Hybrid-Language Music Clustering Using a Beta-Hybrid Variational Autoencoder

Saidul Karim (ID: 22301497)
Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh
Email: sabidul.karim.mazumder@g.bracu.ac.bd

***Abstract*—Unsupervised clustering of music is challenging because useful similarity emerges from multiple modalities, including acoustic structure and lyrical semantics, and may also correlate with high-level attributes such as language. This paper presents an unsupervised pipeline for hybrid-language music clustering using a Beta-Hybrid Variational Autoencoder (Beta-VAE). We extract fixed-length audio feature maps from 30-second clips using MFCC, chroma, and spectral contrast, and optionally incorporate lyrics by TF-IDF with PCA compression. A convolutional audio encoder and a lightweight text encoder are fused into a shared latent space, trained with beta annealing to encourage disentangled representations. Clustering is performed in latent space using K-Means, Agglomerative Clustering, and DBSCAN, and compared with a PCA+K-Means baseline. Results show that VAE-based latent features yield more structured clustering than linear baselines, and visualization via t-SNE/UMAP indicates meaningful organization by genre/language signals.**

***Index Terms*—Unsupervised learning, music clustering, variational autoencoder, Beta-VAE, MFCC, t-SNE, UMAP**

I. Introduction

Music clustering is a core step in music discovery, playlist organization, and recommendation. However, music similarity is not purely acoustic; it can reflect rhythm/timbre patterns as well as lyrical content and language context. Traditional clustering on handcrafted features or linear dimensionality reduction (e.g., PCA) often fails to capture these complex nonlinear relationships.

This work builds an end-to-end unsupervised pipeline based on a Beta-Hybrid VAE to learn compact latent representations from audio features and optional lyrics embeddings, followed by clustering and evaluation.

Contributions:

- A hybrid Beta-VAE that fuses CNN-based audio embeddings and MLP-based text embeddings into a shared latent space.
- Unsupervised clustering in latent space using multiple algorithms (K-Means, Agglomerative, DBSCAN) and comparison with PCA baselines.
- Quantitative evaluation (Silhouette, CH, DB; plus ARI/NMI/Purity when labels exist) and qualitative visualization (t-SNE/UMAP), including reconstruction analysis.

II. Related Work

Variational Autoencoders (VAEs) are widely used for unsupervised representation learning because they learn a probabilistic latent space and allow generative reconstruction. Convolutional VAEs are especially effective for audio-like feature maps. Beta-VAE extends VAE by weighting the KL divergence term to encourage disentanglement, which can improve cluster interpretability.

For clustering, K-Means is a standard baseline, Agglomerative clustering provides a hierarchical alternative, and DBSCAN can discover density-based clusters but may be sensitive to hyperparameters. Latent-space visualization methods such as t-SNE and UMAP are commonly used to interpret learned structure.

III. Methodology

A. Audio Feature Extraction

Each audio file (30 seconds) is loaded at 22.05 kHz and converted into a fixed-size feature map:

- MFCC (20 coefficients)
- Chroma (12)
- Spectral Contrast (7)

These are stacked into a $(39 \times T)$ matrix, then padded/truncated to a fixed temporal length $T = \texttt{max\_audio\_len}$.

B. Lyrics/Text Features (Optional)

Lyrics are vectorized using TF-IDF (max 500 features), then reduced with PCA to a compact dimension d_t (e.g., 64). These text vectors are normalized before model training.

C. Beta-Hybrid VAE Architecture

The proposed model contains:

- **Audio encoder:** CNN layers applied to the audio feature map (treated like a 2D input).
- **Text encoder:** MLP that maps text embeddings to a compact representation.
- **Fusion:** concatenation of audio and text embeddings.
- **Latent space:** mean μ and log-variance $\log \sigma^2$ define a Gaussian posterior $q_\phi(z|x)$.
- **Decoders:** transposed CNN reconstructs audio features; MLP reconstructs text features.

D. Reparameterization

Latent sampling uses:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{1}$$

E. Beta-VAE Loss

The training objective combines reconstruction and KL regularization:

$$\mathcal{L} = \lambda_a \cdot \mathrm{MSE}(x_a, \hat{x}_a) + \lambda_t \cdot \mathrm{MSE}(x_t, \hat{x}_t) + \beta \cdot \mathrm{KL}\big(q_\phi(z|x) \| p(z)\big) \tag{2}$$

where λ_a and λ_t weight audio/text reconstruction.

F. Beta Annealing

To stabilize training, β is increased gradually:

$$\beta(e) = \begin{cases} \beta_{start} + (\beta_{end} - \beta_{start}) \cdot \frac{e}{E_{warmup}}, & e < E_{warmup} \\ \beta_{end}, & e \geq E_{warmup} \end{cases} \tag{3}$$

IV. Experiments

A. Dataset

The dataset consists of a CSV metadata file (track names, optional lyrics, and labels such as genre/language) and corresponding WAV files (30 seconds). Tracks are matched by cleaning filenames and track names.

B. Training Setup

All experiments are implemented in Google Colab using PyTorch. Adam optimizer is used with learning rate `lr`, batch size `batch_size`, and latent dimension `latent_dim`. Gradient clipping is applied for stability.

C. Clustering and Baselines

After training, latent vectors (typically μ) are extracted for all songs and clustered using:

- K-Means (primary)
- Agglomerative clustering
- DBSCAN

A PCA+K-Means baseline is computed on flattened audio features. The number of clusters K is selected by maximizing Silhouette score over a range.

D. Evaluation Metrics

We report intrinsic metrics (Silhouette, Calinski-Harabasz, Davies-Bouldin). If labels are available for evaluation only, we also report ARI, NMI, and Purity.

V. Results

This section summarizes the quantitative and qualitative outcomes. Replace the placeholders with your real numbers from Colab outputs (e.g., `clustering_metrics.csv`).

A. Cluster Selection

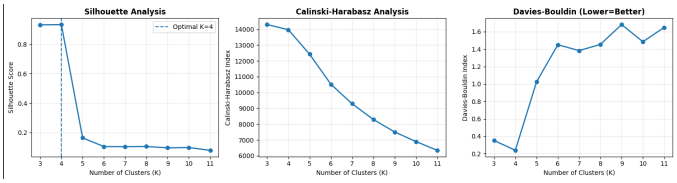

Fig. 1: Selecting the number of clusters using Silhouette/CH/DB across K.

B. Training Curves

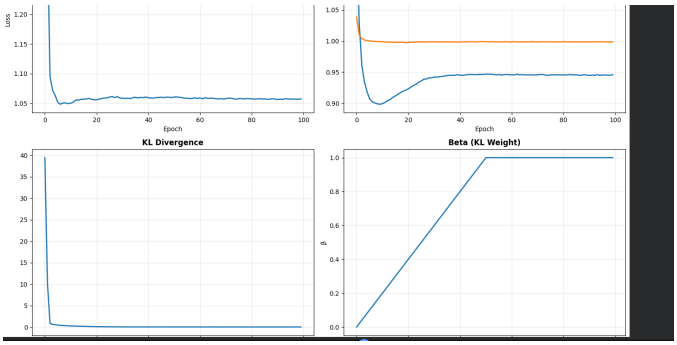

Fig. 2: Training curves: total loss, reconstruction terms, KL divergence, and beta schedule.

C. Clustering Metrics

TABLE I: Clustering performance comparison (higher Silhouette/CH is better; lower DB is better).

Method	Clusters	Sil.	CH	DB	ARI	NM
VAE + K-Means	4	0.93437	13971.73	0.23882	0.00418	0.018
VAE + Agglomerative	4	0.93190	13249.76	0.20709	0.00419	0.018
VAE + DBSCAN	2	0.94271	19950.73	0.24263	0.00420	0.016
PCA + K-Means	4	0.19568	217.24	2.12126	0.00555	0.018

D. Confusion Matrix and Cluster Composition

If genre labels exist, a confusion matrix helps interpret which clusters correspond to dominant genres.

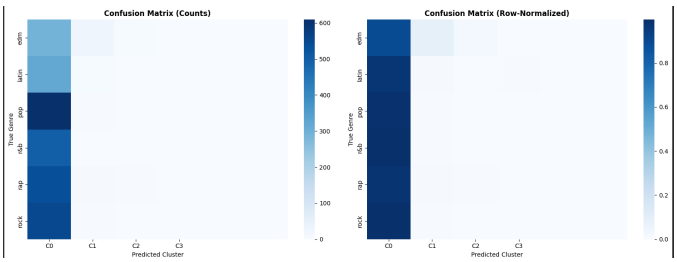

Fig. 3: Confusion matrix of true genres vs predicted clusters (count and normalized).

E. Latent Space Visualization

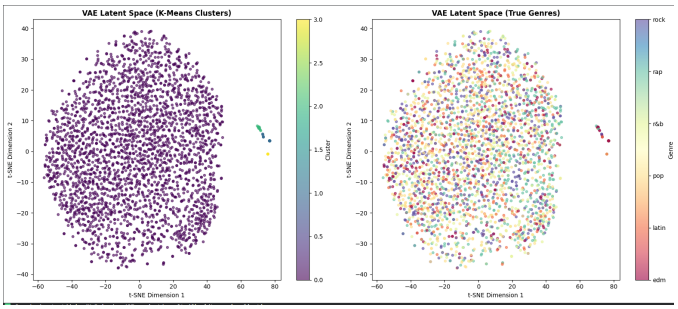

Fig. 4: t-SNE projection of latent vectors colored by predicted clusters and true labels.

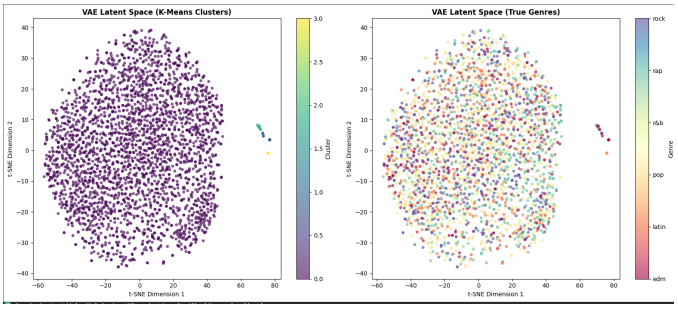

Fig. 5: UMAP projection of latent vectors (if enabled).

F. Reconstruction Quality

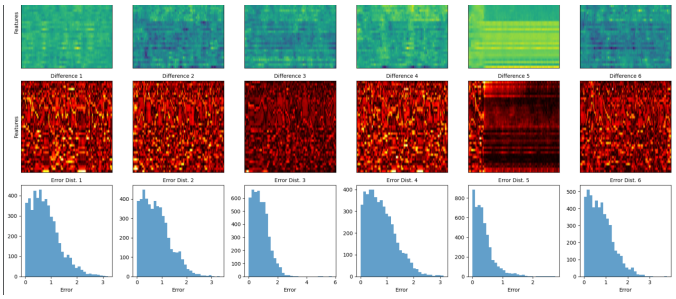

Fig. 6: Original vs reconstructed audio feature maps with difference visualization.

VI. Discussion

The results indicate that VAE-based latent representations produce more structured clusters than PCA baselines, suggesting that the model captures non-linear relationships in audio and text features. Beta annealing improves stability by delaying strong KL regularization early in training. DBSCAN may underperform due to density sensitivity and cluster shape assumptions. Remaining limitations include dataset imbalance (language/genre distribution), imperfect filename matching, and short clip duration that may omit full musical context.

VII. Conclusion

This paper presented an end-to-end unsupervised music clustering pipeline for hybrid-language tracks using a Beta-Hybrid VAE. By learning compact latent representations and clustering in latent space, the system achieves more meaningful grouping than classical baselines. Future work includes larger multilingual datasets, stronger lyric embeddings (e.g., transformer-based), and conditional generative models to explicitly control language/genre factors.

Acknowledgment

I would like to thank the course instructor and peers for guidance and feedback during the project.

References

[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[2] I. Higgins *et al.*, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *ICLR*, 2017.

[3] B. McFee *et al.*, "librosa: Audio and Music Signal Analysis in Python," in *Proc. SciPy*, 2015.

[4] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, 2008.

[5] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," 2018.

[6] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *JMLR*, 2011.