Discretizing VAEs for Lossy Image Compression

Zhihao Duan
duan90@purdue.edu
PhD student at Video and Image Processing Lab, ECE
Work on data compression

April 2022

This is part of my ongoing PhD research

Outline

- Motivation
  - VAE vs. rate distortion theory

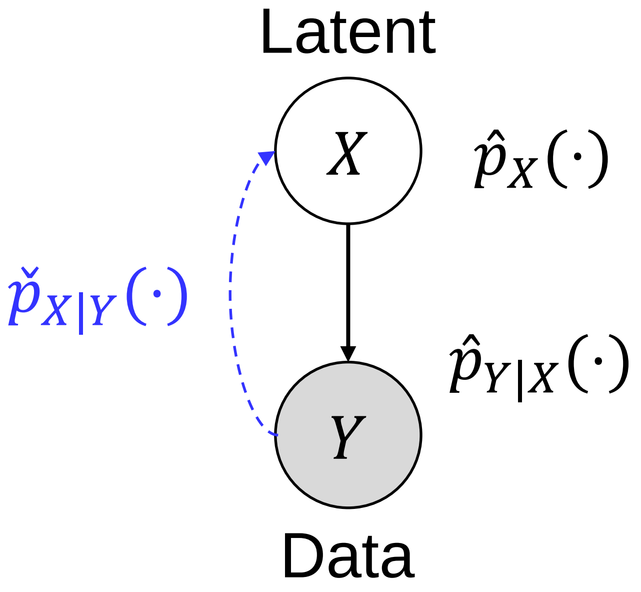
- Method
  - We want a VAE with integer-valued latent variable
  - Set the components of a continuous VAE (prior, emission, recognition) to make it "quantization-aware"

- Experimental results
  - It works well

Motivation

- Recall: VAE and min. free energy

$$\min \underbrace{D_{KL}\left(\check{p}_{X|Y} \parallel \hat{p}_X\right)}_{\text{"Rate"}} + \underbrace{E_{\check{p}_{X|Y}}\left[\log \frac{1}{\hat{p}_{Y|X}(Y|\check{X})}\right]}_{\text{"Distortion"}}$$



- People have used VAEs to estimate the *R*-*D* function for natural images [1]

- But VAEs cannot be directly used for lossy compression
- A workaround: discretize X

[1] Yibo Yang, Stephan Mandt. (2022). Towards Empirical Sandwich Bounds on the Rate-Distortion Function. The 10th International Conference on Learning Representations.

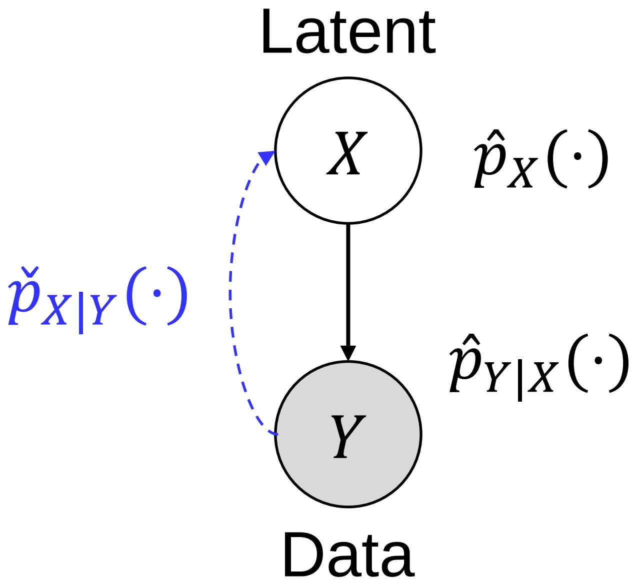
Discretize a VAE

- The simplest method: nearest int. quantization of X
  - Deterministic, not a distribution
  - Gradient is zero almost everywhere

- We use uniform distribution to model quantization error

$$\check{p}_{X|Y}(\cdot\,|y) = U(f(y) - 0.5, f(y) + 0.5)$$

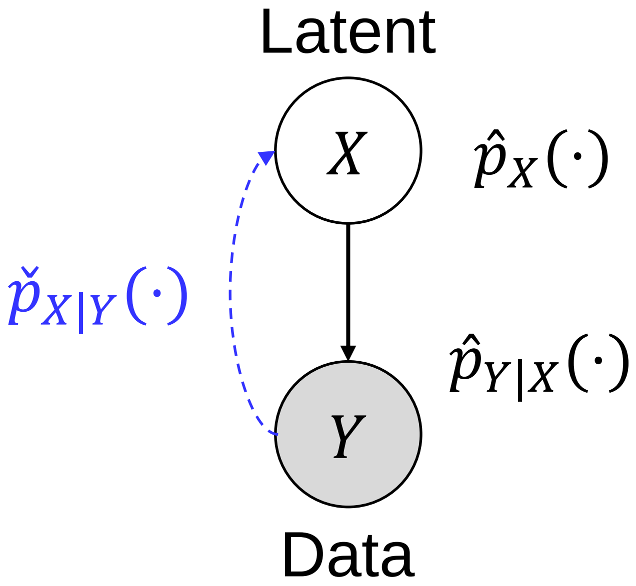
$f(\cdot)$: a neural network (or encoder)



- Training: $x \leftarrow f(y) + u$, where u $\sim U(-0.5,0.5)$
- Testing: $x \leftarrow \lfloor f(y) \rceil$

Discretize a VAE

- We have chosen $\check{p}_{X|Y}(\cdot\,|y)$ to be uniform
- What should $\hat{p}_X(\cdot)$ be?
  - The shape of the pdf should be like $\check{p}_{X|Y}(\cdot)$
  - Should be non-zero everywhere



Discretize a VAE

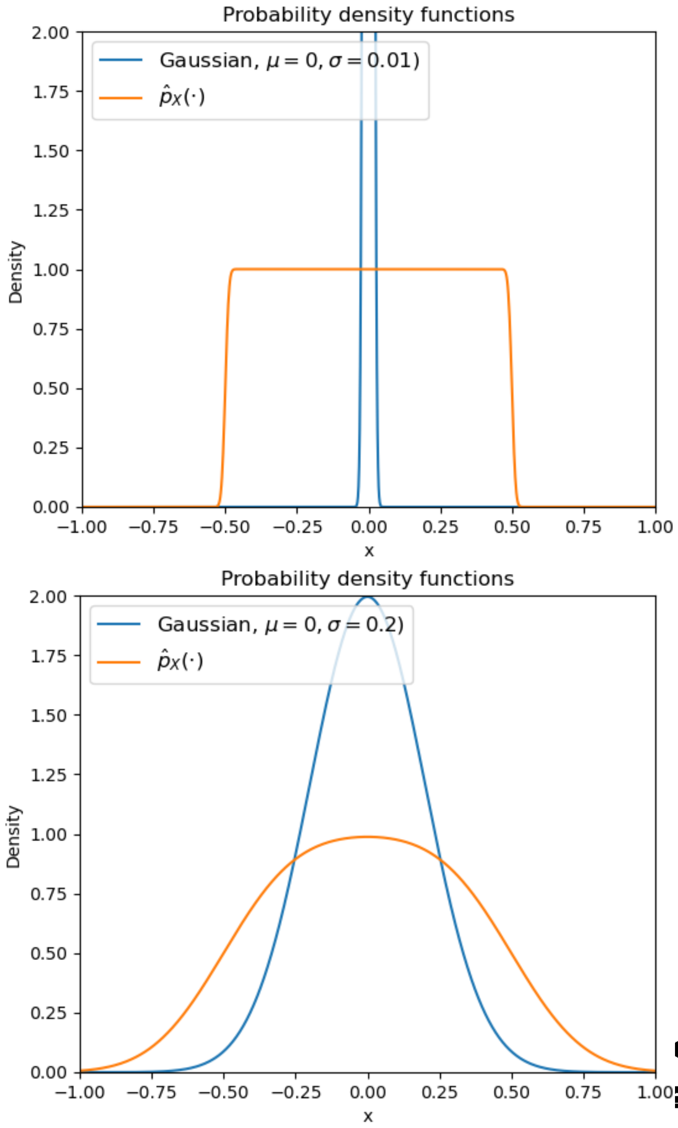
- We have chosen $\check{p}_{X|Y}(\cdot\,|y)$ to be uniform
- What should $\hat{p}_X(\cdot)$ be?
  - The shape of the pdf should be like $\check{p}_{X|Y}(\cdot)$
  - Should be non-zero everywhere
- A good choice:

$$\hat{p}_X(x) \;\propto\; F_{\mu,\sigma}(x+0.5) - F_{\mu,\sigma}(x-0.5)$$

$F_{\mu,\sigma}$: cdf of Normal(μ, σ^2)

Notes:

- $\ldots, \hat{p}_X(-1), \hat{p}_X(0), \hat{p}_X(1), \ldots$ is a discrete distribution
- This enables us to encode quantized X into bits



Discretize a VAE

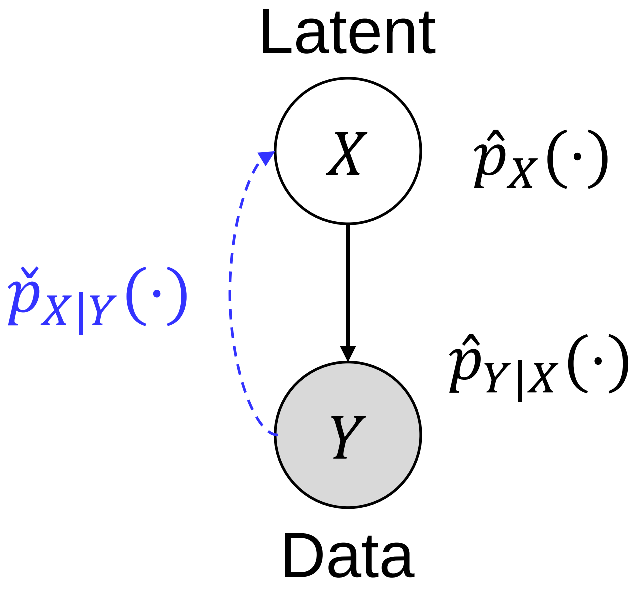
The emission distribution:

$$\hat{p}_{Y|X}(y|x) \propto e^{-\lambda \cdot d(g(x),y)},$$

where

- $g(\cdot)$ is a neural network (decoder)
- $g(x)$: reconstruction
- λ is a scalar

- Negative log-likelihood = $\lambda \cdot d(g(x), y)$



Experiment

- We have specified all components of a VAE
- Distortion metric $d(\cdot)$: mean squared error (MSE)

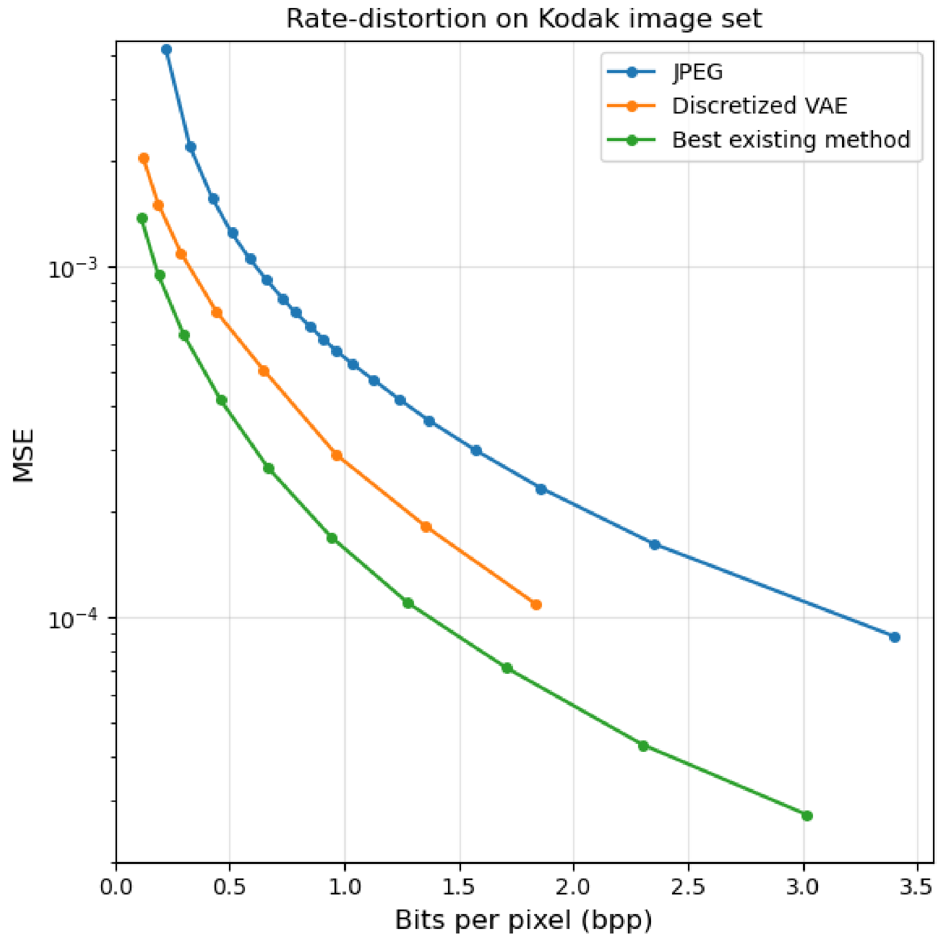
- Training: min. free energy
  - – COCO dataset: 118,287 natural images.
  - – 64x64 image patches

- Testing: nearest int. quantization + coding
  - – Kodak image set: 24 natural images

Results

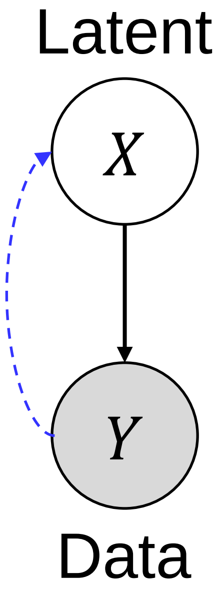
Methods in comparison:

- JPEG
- Best existing method
  - Based on deep learning
  - A VAE coupled with an (spatially) autoregressive model
- Discretized VAE (ours)

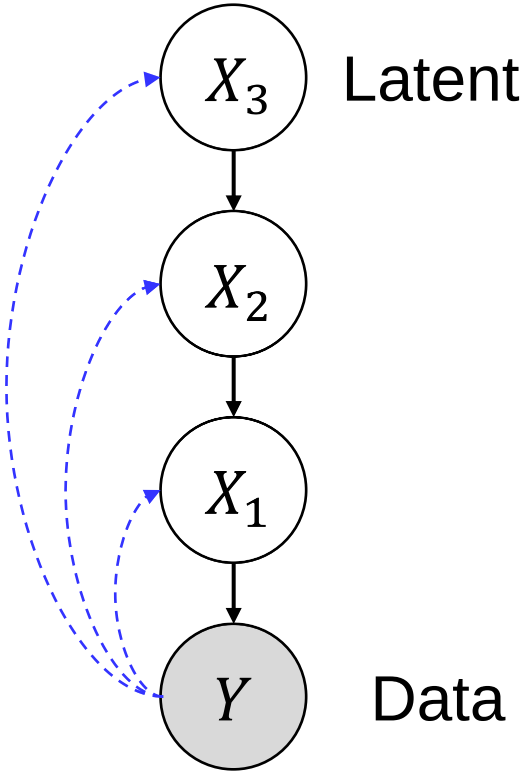


Results

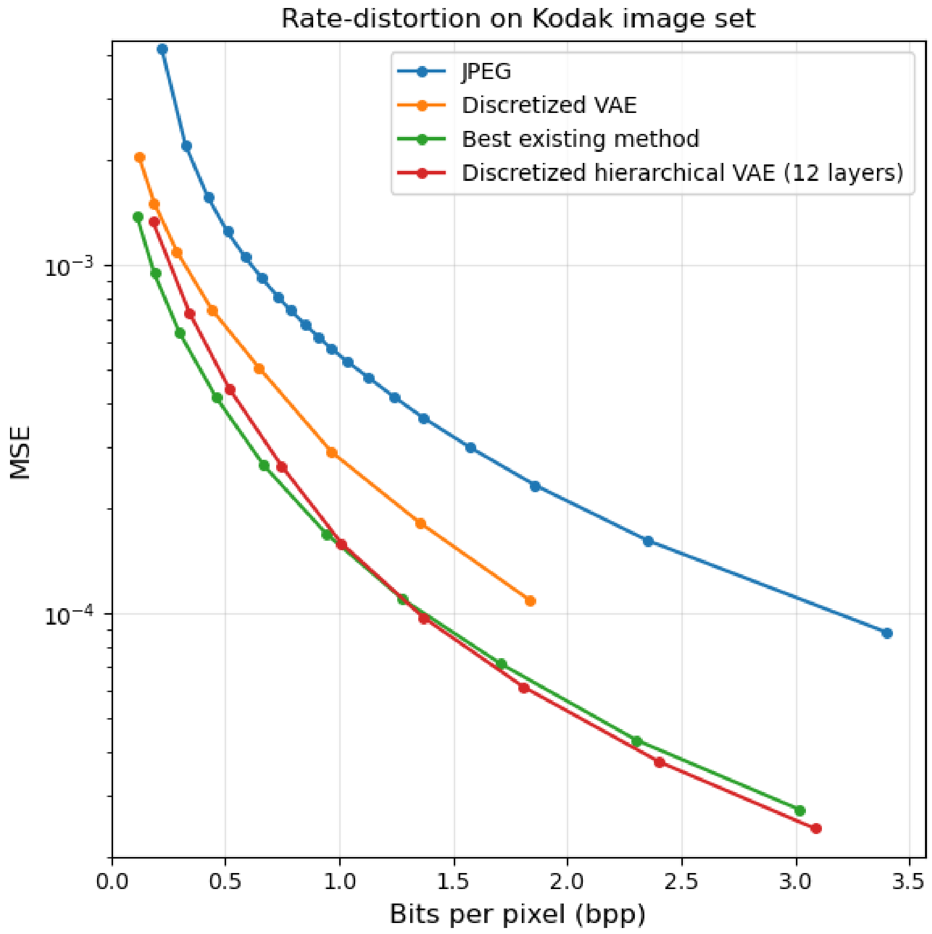
If use a hierarchical VAE instead of a standard one:



One layer VAE



Hierarchical VAE
(3 layers)



Unconditional Samples (64x64)

- Training set image patches:

- Standard VAE samples:

- Hierarchical VAE samples
  - Continuous latent variables:

  - Integer latent variables:
    - (With same random seed)

Conclusion

What I learned in this project:

- Latent variable models (in particular VAEs) works well for lossy compression
- Neural network architecture (ResNets, more layers) and training tricks (learning rate, gradient clipping) matters a lot

Existing work:

- Nearest int. quantization and uniform noise for compression
- Hierarchical VAEs

My work:

- Formulate the uniform noise approach into the VAE framework

Future directions:

- Apply normalizing flows to the recognition distribution