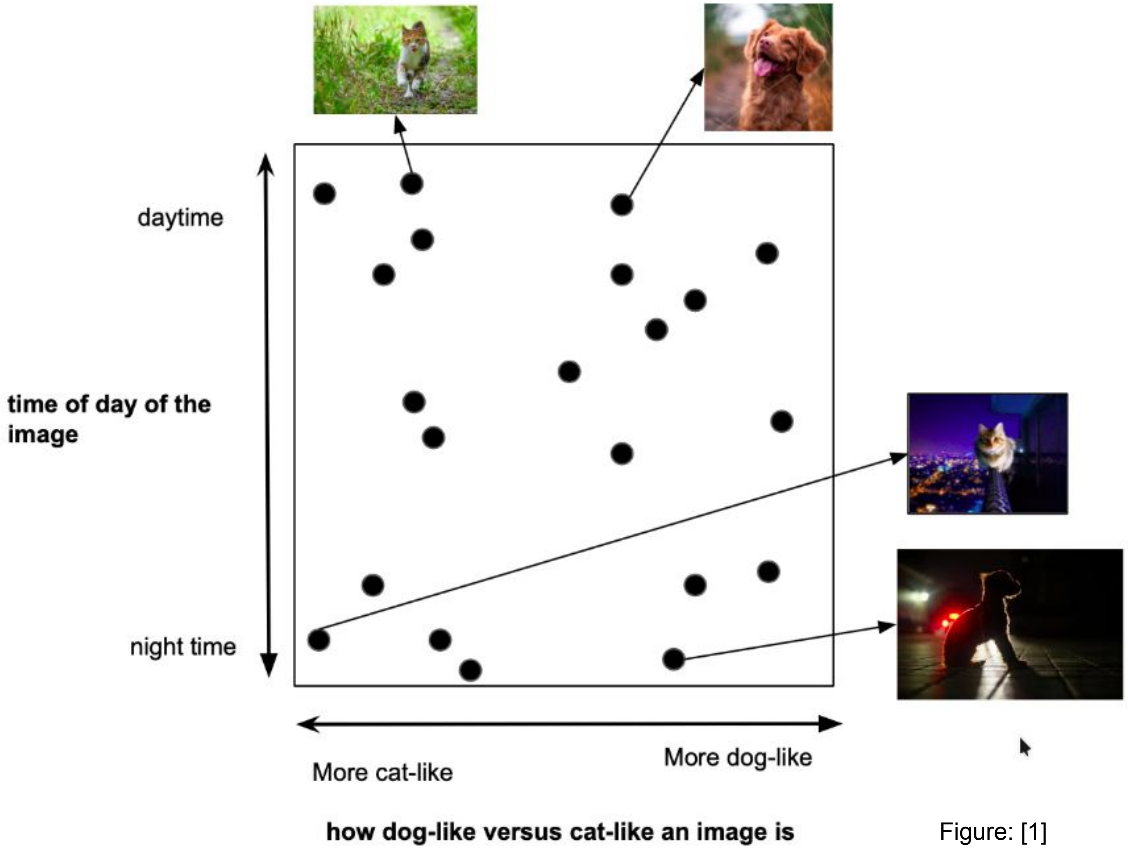
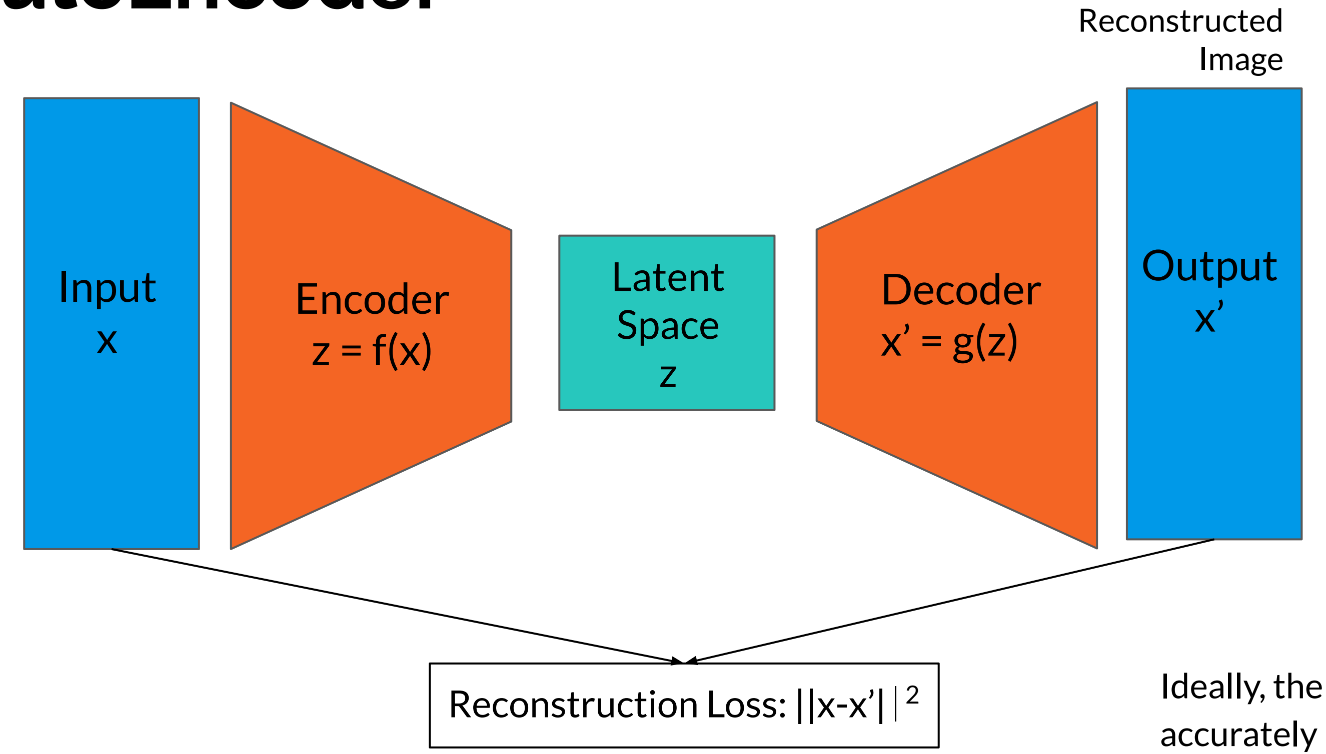


Figure: [1]
Charlie Snell

AutoEncoder

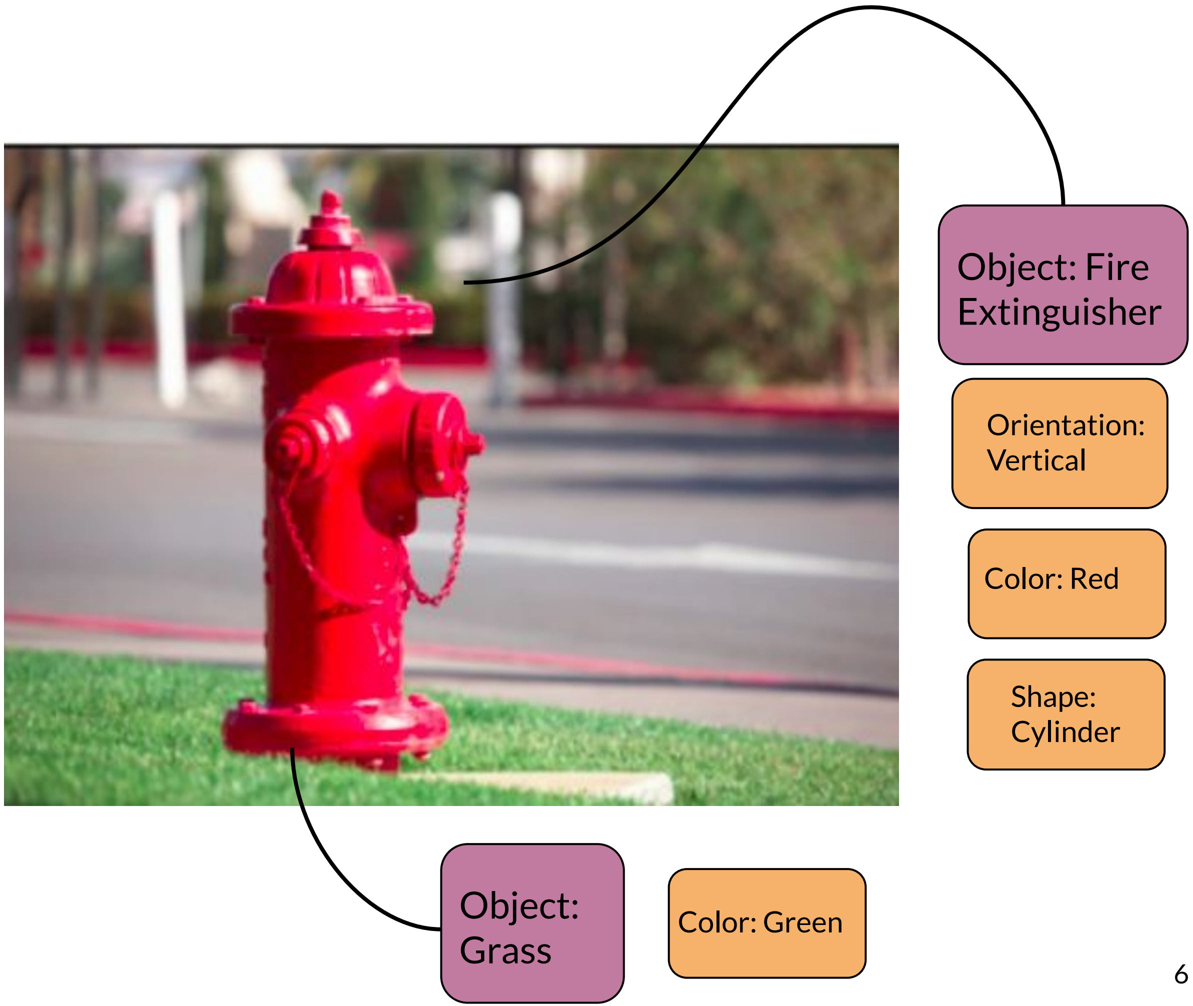


Ideally, the decoder should be able to accurately reconstruct the raw data from the encoder's latent representation, i.e., x=x'

Discrete Space

Latents do not necessarily need to be continuous vectors, it really just needs to be some numerical representation for the data.

A lot of the data we encounter in the real world favors a discrete representation. For example, images contain **discrete objects** with some **discrete set of qualifiers.**



Encoder

image to
discrete codes

56	73	67	23	81	19	...

Decoder

56	73	67	23	81	19	...

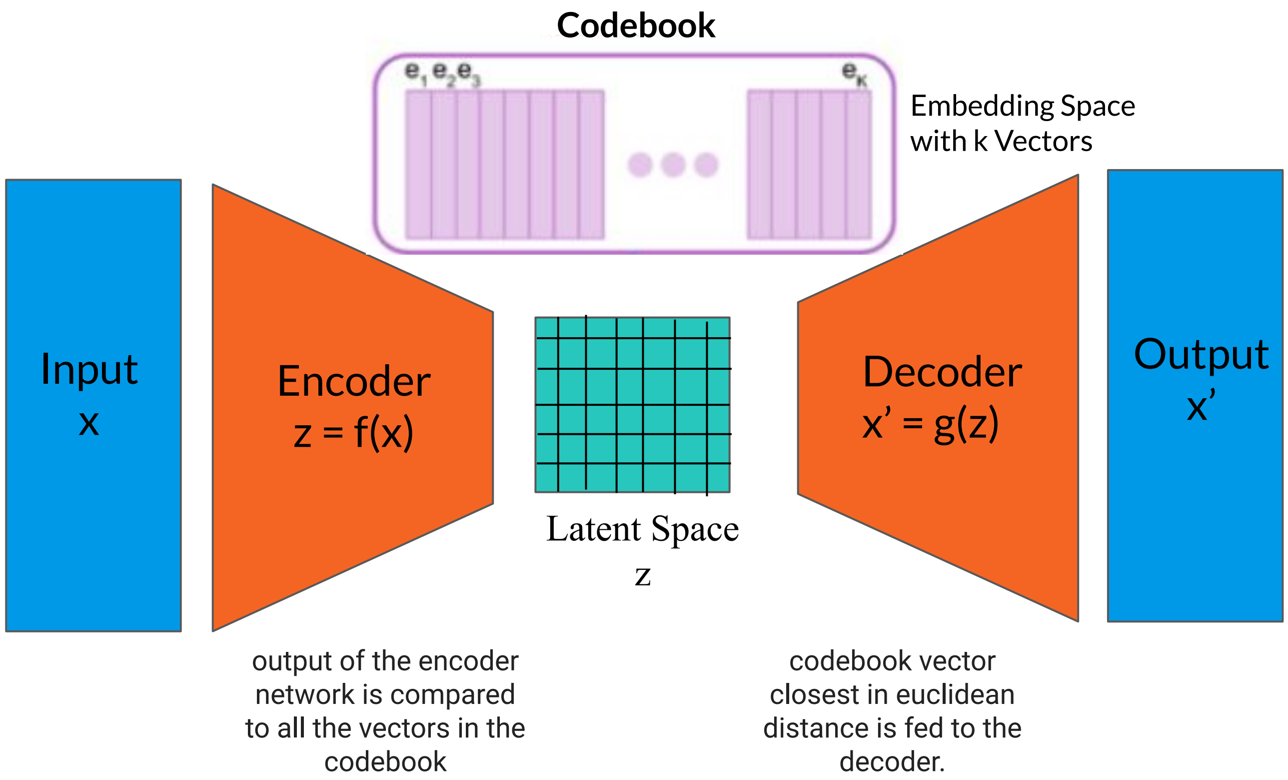
discrete codes
to image

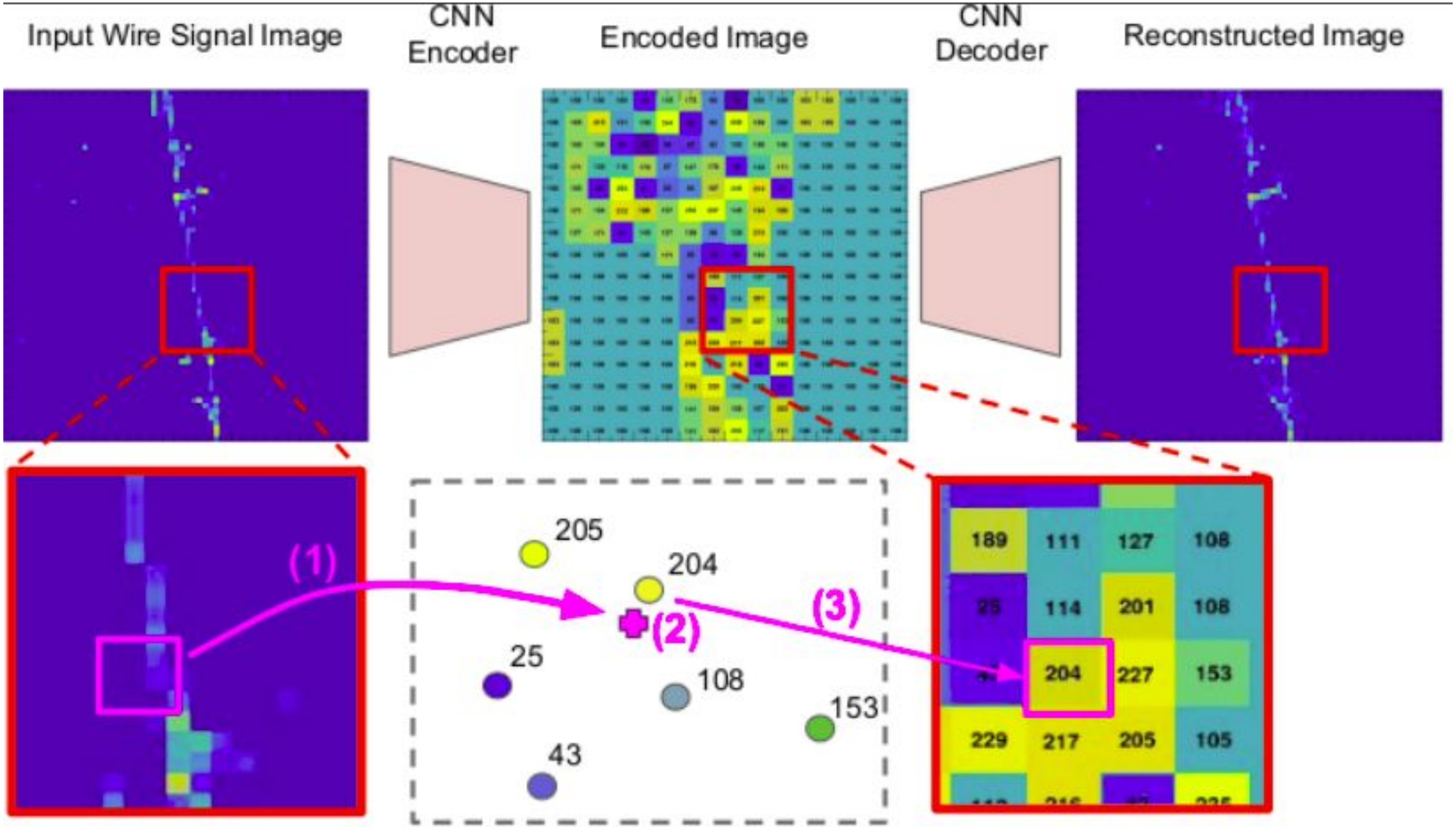
Figure: [2]
Charlie Snell

Quantizing Autoencoders: VQ VAE

Discrete codebook component added to the network.

The codebook is basically a list of vectors associated with a corresponding index.




Input Wire Signal Image
CNN Encoder
Encoded Image
CNN Decoder
Reconstructed Image
(1)
205
204
(2)
(3)
25
108
153
43
189
111
127
108
114
201
108
204
227
153
229
217
205
105

Producing Multiple Codes

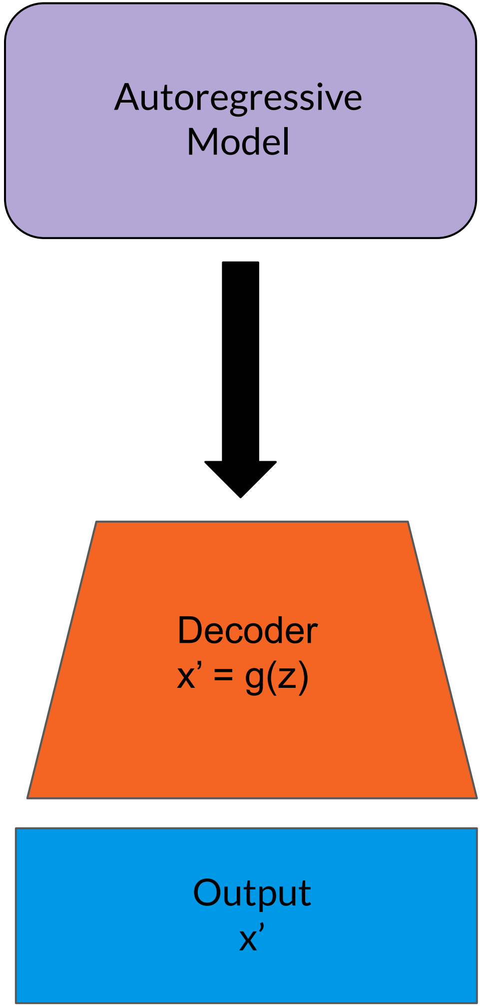
- There is such a restricted set of vectors that can be fed to the decoder (just the set of codebook vectors).

How could one ever expect to generate the huge quantity and diversity of possible images when the decoder can only accept the set of codebook vectors as input?

- But the encoder does not output just one vector, but instead it usually produces a series of vectors.
- For instance, with images of size 32x32, the encoder might output a 32x32 grid of vectors, each of these are quantized and then the entire grid is fed to the decoder. If we have a codebook of size 512, and then our decoder can basically output $512^{32*32}=2^{9216}$ distinct images!

Learning the Prior

- Network to learn the probability distribution of the codes.
- Generate new data from the distribution by sampling from this prior and feeding the samples to the decoder.



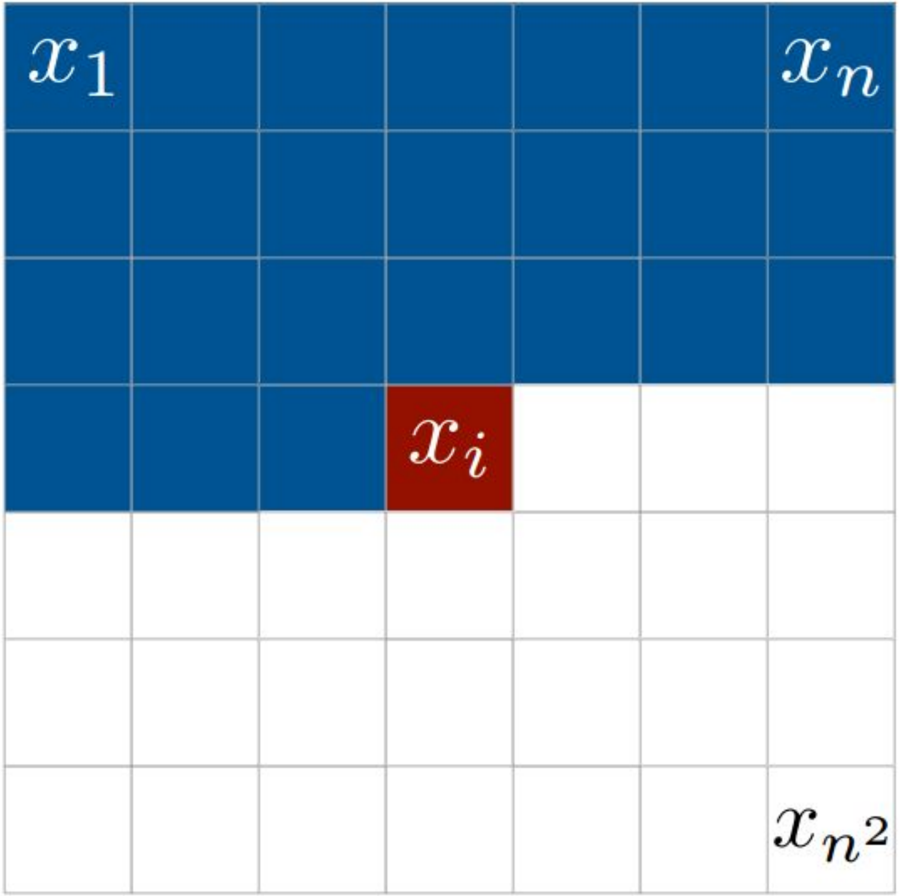
- Learns the distribution of the discrete codes.
- Generates Latent Maps.

Autoregressive Model

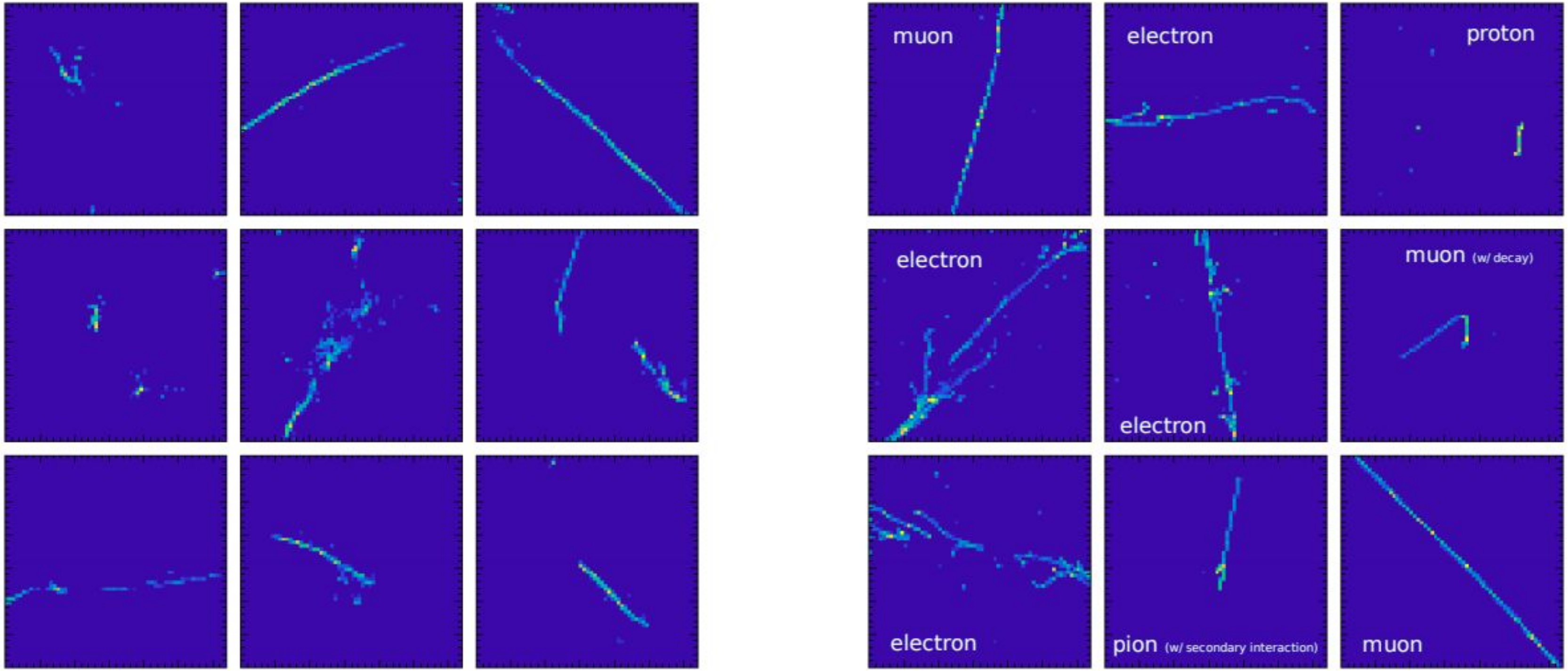
Given all previous latent codes in the sequence, predict the next one.

For images, we can apply autoregressive learning to a sequence that goes from top left to bottom right.

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1}).$$



Events Generated from VQ VAE and PixelCNN



(a) Generated images

(b) Training images

The quality of the output was quantified using track and shower labels.

Figure: [3] Lutkus et. al.

What's next?

Conditional Image Generation

Conditional Image Generation

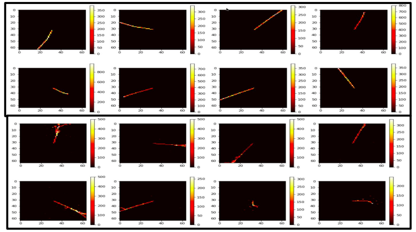
The Idea

Generate the trajectory of a particle given the initial momentum.

Generate Dataset

Train VQ VAE
Architecture: VQ VAE2

Train the conditional Autoregressive Network
Architecture: PixelSNAIL



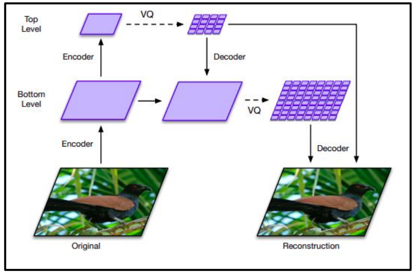


Figure: [4] Ali Razavi et. al.

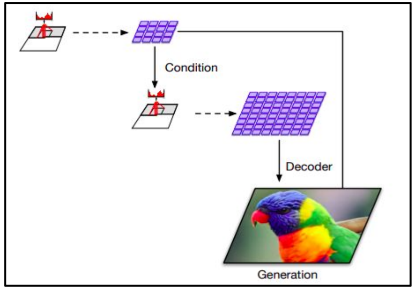
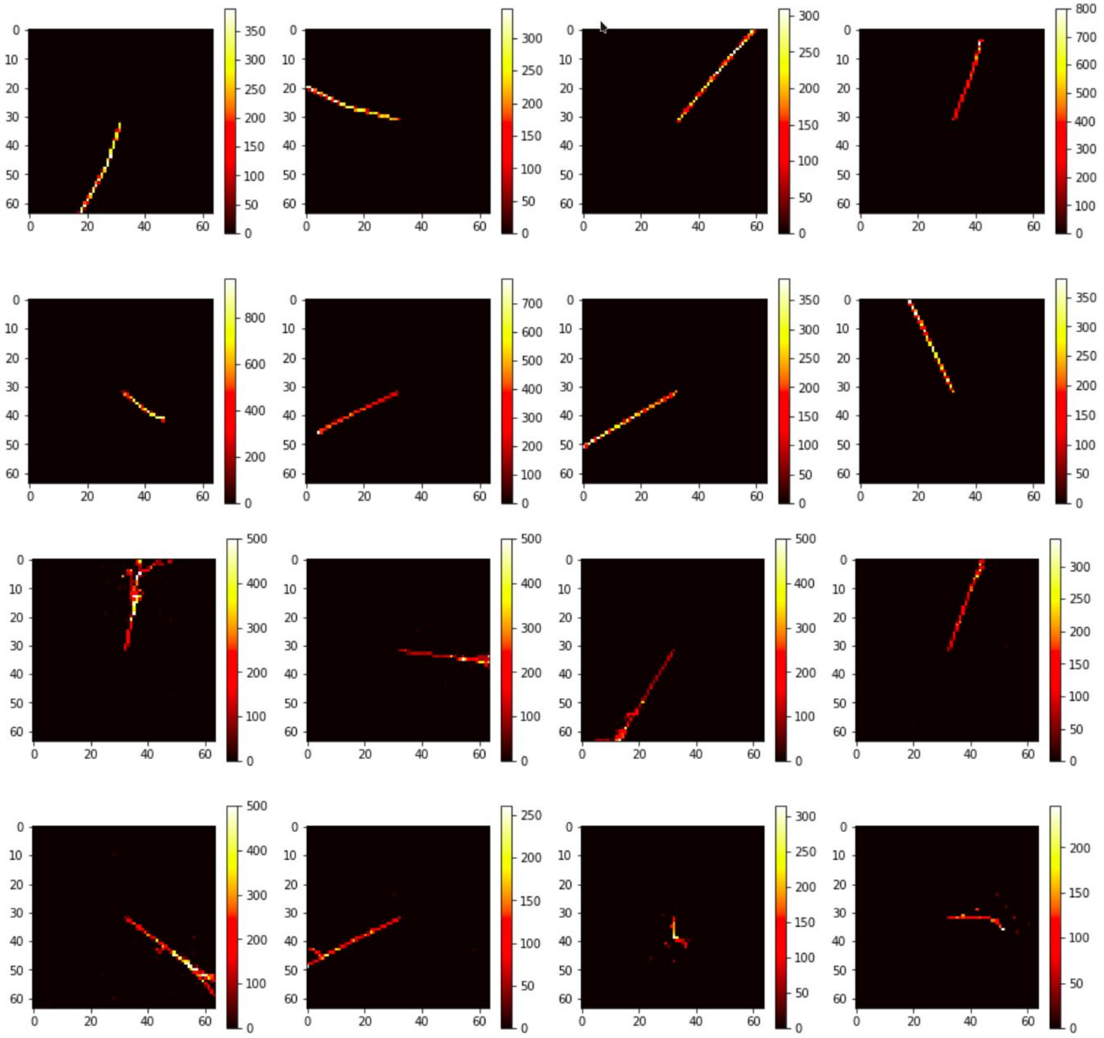


Figure: [5] Ali Razavi et. al.

The Dataset

- Particles: showers and tracks
- Cropped such that the particle is at the center of the image

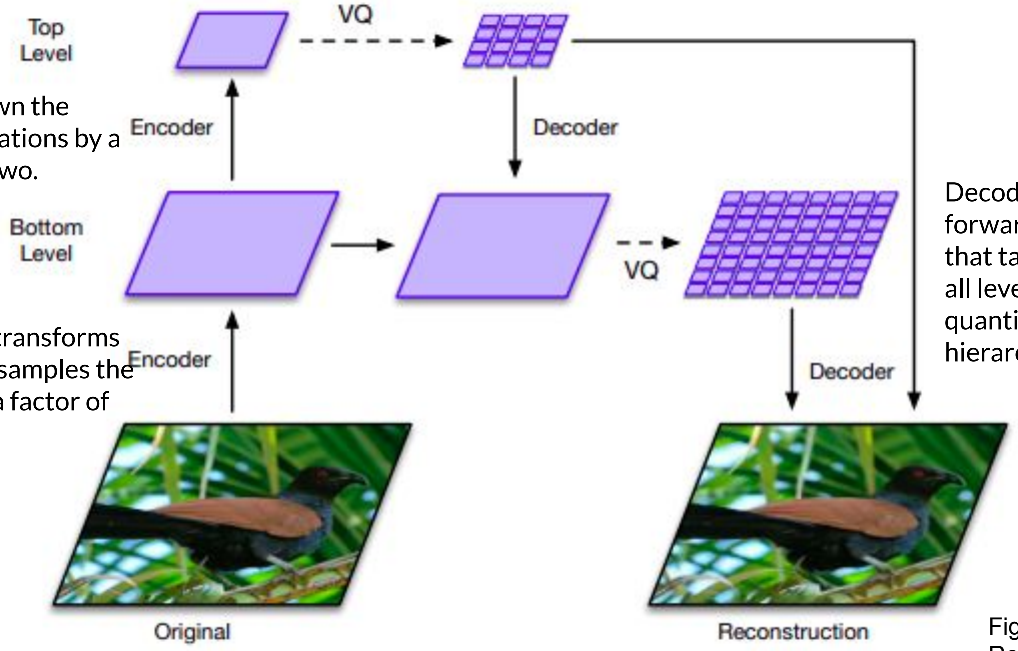


Hierarchical Autoencoders: VQ VAE2

- Top latent code: models global information.
- Bottom latent code: responsible for representing local details.

Scales down the representations by a factor of two.

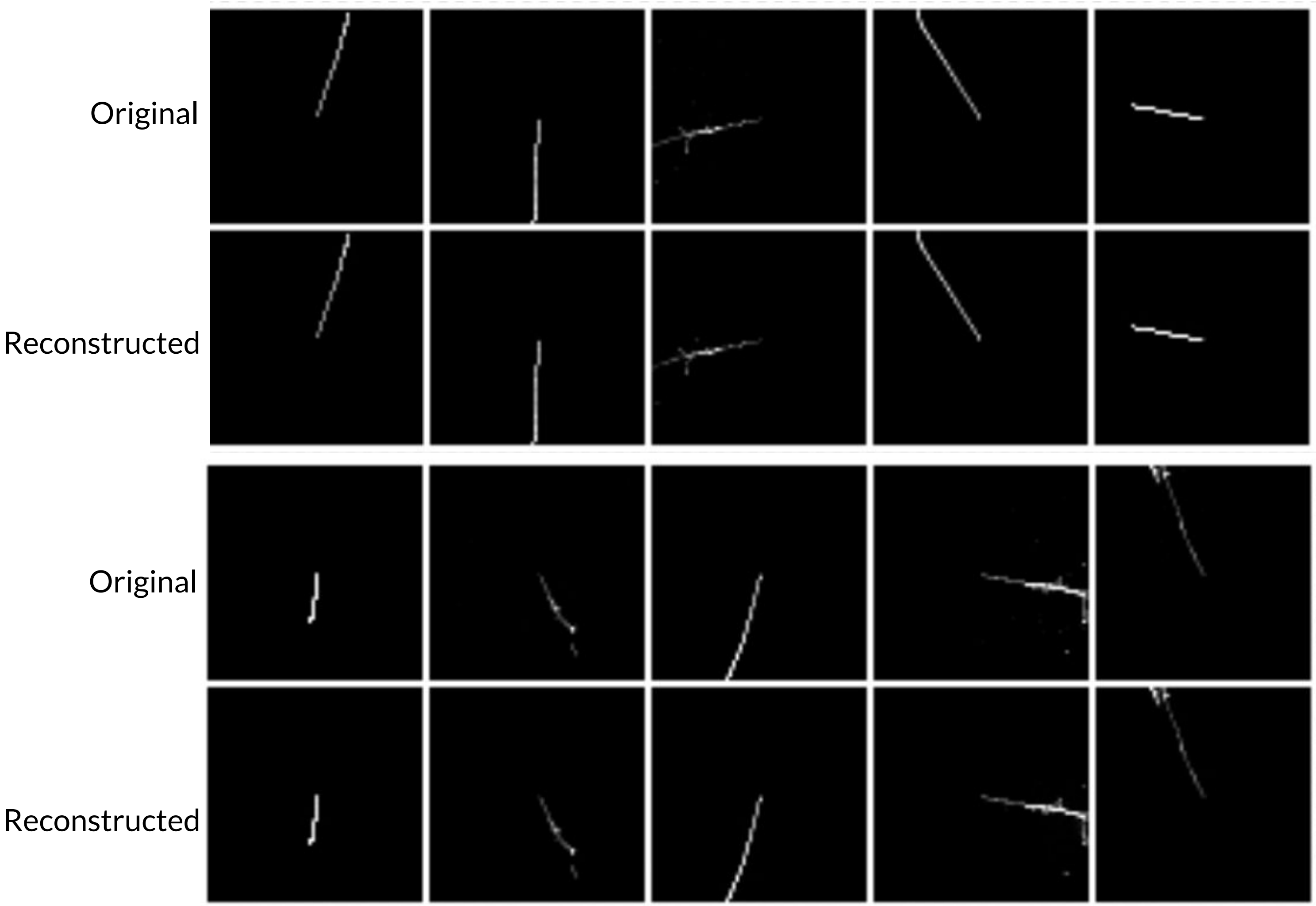
Encoder: transforms and downsamples the image by a factor of four.



Decoder: feed forward network that takes as input all levels of the quantized latent hierarchy.

Figure: [6] Ali Razavi et. al.

VQ VAE 2 Results



PixelSNAIL

- Prior over the top latent map: Responsible for structural global information
- Prior over the bottom latent map: Responsible for the local information

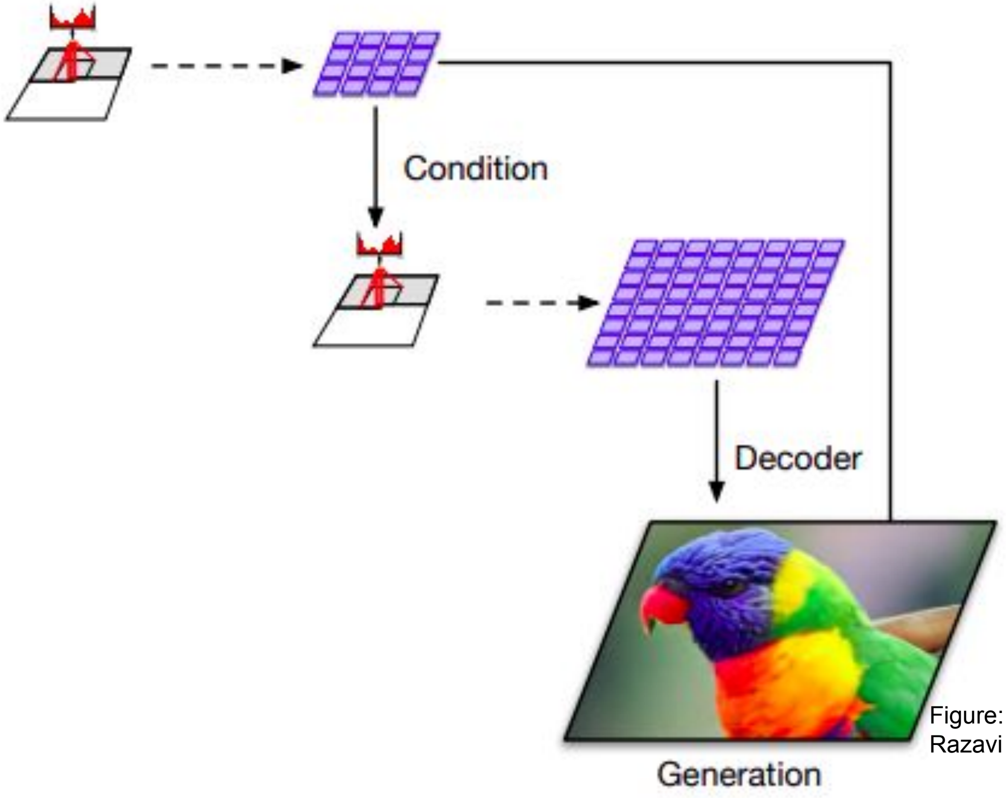


Figure: [7] Ali Razavi et. al.

Summary

- We train a network using VQ VAE (autoencoder) and PixelCNN (autoregressive model) to generate images (unconditioned on particle and momentum) that resemble LArTPC events.
- We are currently pursuing more sophisticated architectures to improve the quality of the generated images.
- Additionally, we are working on designing a network that can generate images conditionally.

Acknowledgements

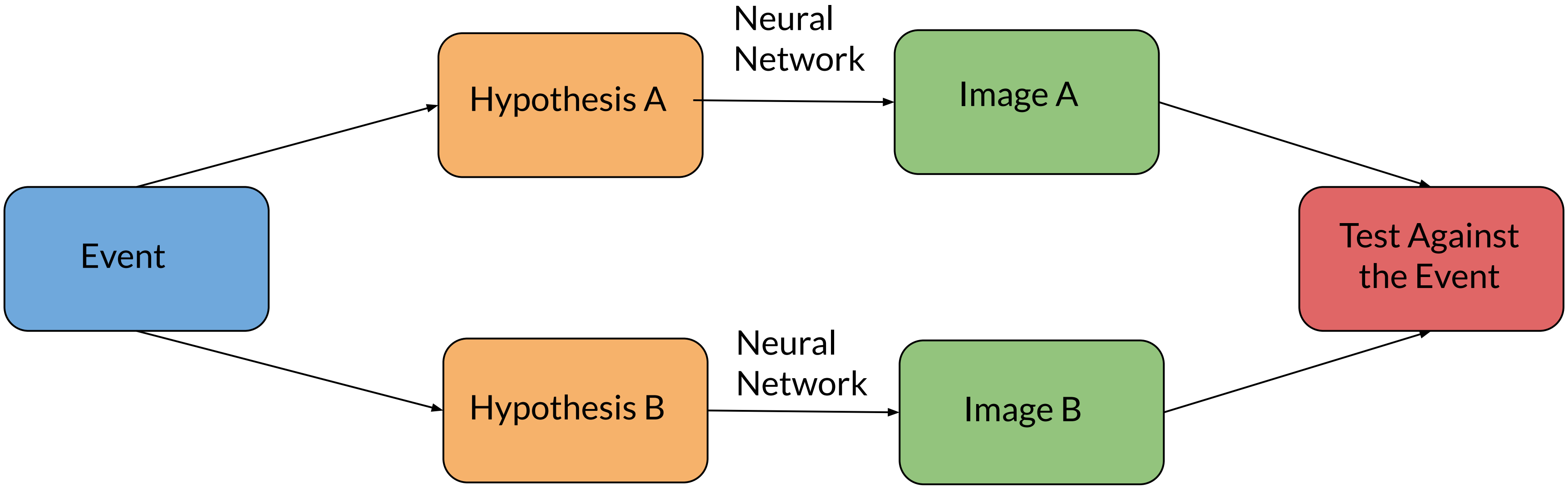
This material is based upon work supported by the U.S. Department of Energy (DOE) and the National Science Foundation (NSF). T.W. was supported by the U.S. DOE, Office of High Energy Physics. S.A. was funded by the NSF.

References

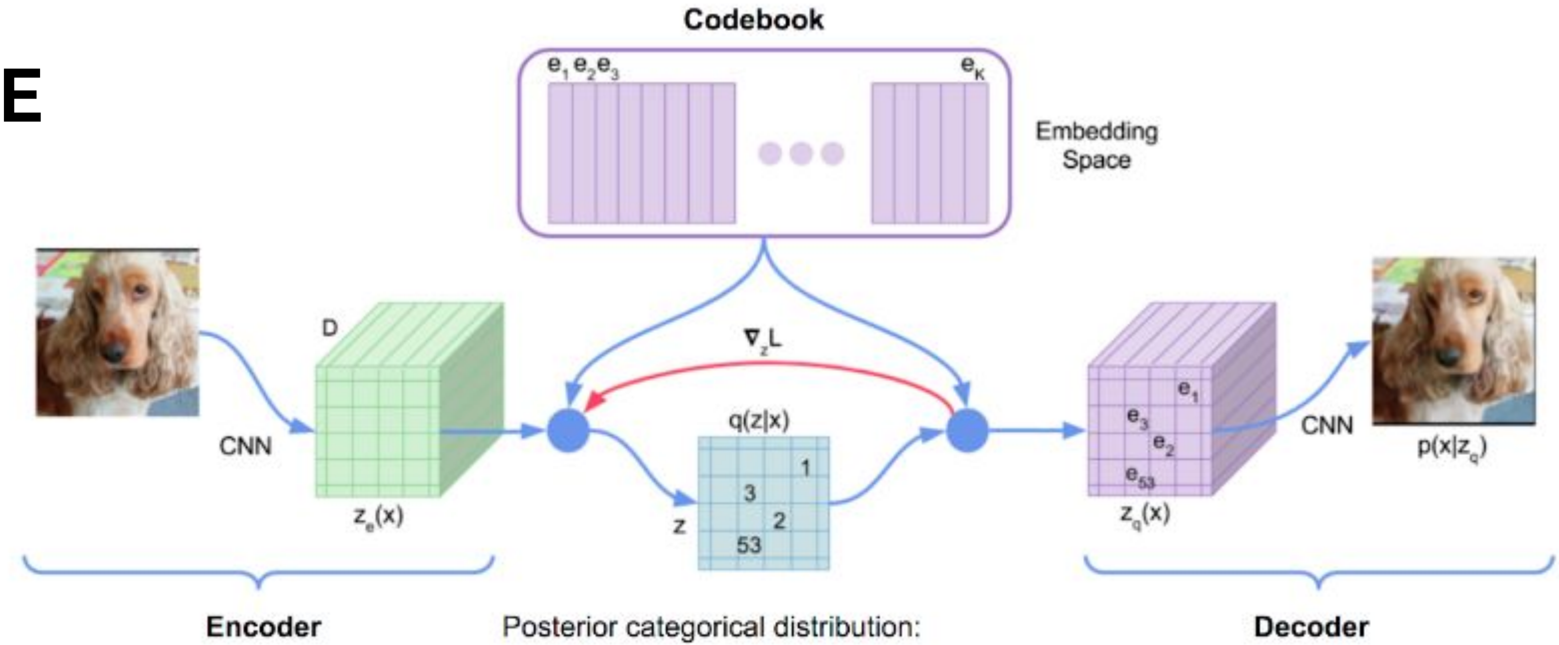
1. Towards Designing And Exploiting Generative Networks For Neutrino Physics Experiments Using Liquid Argon Time Projection Chambers, Paul Lutkus, Taritree Wongjirad, Shuchin Aeron (Conference paper at ICLR 2021) [https://simdl.github.io/files/51.pdf]
2. Generating Diverse High-Fidelity Images with VQ-VAE-2, Ali Razavi and Aaron van den Oord and Oriol Vinyals (arxiv:1906.00446)
3. PixelSNAIL: An Improved Autoregressive Generative Model, Xi Chen and Nikhil Mishra and Mostafa Rohaninejad and Pieter Abbeel (arxiv: 1712.09763)
4. Understanding VQ-VAE (DALL-E Explained Pt. 1), Charlie Snell [https://ml.berkeley.edu/blog/posts/vq-vae/]
5. Conditional Image Generation with PixelCNN Decoders, Van der Oord et al. (arXiv:1606.05328)
6. Neural Discrete Representation Learning, Van der Oord et al. (arXiv:1711.00937)

Appendix

Application of Generative Network: Hypothesis Testing



VQ VAE



Posterior categorical distribution:

$$q(\mathbf{z} = \mathbf{e}_k|\mathbf{x}) = \begin{cases} 1 & \text{if } k = \arg\min_i \|\mathbf{z}_e(\mathbf{x}) - \mathbf{e}_i\|_2 \\ 0 & \text{otherwise.} \end{cases}$$

x: raw input
z: latent space representation
D: Dimension of Embedding Space
k: Number of Codebook Vectors
e_i: i^{th} codebook vector
$z_e(x)$: encoder vector for input x
$z_q(x)$: resulting quantized vector that is passed as input to the decoder

Figure: [8] Van der Oord et al. Neural Discrete Representation Learning

Track and Shower Labels

Input
Image

Label
Image

Cyan: Showers
Yellow: Tracks

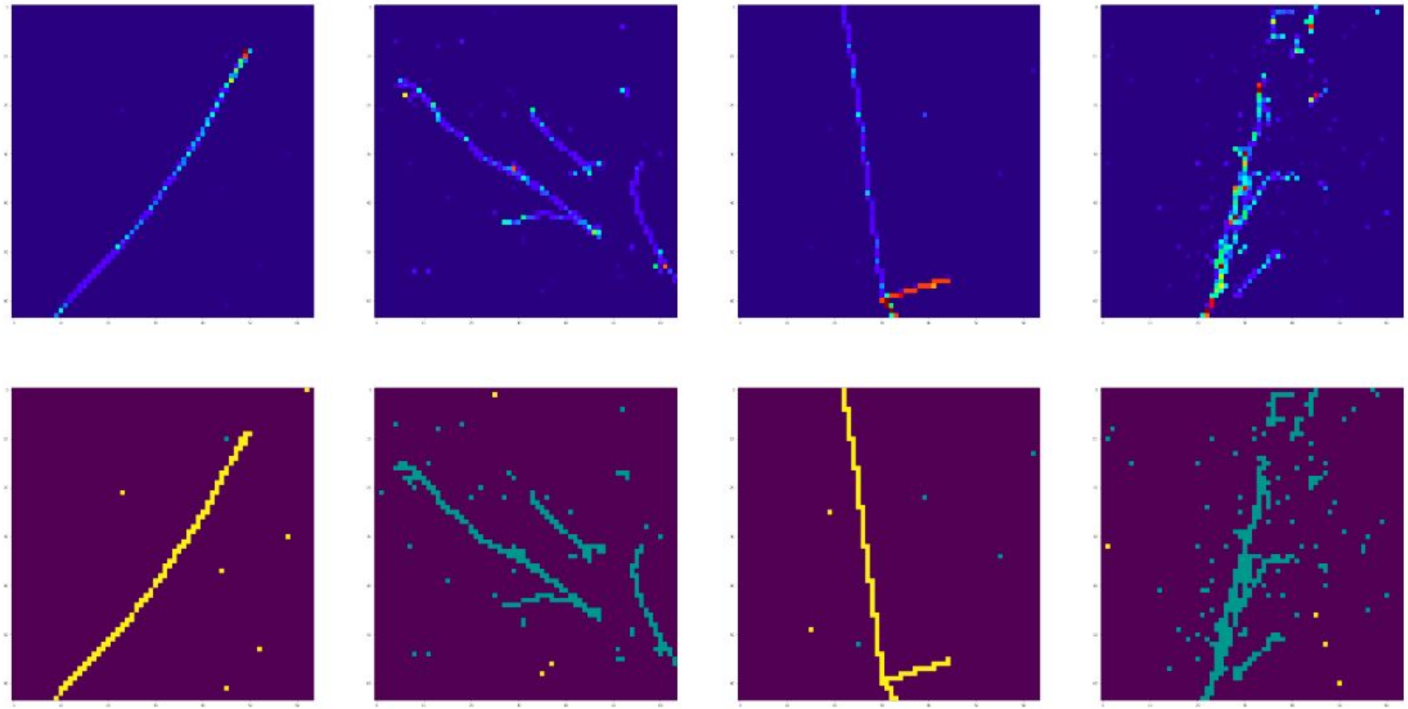
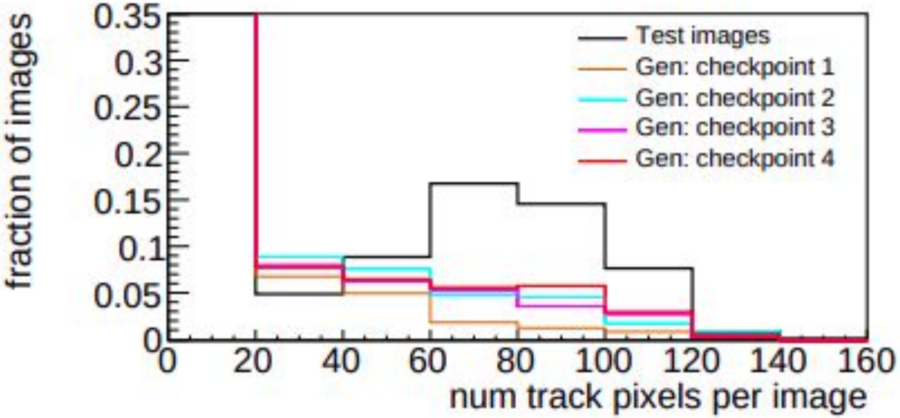


Figure: [9] Lutkus et. al.

Results from VQ VAE and PixelCNN



Comparing Frequency of SSNet Labels per Image

Figure: [10] Lutkus et. al.