Geodesic Post-Hoc Quantization for Variational Autoencoders

Marco Galletti (2109043) [1]

Abstract

This project revisits the classic VQ-VAE by implementing a post-hoc quantization pipeline. This work compares an end-to-end trained VQ-VAE against a post-hoc method that uses a novel geodesic clustering algorithm based on a metric derived from the VAE decoder's geometry on the CIFAR-10 dataset. The results reveal that the inherent "training mismatch" of the proposed framework proves to be a significant bottleneck, with the VQ-VAE achieving superior results across all metrics.

1. Introduction

Generative models, particularly Variational Autoencoders (VAEs), are powerful tools for learning low-dimensional representations of complex data. The Vector Quantized VAE (VQ-VAE) extends this by learning a discrete latent space, which is often more suitable for downstream tasks such as autoregressive modeling. Standard VQ-VAEs learn this discrete representation end-to-end with the encoder and decoder. This work explores an alternative quantization pipeline, where a continuous-latent VAE is first trained, and a discrete codebook is then built by applying clustering algorithms to its latent space. The primary contribution is the implementation and evaluation of a geometrically-motivated a posteriori quantization method on the CIFAR-10 dataset, comparing it directly against a standard VQ-VAE to investigate the trade-offs between these two paradigms. The code for this project is available at `https://github.com/m4rch1n0/vqvae`.

2. Related Work

The project builds upon three main areas of research: discrete latent variable models, the geometric analysis of latent spaces, and graph-based clustering methods.

Discrete Latent Representations. The VQ-VAE (Van Den Oord et al., 2017) is the foundational model for learning discrete representations for generative tasks. It introduces the paradigm of learning a discrete codebook with the encoder and decoder, and uses a powerful autoregressive model over the resulting spatial grid of codes to generate high-fidelity images. The research in this project deliberately deviates from this by exploring a post-training quantization scheme, which separates the continuous representation learning from the codebook creation.

Geometry of VAE Latent Spaces. The core motivation for the geodesic method comes from works that model the latent space of a VAE as a Riemannian manifold. Arvanitidis et al. (Arvanitidis et al., 2017) established the use of the pull-back metric tensor, defined by the decoder's Jacobian ($M(z) = J(z)^T J(z)$), to better represent the geometry of the data manifold. They argued for using geodesic distances, rather than Euclidean ones, for tasks like interpolation and clustering. Shao et al. (Shao et al., 2018) further developed practical methods for geodesic computation, though they noted that for some image VAEs, the learned manifolds can be nearly flat, making Euclidean distance a reasonable approximation. This work leverages these geometric insights directly, using Jacobian-vector product to approximate local geodesic distances for clustering.

Graph-Based Geodesic Approximation. The specific technique of approximating geodesic distances by computing shortest paths on a k-Nearest Neighbors (k-NN) graph was popularized by Isomap (Tenenbaum et al., 2000) in the context of manifold learning. This approach provides a computationally feasible way to estimate global manifold distances from local neighborhood information. The methods presented here are a direct application of this principle: a k-NN graph is constructed and shortest-path distances (via Dijkstra's algorithm) are used as the metric for the k-medoids clustering algorithm, with the key novelty being the re-weighting of the graph's edges with the learned Riemannian metric.

3. Method

The pipeline consists of three main stages: continuous representation learning, post-hoc quantization, and autoregressive modeling.

[1]Sapienza University of Rome Email: Marco <galletti.2109043@studenti.uniroma1.it>.

Machine Learning 2025, Sapienza University of Rome, 2nd semester a.y. 2024/2025.

3.1. Model Architectures[1]

Spatial VAE. The VAE encoder consists of three convolutional layers (with 64, 128, and 256 channels respectively, stride 2, and Batch Normalization) that downsample the input image to a 4×4 spatial grid. Two final 1×1 convolutions produce the mean and log-variance for a 32-dim latent space at each spatial location. The decoder mirrors this architecture using transposed convolutions to reconstruct the image.

Transformer. A decoder-only Transformer is used for autoregressive modeling of the quantized latent codes. The model has 4 layers, 4 attention heads, an embedding dimension of 256, and is trained on flattened 16-token sequences.

3.2. Training Details

The `SpatialVAE` was trained for 200 epochs using the AdamW optimizer with a learning rate of 10^{-3}, a weight decay of 10^{-5}, and a cosine learning rate schedule. The Transformer model was also trained for 200 epochs with AdamW, but with a learning rate of 3×10^{-4} and a weight decay of 0.01. The batch size was 256.

3.3. Post-Hoc Quantization Method

After training the VAE, the mean latent vectors (μ) are extracted from the training set and a discrete codebook is built using a geometrically rigorous clustering scheme. This is the **Full Riemannian Geodesic Clustering** method. The process is as follows:

1. **k-NN Graph Construction:** A k-Nearest Neighbors graph (with k=50) is constructed from the flattened latent vectors to establish local connectivity.

2. **Riemannian Re-weighting:** *All* unique edges in this graph are re-weighted using a metric derived from the VAE decoder's geometry, approximated via efficient Jacobian-vector products.

3. **Geodesic K-Medoids:** K-Medoids clustering (with K=512) is performed on the re-weighted graph. Distances are computed as the shortest paths on the graph using **Dijkstra's algorithm**, and the process is initialized with a K-Means++ strategy. Clustering is performed on the largest connected component (LCC) of the graph to ensure all points are mutually reachable.

3.4. Autoregressive Modeling

Once the codebook is built, the spatial latent grids for each training image are quantized, converting them into sequences of discrete integer codes. A decoder-only Transformer model is then trained to predict the next code in a sequence, **conditioned on the previous ones**. This allows for autoregressive generation of novel sequences of codes, which can then be mapped back to latent vectors via the codebook and rendered into images by the VAE decoder.

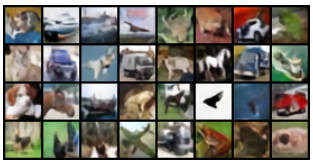
4. Experiments and Results

The evaluation was performed on the CIFAR-10 dataset. Performance was assessed using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

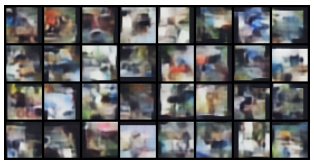
The results (Table 1) suggest the superiority of the end-to-end approach, which outperforms the post-hoc method across all key metrics. This suggests the solidity of jointly training the encoder, decoder, and codebook on complex data distributions.

Table 1. CIFAR-10 Method Comparison.[3]

Method	PSNR (dB)	SSIM	LPIPS
Post-Hoc (Full Riemannian)	8.72	0.005	0.564
Baseline VQ-VAE	**9.49**	**0.019**	**0.475**



(a) Baseline VQ-VAE (Reconstructions)



(b) Post-Hoc Spatial Geodesic (Generated)

Figure 1. Qualitative comparison on CIFAR-10. (a) Reconstructions from the end-to-end VQ-VAE. (b) Samples generated by the Transformer in the post-hoc spatial geodesic pipeline.

5. Discussion and Conclusions

On CIFAR-10, the "training mismatch" in post-hoc methods appears to create a performance bottleneck that is overcome by the joint, end-to-end optimization of a standard VQ-VAE. In conclusion, this project reveals a complex trade-off between geometric fidelity and reconstruction performance. While leveraging a metric derived from the VAE decoder's geometry can substantially reduce latent-space quantization error (as detailed in the Appendix), this does not guarantee superior image quality on complex datasets. For CIFAR-10, the end-to-end trained VQ-VAE achieves superior results. Future work could address the limitation of the post-hoc approach by fine-tuning the VQ-VAE decoder on the discrete codebook, potentially combining the benefits of geometric clustering with the performance of a jointly-optimized system. A full comparative analysis on the simpler Fashion-

[1]Model implementations are available at `vqvae/src/models/`. The vanilla VAE used in the Appendix is also available here.

[3]Metrics are computed consistently across all methods. While absolute values may be implementation-dependent, the relative performance ranking is valid.

MNIST dataset, which shows different performance characteristics, is provided in the Appendix.

References

Arvanitidis, G., Hansen, L. K., and Hauberg, S. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*, 2017.

Shao, H., Kumar, A., and Thomas Fletcher, P. The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 315–323, 2018.

Tenenbaum, J. B., Silva, V. d., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.