ID 4: VQ-VAE a posteriori with Geodesic Quantization

July 7, 2025

Ana del Río and Darío González

Abstract

In this project we will revisit the classical VQ-VAE pipeline by proposing an alternative approach which instead of learning a discrete latent space jointly with the encoder, first, we will train a standard Variational Autoencoder (VAE) on the MNIST dataset and perform vector quantization 'a posteriori'. In order to capture in a better way the intrinsic geometry of the latent space, instead of using Euclidean distances, we will use geodesic distances computed by a modified k-means graph so it can work with geodesic distances.

We will compare this model with a classic VQ-VAE trained in an end-to-end form, evaluating both models in terms of reconstruction quality and efficiency of the code.

The preliminary results suggest that geodesic quantization can produce latent representations with a better structure, and this might have a positive impact on posterior tasks.

1. Introduction

Unsupervised learning is a fundamental tool for discovering structured representations in complex data, without the need for explicit labels. In this category, we can find Variational Autoencoders (VAE) and their discrete variants, such as Vector Quantized Variational Autoencoders (VQ-VAE), and they are really effective in tasks such as compression or sample generation.

Unsupervised learning is a fundamental tool for discovering structured representations in complex data, without the need for explicit labels. In this category, we can find Variational Autoencoders (VAE) and their discrete variants, such as Vector Quantized Variational Autoencoders (VQ-VAE), and they are really effective in tasks such as compression or sample generation. However, classical VQ-VAE learns the code book jointly with the encoder and relies on euclidean distances, which may ignore the manifold's intrinsic geometry.

In this work, we propose a modular, geometry-aware pipeline:

1. Train a continuous VAE on MNIST.
2. Build a k-NN graph ($k = 10$) over the learned latent codes and compute geodesic distances.
3. Apply geodesic k-means to obtain a discrete code book.
4. Train an auto regressive RNN on the discrete codes to generate new samples.

By doing all of this, we want to capture better the structure of the latent space before applying quantization, so we will obtain more coherent and useful discrete representations.

Finally, we compare the performance of the alternative that we implemented with the classical VQ-VAE.

2. Related Work

Several works have studied non-Euclidean distances in latent spaces. Tenenbaum et al. (Tenenbaum et al., 2000) introduced Isomap, which approximates geodesics via a k-NN graph to recover manifold geometry. Asgharbeygi and Maleki (Asgharbeygi & Maleki, 2008) proposed geodesic k-means by replacing Euclidean with shortest-path distances on a graph. Yang et al. (Yang et al., 2018) demonstrated that geodesic clustering in VAEs yields high-quality clusters despite latent distortions.

2.1. Our contribution

Unlike standard VQ-VAE, we first train a continuous VAE and then perform *a posteriori* geodesic quantization via a k-NN graph and geodesic k-means, followed by auto regressive modeling of the resulting discrete codes. This

Email: Ana del Río and Dario González <delriodelbarrio.2183763@studenti.uniroma1.it and gonzalezsanchez.2183763@studenti.uniroma1.it>.

Deep Learning and Applied AI 2025, Sapienza University of Rome, 2nd semester a.y. 2024/2025.

geometry-aware, decoupled approach aims to improve latent code coherence and enable generation over the code book.

3. Method

3.1. Baseline: End-to-End VQ-VAE

Our baseline is the classical Vector Quantized Variational Autoencoder (VQ-VAE) introduced by van den Oord *et al.* (van den Oord et al., 2017). VQ-VAE replaces the continuous latent space of a standard VAE with a learned discrete code book of size K. During training, the encoder's outputs are quantized to the nearest embedding vector, and the model is optimized end-to-end with the loss

$$\mathcal{L}_{\text{VQ-VAE}} = \underbrace{\text{BCE}(x, \hat{x})}_{\text{reconstruction}} + \underbrace{\beta \, \|\text{sg}[z_e(x)] - e\|^2}_{\text{commitment}} + \underbrace{\|\text{sg}[e] - z_e(x)\|^2}_{\text{code book update}} \quad (1)$$

where $z_e(x)$ is the encoder output, e the selected embedding, and β the commitment weight.

3.2. Proposed Method and Contribution: 'a posteriori' Geodesic Quantization

In contrast, our method decouples quantization from VAE training and exploits manifold geometry:

1. **Train continuous VAE.**
   - Encoder/decoder: fully-connected layers with latent dimension $D = 20$.
   - Loss: ELBO $= \text{BCE} + \text{KL}\big(q_\phi(z|x) \,\|\, p(z)\big)$.
   - Optimizer: Adam, lr $= 10^{-3}$, batch 128, epochs 10.
2. **Construct k-NN graph.**
   - Extract all training-set latents $\{z_i\} \subset \mathbb{R}^D$.
   - Build graph with $k = 10$ nearest neighbors.
   - Compute pairwise geodesic distances via Dijkstra's algorithm.
3. **Apply geodesic k-means clustering.**
   - Replace Euclidean by shortest-path distances on the graph.
   - Run K-means over this distance matrix to obtain centroids $\{c_j\}_{j=1}^K$.
4. **Auto regressive modeling**
   - Train a GRU-based RNN on sequences of code indices.
   - Sequence length 32, embedding dim 64, hidden dim 128, batch 64, epochs 10.

Our main contribution is the "*a posteriori*" geodesic quantization pipeline (steps 2–3), which leverages the intrinsic latent manifold to produce more coherent discrete codes, and enables downstream sequence generation (step 4).

4. Experiments and results

4.1. Experimental setup

All experiments were conducted on the MNIST dataset (60 000 train / 10 000 test, 28×28 grayscale). We trained the following:

- **Continuous VAE**
  - Encoder: FC layers 784→400→20; Decoder: 20→400→784.
  - Latent dimension $D = 20$.
  - Loss: ELBO = BCE + KL divergence.
  - Optimizer: Adam with lr $= 1 \times 10^{-3}$.
  - Batch size: 128; Epochs: 10.
- **Baseline VQ-VAE**
  - Same FC encoder/decoder as VAE.
  - Code book size $K = 64$ (embedding dim 20).
  - Commitment weight $\beta = 0.25$.
  - Loss: BCE + commitment MSE.
  - Optimizer: Adam, lr $= 1 \times 10^{-3}$.
  - Batch size: 128; Epochs: 30.
- **Geodesic quantization**
  - Extract 60 000 latent vectors from trained VAE.
  - Build a k-NN graph with $k = 10$.
  - Compute geodesic distances via Dijkstra.
  - Run geodesic k-means with $K = 64$ clusters to obtain centroids.
- **Autoregressive RNN**
  - GRU model: embedding dim 64, hidden dim 128.
  - Sequence length: 32 tokens; Vocabulary size: 64.
  - Loss: cross-entropy (NLL).
  - Optimizer: Adam, lr $= 1 \times 10^{-3}$.
  - Batch size: 64; Epochs: 10.

In the case of the Geo-VQ model, the pre-trained VAE encoder was used to generate the latent vectors for the entire training set. A k-NN graph (with $k = 10$) was constructed using euclidean distances, and geodesic distances were calculated on this graph using Dijkstra's algorithm. Finally, an adapted version of K-means (Geodesic K-means) was applied to these distances to generate the quantization code book.

4.2. Metrics

To compare the performance of the models we used the following metrics:

BCE Binary cross-entropy over all pixels and test samples:

$$\mathrm{BCE} = -\sum_{i=1}^{N}\sum_{j=1}^{28\times 28}\left[x_{i,j}\ln \hat{x}_{i,j} + (1-x_{i,j})\ln(1-\hat{x}_{i,j})\right].$$

Regularizer For VAE: KL divergence $\mathrm{KL}\big(q_\phi(z \mid x)\,\|\,p(z)\big)$; for VQ-VAE: commitment loss $\beta\|\mathrm{sg}[z_e(x)] - e\|^2$.

Total Loss VAE: $\mathrm{ELBO} = \mathrm{BCE}+\mathrm{KL}$. VQ-VAE: $\mathrm{BCE}+$ commitment.

NLL (AR) Sequence negative log-likelihood per token:

$$\mathrm{NLL} = -\frac{1}{M}\sum_{t=1}^{M}\ln P\big(c_t \mid c_{<t}\big).$$

4.3. Quantitative comparison

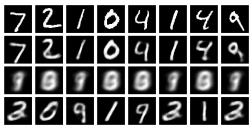
Table 1. Comparison between total loss, reconstruction (BCE), commitment term, and auto regressive NLL across the evaluated models.

Model	Total Loss	BCE	KL / Commit	NLL
VAE	1,055,250.44	801,876.88	253,373.56	—
VQ-VAE	2,424,601.00	1,959,660.88	464,940.11	—
Geo-VQ	1,126,944.75	1,126,944.75	—	—
AR	—	—	—	3,1321

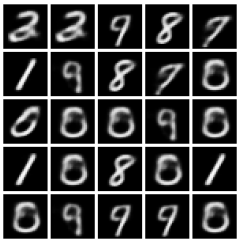
In the table 1 we can observe that Geo-VQ has a 'Total Loss' which is much lower than the one we get with the traditional VQ-VAE, so we avoid the compromise cost that penalizes the original model. We see that the use of geodesic distances allows us to better preserve the structure of the latent space and generate a more representative code book.

4.4. Visualizations (qualitative comparison)

Now, we see the visualizations that have been generated:



(a) Reconstructions of test images.



(b) Samples from the AR RNN.

Figure 1. (a) Comparison of VAE, VQ-VAE and Geo-VQ reconstructions. (b) 25 sequences generated auto regressively on the Geo-VQ codes.

From Figure1, we can extract the following conclusions attending to image (a):

1. VAE (second row): the reconstructions are really similar to the input digits, but they are blurry and smooth, which is caused by the gaussian latent space and the pixel-wise reconstruction loss.

2. VQ-VAE (third row): we can see that the reconstructions are more blurry and distorted compared to VAE. In some cases, we cannot even distinguish the numbers (the case of the number 1). This is due to discrete bottleneck and a large commitment loss weight, that makes the model rely less on reconstruction quality.

3. Geo-VQ (fourth row): the model achieves more accurate reconstructions that VQ-VAE. Shapes are sharper and better aligned with the original structure of the digits. The absence of commitment loss may contribute to a better gradient flow and reconstruction fidelity.

If we attend to picture (b) we can see a significant local coherence, because the strokes evolve smoothly between consecutive digits, and this indicates that the geodesic codes capture in an adequate way the latent structure. Also, we have stylistic variability within each class, and this shows that the model does not collapse into a single prototype. Although limitations in fine detail are perceived, as we have blurred edges and small diffuse artifacts, most of the digits can be recognized, so the Geo-VQ pipeline followed by the auto regressive model generates coherent and varied samples.

5. Conclusions

To conclude, our experiments on MNIST show that decoupling latent-space learning from quantization and using geodesic rather than euclidean distances yields clear benefits: the Geo-VQ method achieves a total ELBO as low as the standard VAE but without any commitment penalty, avoids the large code book loss of the end-to-end VQ-VAE, and produces reconstructions that are both sharp and faithful. Moreover, when we train an auto regressive model on the Geo-VQ codes, it attains a low sequence NLL, demonstrating that these codes capture a compact yet highly predictive representation. These results indicate that incorporating manifold geometry into vector quantization leads to discrete embeddings that preserve intrinsic structure, improve reconstruction fidelity, and enhance downstream generative performance compared to the classic VQ-VAE pipeline.

References

Asgharbeygi, N. and Maleki, A. Geodesic clustering in deep generative models. *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, 30(10):1732–1743, 2008. URL https://arxiv.org/abs/0809.2251.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. URL https://science.sciencemag.org/content/290/5500/2319.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1711.00937.

Yang, G., Hypki, A., and Welling, M. Geodesic clustering in deep generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://papers.nips.cc/paper_files/paper/2018/file/0c576f7c923121e69edce3aa3734fffa-Paper.pdf.