Variational Autoencoders for Hybrid Language Music Clustering

Mushfique Tajwar
Department of Computer Science and Engineering
BRAC University
mushfique.tajwar@g.bracu.ac.bd

January 6, 2026

Abstract

This work presents a lightweight, reproducible unsupervised learning pipeline for clustering music tracks using variational autoencoders (VAEs). Audio tracks are converted into fixed-length log-mel spectrogram vectors and encoded into a low-dimensional latent space using an MLP-based VAE (Easy/Medium) or a β-VAE (Hard). The latent representations are clustered using K-Means and other standard clustering algorithms (Agglomerative Clustering and DBSCAN), and cluster quality is evaluated with internal indices (Silhouette, Calinski–Harabasz, Davies–Bouldin) and optionally label-based metrics (ARI/NMI/Purity) inferred from GTZAN-style filename prefixes. A PCA + K-Means baseline is included, and embeddings are visualized with t-SNE/UMAP.

Code availability. The code for this project is available at https://github.com/mushfique-tajwar/Variational-Autoencoders-Hybrid-Music-Clustering.

1 Introduction

Unsupervised clustering of music is useful for discovery, organization, playlist generation, and understanding latent structure in large audio collections. However, raw audio waveforms are high-dimensional and difficult to cluster directly. Representation learning approaches, especially variational autoencoders (VAEs), provide a principled method to learn compact, approximately continuous latent spaces suitable for downstream clustering.

This project implements a full end-to-end pipeline for VAE-based clustering on an audio dataset in a GTZAN-like format (e.g., `blues.00000.au`). The repository supports three incremental tasks: (i) an Easy task using an MLP-VAE on flattened log-mel features with K-Means and PCA baseline, (ii) a Medium task allowing optional audio+lyrics feature fusion and multiple clustering methods, and (iii) a Hard task implementing a β-VAE for stronger regularization / disentanglement.

2 Related Work

VAEs [4] learn generative latent-variable models by maximizing an evidence lower bound (ELBO). β-VAEs [3] modify the ELBO to encourage disentangled representations by scaling the KL term. For clustering evaluation without labels, internal quality indices such as Silhouette [5], Calinski–Harabasz [1], and Davies–Bouldin [2] are commonly used. With labels, ARI and NMI provide agreement measures between predicted clusters and ground truth.

3 Method

3.1 Audio feature extraction

Given an audio file $x(t)$, a mel spectrogram is computed using[1]:

$$S = \log_{10}(\epsilon + \text{MelSpec}(x)), \tag{1}$$

where ϵ is a small constant to avoid taking log of zero. The input size is fixed by padding or truncating the time dimension to `target_frames` and flattening to a vector of dimension $d = n_{\text{mels}} \times$ `target_frames`. Features are standardized per dimension using dataset mean and standard deviation.

3.2 Lyrics features and multimodal fusion (Medium)

When lyrics files are present, TF-IDF features with unigrams and bigrams are computed and concatenated with audio features. Missing lyrics are treated as empty strings.

3.3 VAE model

The Easy/Medium tasks use an MLP-VAE (`src/models.py`) with an encoder network producing μ and $\log \sigma^2$:

$$q_\phi(z \mid x) = \mathcal{N}\big(z; \mu_\phi(x), \text{diag}(\sigma^2_\phi(x))\big), \tag{2}$$

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

The decoder reconstructs $\hat{x}$ from z using a symmetric MLP.

Objective. The model is trained by minimizing MSE reconstruction with KL regularization:

$$\mathcal{L} = \underbrace{\text{MSE}(x, \hat{x})}_{\mathcal{L}_{\text{recon}}} + \beta \underbrace{\text{KL}\left(q_\phi(z \mid x) \,\|\, p(z)\right)}_{\mathcal{L}_{\text{KL}}}, \qquad p(z) = \mathcal{N}(0, I). \tag{4}$$

The Hard task uses a β-VAE by setting $\beta > 1$.

3.4 Clustering and visualization

After training, each track is encoded into its posterior mean μ and the resulting embeddings are clustered. K-Means is applied in all tasks, and in Medium Agglomerative Clustering and DBSCAN are additionally evaluated. For visualization, latent vectors are embedded to 2-D using t-SNE or UMAP.

4 Experiments

4.1 Dataset

Experiments use the repository's `dataset/audio` directory containing `.au` files with GTZAN-style naming. When `-use_labels` is enabled, a coarse label is inferred as the filename prefix before the first dot (e.g., `blues` from `blues.00000.au`).

Dataset sources. The datasets used in this project were downloaded from Kaggle:

(1) audio (GTZAN genre collection): https://www.kaggle.com/datasets/carlthome/gtzan-genre-collection;

(2) English lyrics: https://www.kaggle.com/datasets/suraj520/music-dataset-song-information-and-lyrics; and

(3) Bangla lyrics: https://www.kaggle.com/datasets/meherabhasansajid/bangla-song-lyrics-dataset-with-genres-and-artists.

[1] Implementation in `src/data.py` uses `librosa.feature.melspectrogram`.

4.2 Training details

Across tasks, models are trained using Adam with learning rate 10^{-3} and mini-batch sizes of 32 on CPU by default. Unless specified, `latent_dim=16` is used and models are trained for 30–50 epochs.

4.3 Evaluation metrics

For internal clustering quality, the following are reported: Silhouette score, Calinski–Harabasz index, and Davies–Bouldin index. When inferred labels are enabled, ARI, NMI, and cluster purity are also reported.

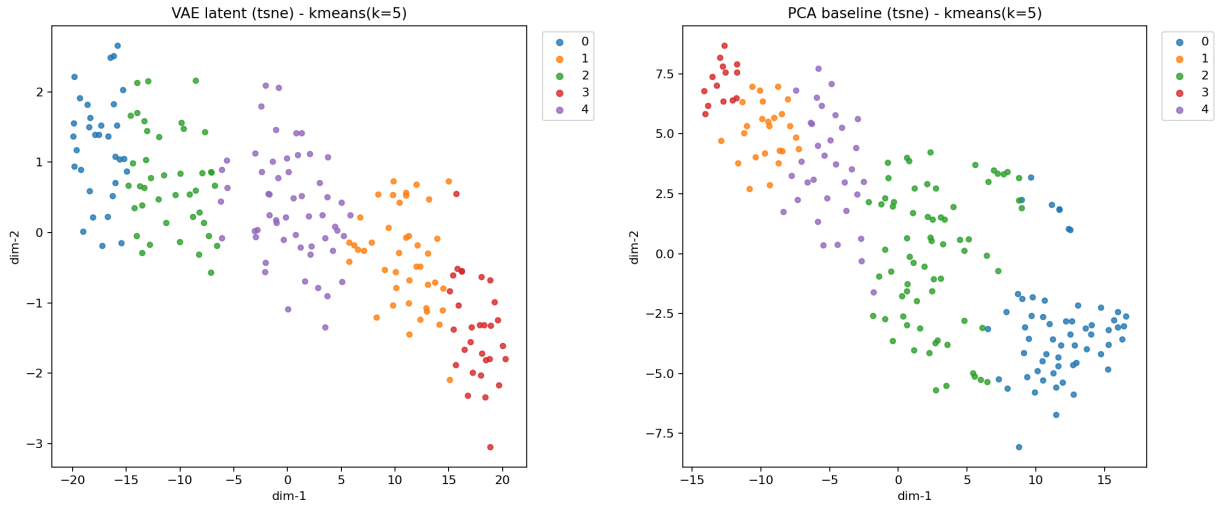
5 Results

This repository writes results to `results/easy`, `results/medium`, and `results/hard`. Tables are auto-populated from the CSV files produced by the scripts (e.g., `results/easy/metrics_easy.csv`).

5.1 Easy task: VAE + K-Means vs PCA baseline

Table 1: Easy task metrics from `results/easy/metrics_easy.csv`.

Method	Silhouette	CH	DB	ARI	NMI	Purity
vae+kmeans	0.378	424.69	0.829	0.092	0.139	0.740
pca+kmeans	0.237	125.21	1.574	0.113	0.167	0.740



(a) VAE latent (t-SNE)

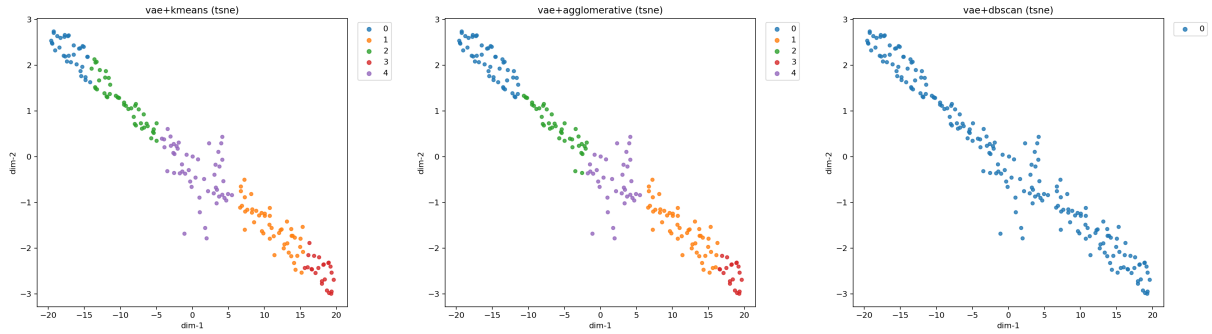
(b) PCA baseline (t-SNE)

Figure 1: Easy task cluster visualizations (paths assume running from `report/`).

5.2 Medium task: feature fusion and clustering methods

Table 2: Medium task metrics from `results/medium/metrics_medium.csv`.

Method	Silhouette	CH	DB	ARI	NMI	Purity
vae+kmeans	0.493	700.44	0.594	0.094	0.143	0.740
vae+agglomerative	0.475	636	0.596	0.129	0.174	0.740
vae+dbscan	–	–	–	0.000	0.000	0.500



(a) VAE + K-Means (b) VAE + Agglomerative (c) VAE + DBSCAN

Figure 2: Medium task latent-space visualizations (t-SNE) for different clustering algorithms.

5.3 Hard task: β-VAE

Table 3: Hard task metrics from `results/hard/metrics_hard.csv`.

Method	Silhouette	CH	DB	ARI	NMI	Purity
beta-vae+kmeans	0.253	216.96	1.106	0.053	0.133	0.745

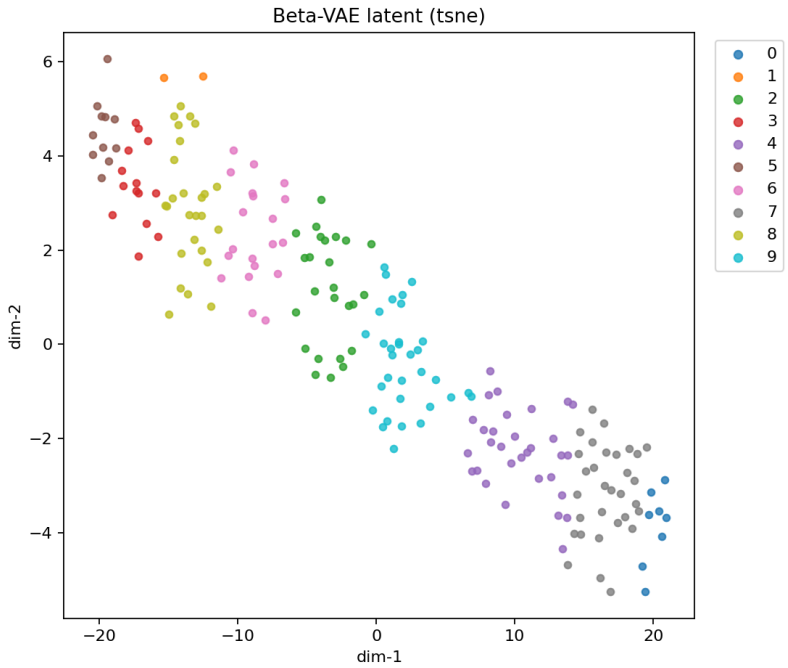


Figure 3: β-VAE latent visualization (t-SNE) (path assumes running from `report/`).

6 Discussion

The Easy-task setup demonstrates that a VAE can learn a compact embedding that is more suitable for clustering than the raw flattened spectrogram space, while PCA provides a linear baseline. In the Medium task, adding lyrics (TF-IDF) may improve or degrade clustering depending on lyrics availability/quality and the relative scaling of modalities; this highlights the importance of careful multimodal fusion. In the Hard task, increasing β strengthens the KL constraint, typically producing smoother, more factorized latent spaces but with a trade-off in reconstruction quality.

Limitations. This implementation favors simplicity and reproducibility over state-of-the-art performance: the VAE is fully-connected (not convolutional), the audio representation is fixed-length and does not model temporal dynamics explicitly, and inferred labels from filenames only approximate ground truth.

7 Conclusion

This work implements an end-to-end VAE-based music clustering pipeline with clear baselines, multiple clustering options, and standard quality metrics. The repository produces reproducible quantitative tables and qualitative visualizations. Future work can explore convolutional/audio-transformer encoders, learned text embeddings, contrastive learning, and stronger multimodal fusion methods.

References

[1] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[2] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[3] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017.

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[5] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.