UNIVERSIDAD TORCUATO DI TELLA
TD VI: Inteligencia Artificial

TP 4: Encodeador de música

Integrantes:
Isabel Núñez
Camilo Suárez
Valentina Vitetta

5 de Diciembre del 2024

Contents

1 Introducción

El objetivo de este trabajo práctico es desarrollar un modelo basado en redes neuronales capaz de encodear canciones en un espacio latente. Estos vectores latentes deben ser compactos para reducir la dimensionalidad de las canciones, pero también deben permitir la reconstrucción del audio original con una calidad aceptable.

Además, se busca analizar los vectores latentes obtenidos mediante métodos no supervisados para identificar relaciones entre las canciones representadas en este espacio latente. Este análisis permitirá evaluar si el modelo puede aprender representaciones significativas que reflejen similitudes estructurales o de género entre las canciones.

Por último, se evalúa la capacidad del modelo para generalizar más allá del dataset utilizado. Esto incluye probar su desempeño al encodear música nueva, fuera del conjunto de entrenamiento, y explorar la posibilidad de realizar modificaciones sobre los vectores latentes para generar canciones nuevas.

El presente informe describe los experimentos realizados, los resultados obtenidos, el mejor modelo alcanzado y las limitaciones encontradas en relación con estos objetivos.

2 Encodear la canción en un vector latente

Para encodear canciones en vectores más pequeños que luego permitan su reconstrucción sin perder su estructura, decidimos realizar diversos experimentos. Como función de pérdida, utilizamos MSE, junto con el optimizador ADAM. Al no tener una métrica de accuracy debido a que no estamos prediciendo nada, decidimos guiar la elección del mejor modelo utilizando la loss en validación, puesto que notamos que una menor pérdida implicaba una mejor reconstrucción de las canciones.

En primer lugar, decidimos intentar hacer un autoencoder basándonos en el mejor modelo del trabajo práctico anterior, el cual consistía en tres capas convolucionales, tres reducciones de dimensionalidad con max pooling y cuatro hidden layers, con funciones de activación ELU y Shifted ReLU. No obstante, al adaptar el modelo para que funcionara como un autoencoder, las canciones resultantes eran pitidos ininteligibles, con una pérdida de 0.05 aproximadamente.

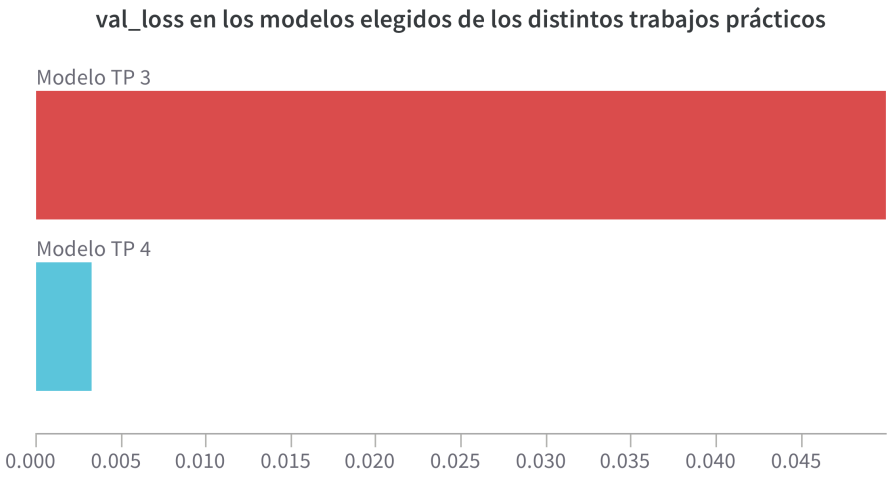

Figure 1: Comparación entre los modelos elegidos en cada trabajo práctico

De esta manera, decidimos cambiar completamente la arquitectura del autoencoder. Optamos por comenzar a experimentar con una convolución y una convolución traspuesta, seguidas de la función de activación tanh para que los valores de la salida se encuentren entre -1 y 1, tal como las canciones de entrada. Al experimentar con distintos tamaños de vectores latentes, lo cual lo hicimos modificando el kernel, padding, stride y la cantidad de feature maps, llegamos al mejor modelo propuesto. Esta arquitectura logra alcanzar un vector latente de dimensión 64 x 216 = 13824. Este es el tamaño más pequeño que encontramos en el que las canciones reconstruidas no pierden su estructura y siguen siendo "escuchables" ya que con vectores latentes de menor dimensión, las reconstrucciones se encontraban saturadas.

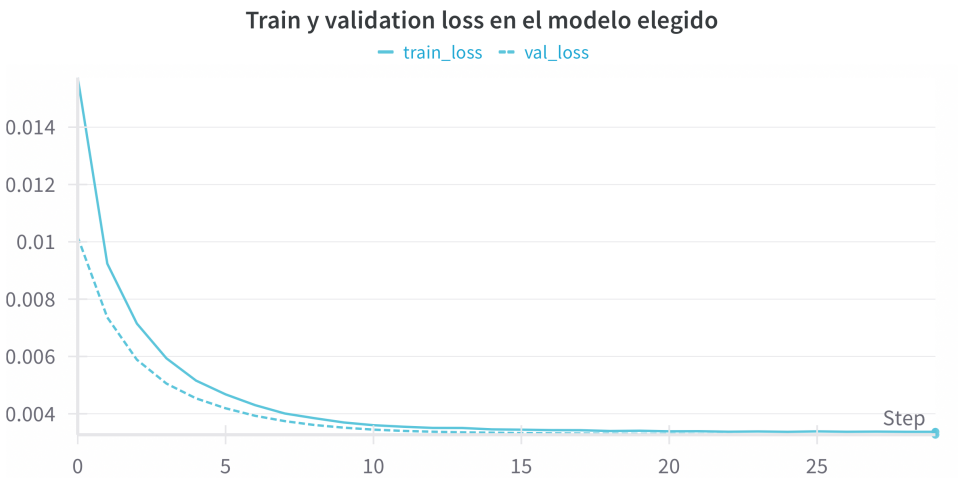

Figure 2: Mejor modelo - Train y validation loss

Luego, realizamos una serie de experimentos agregando capas convolucionales, deconvolucionales o fully-connected junto con funciones de activación como ReLU, con el objetivo de aumentar la capacidad del modelo para capturar características más complejas. Sin embargo, estos modelos no lograron mejorar los resultados encontrados. Además, notamos que las canciones reconstruidas perdían claridad, presentando distorsiones que hacían que las reconstrucciones fueran menos "escuchables".

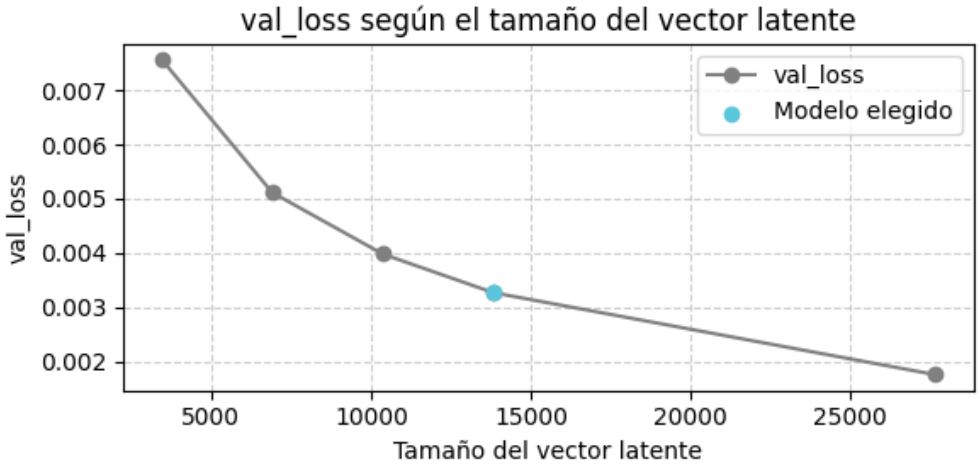

Figure 3: Ganancia de definición a medida que aumenta el tamaño del vector latente

En consecuencia, decidimos mantener la arquitectura más simple pero efectiva, con una sola capa convolucional y una capa deconvolucional, ya que esta alcanzó el mejor balance entre tamaño del vector latente (13824) y calidad en la reconstrucción de las canciones. Como fue mencionado, durante el proceso de experimentación, utilizamos la pérdida en el conjunto de validación (0.00328

para el modelo elegido) como métrica principal para seleccionar el modelo más adecuado. Una vez elegido este modelo, lo evaluamos en el conjunto de test, obteniendo una loss de 0.00338, lo cual confirma la efectividad de esta arquitectura y su capacidad para generalizar.

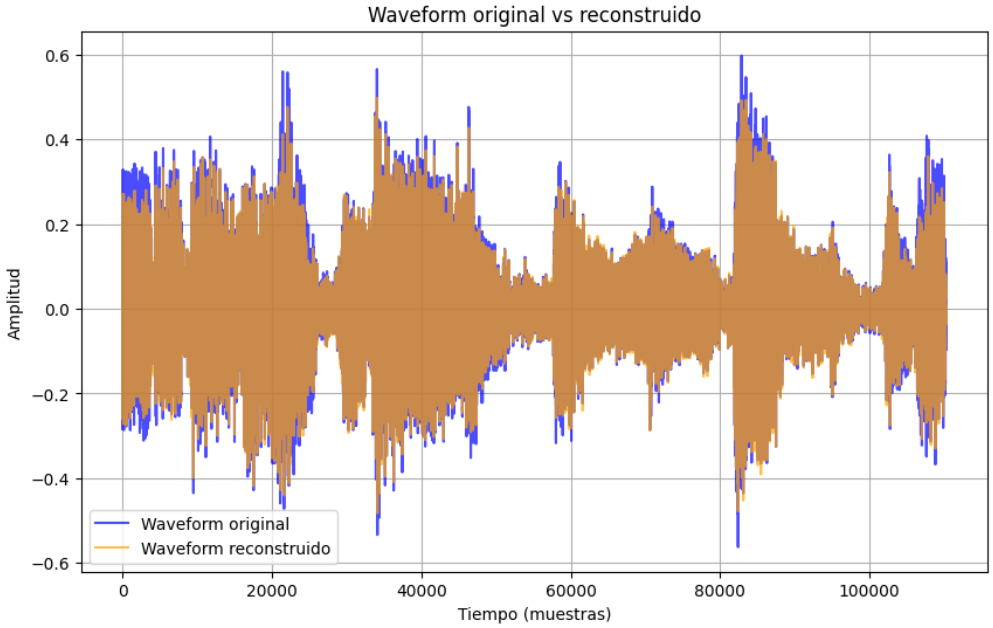

Figure 4: Reconstrucción del waveform con el modelo seleccionado

3 Análisis de Espacio Latente

Para realizar el análisis del espacio latente, trabajamos con tres configuraciones diferentes. Inicialmente utilizamos el tamaño original propuesto en el inciso anterior, con una dimensión de 64 x 216. Luego, creamos un espacio latente reducido, aproximadamente de la mitad del tamaño original de 32 x 216, y finalmente un espacio ampliado, con el doble de dimensiones de 128 x 216. Al analizar las pérdidas de entrenamiento y validación, observamos que estas disminuyen conforme aumenta el tamaño del espacio latente, lo que evidencia una mayor capacidad para capturar información de los audios en configuraciones más grandes.

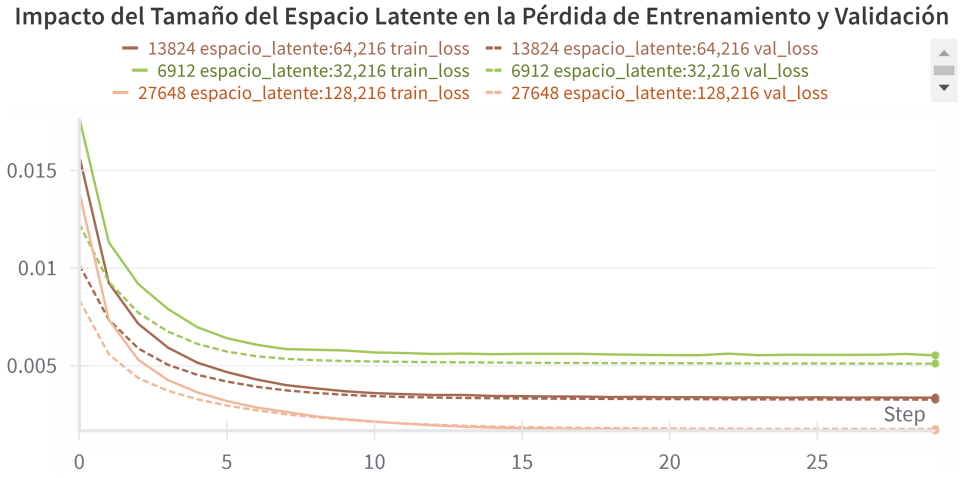

Figure 5: Comparación entre los espacios latentes

Estas configuraciones nos permitieron explorar cómo la variación en el tamaño del espacio latente afecta a la representación de las canciones y su análisis posterior utilizando métodos no supervisados como clustering y reducción dimensional con PCA.

Inicialmente, aplicamos el método del codo para determinar el número óptimo de clusters. Los tres tamaños de espacio latente produjeron resultados similares en términos de agrupación. Por esta razón, optamos por trabajar con 8 clusters y aplicar k-means. Sin embargo, al realizar el clustering, este enfoque mostró que los datos no se diferenciaban claramente. La mayoría de los vectores terminaban agrupados en un gran cluster principal, mientras que los restantes formaban clusters muy pequeños, compuestos por solo una o dos canciones. Indicando una limitada capacidad del espacio latente para capturar patrones distintivos.

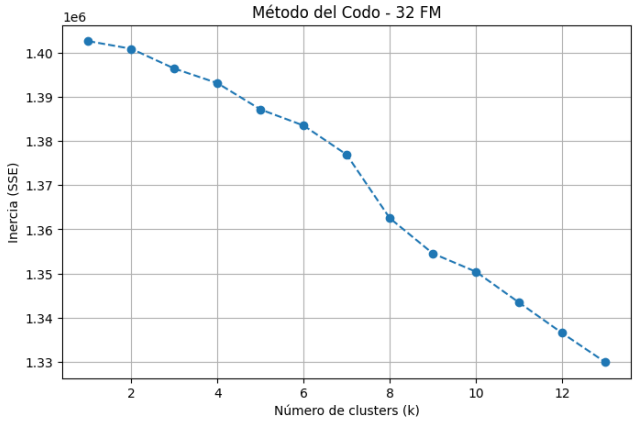

Figure 6: Método del codo para espacio latente de 32 FM

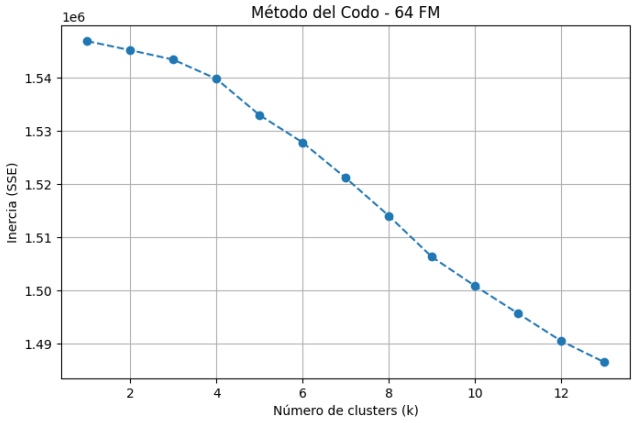

Figure 7: Método del codo para espacio latente de 64 FM

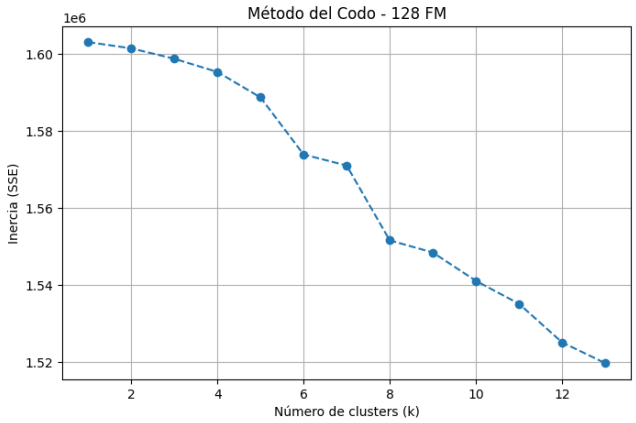

Figure 8: Método del codo para espacio latente de 128 FM

Al analizar los clusters que contenían únicamente dos canciones, observamos que, en todos los casos, estas correspondían a canciones repetidas en el dataset, como los pares de índices (212 y 10), (83 y 572), y (303 y 709). Este patrón se repitió para los 3 espacios latentes, destacando que el modelo logró identificar y relacionar correctamente las canciones idénticas, agrupándolas dentro del mismo cluster. Sin embargo, en los clusters con un único elemento o con una gran cantidad de elementos, no logramos identificar patrones claros ni relaciones significativas que permitieran extraer información relevante sobre el dataset.

Cluster	**Cantidad de puntos**
1	1
2	782
3	1
4	1
5	1
6	1
7	2
8	1

Table 1: Distribución de puntos por cluster para 128 FM

Para profundizar e intentar encontrar relaciones en el espacio latente, aplicamos PCA con dos componentes principales a cada uno de los espacios analizados. Este método mostró un patrón común en los tres tamaños del espacio latente. Dos outliers (índices 303 y 148) aparecían de forma consistente en la representación de la variabilidad. Al reproducir estas canciones, descubrimos que compartían características particulares, como velocidades altas, lo que podría explicar su comportamiento atípico en el espacio latente.

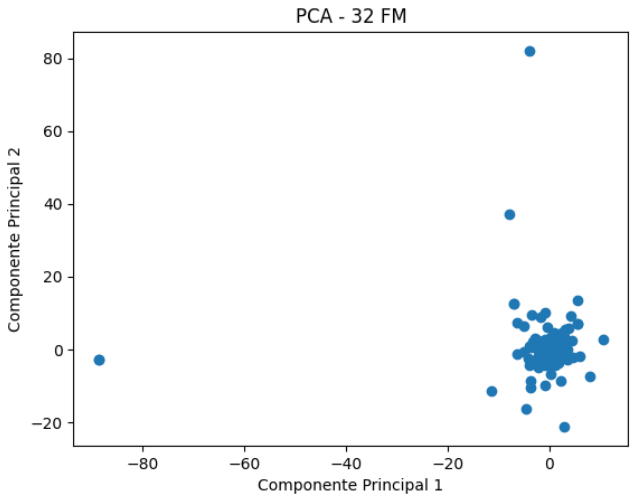

Figure 9: PCA para espacio latente de 32 FM

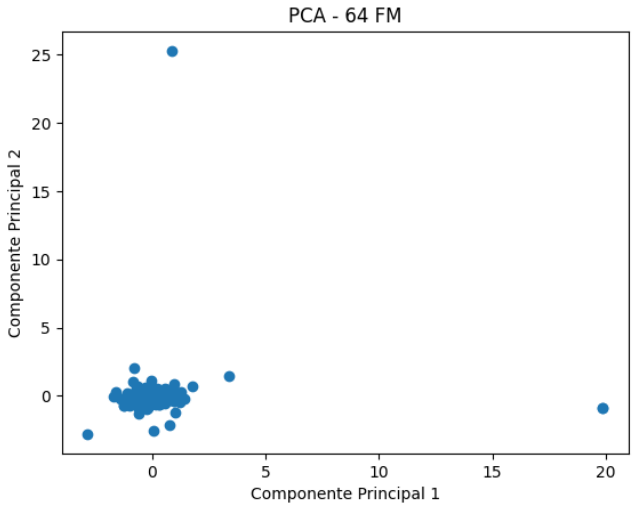

Figure 10: PCA para espacio latente de 64 FM

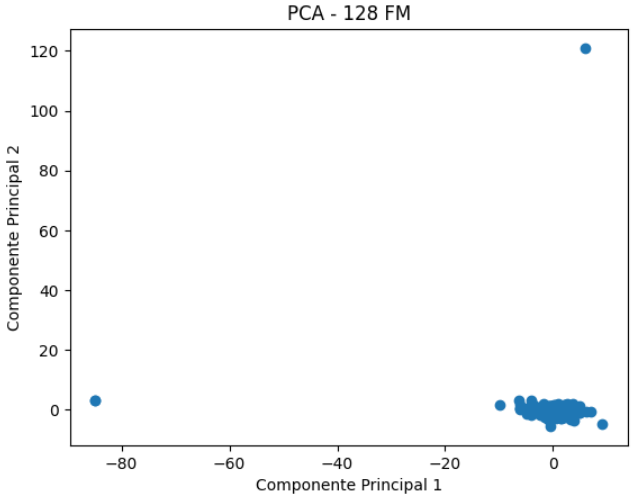

Figure 11: PCA para espacio latente de 128 FM

A pesar del análisis realizado, no pudimos encontrar patrones significativos en los tres espacios latentes. Creemos que esto probablemente se deba a la simplicidad del modelo, que aunque logra reconstruir las frecuencias de audio con precisión, no alcanza a aprender características comunes como género o tonalidad entre las canciones que sean identificables en el espacio latente. Este comportamiento nos sugiere que, aunque el modelo es efectivo en su tarea principal, no cuenta con la complejidad suficiente para aprender relaciones más profundas presentes en los datos musicales.

4 Encodear música nueva

Realizamos pruebas de codificación utilizando canciones que no formaban parte del dataset de entrenamiento, creando un nuevo conjunto de datos con una variedad intencional. Este nuevo dataset incluyó ejemplos como una canción de cumbia, una canción en formato 8D, una grabación en vivo, un audio de WhatsApp, entre otros.

Los resultados fueron en general positivos: la mayoría de las canciones se reconstruyeron con una calidad sorprendentemente buena, considerando que el modelo no había sido entrenado con este tipo de datos. Sin embargo, observamos un desempeño notablemente deficiente en el caso del audio de WhatsApp, que resultó extremadamente saturado e ininteligible, dificultando completamente la comprensión del contenido hablado.

En conclusión, el modelo demostró ser capaz de codificar música nueva con buenos resultados, pero encontró limitaciones significativas al procesar un audio de WhatsApp debido a la naturaleza y calidad de este tipo de grabaciones.

5 Generación de música

En esta sección, se evaluó la posibilidad de modificar vectores latentes generados por el autoencoder para crear nuevos audios, realizando varias pruebas.

Primero, se generaron audios desde vectores completamente aleatorios, obteniendo sonidos artificiales con patrones sugerentes, aunque mayormente ruido. Luego, se añadió ruido a vectores latentes reales, conservando algunas características de la canción original pero con distorsiones proporcionales al ruido agregado. También se modificaron componentes específicos de los vectores, logrando cambios específicos en el audio, aunque en algunos casos el resultado fue ruido o distorsiones como pitidos. Finalmente, se realizó una interpolación entre canciones, combinando vectores latentes de dos canciones distintas, y los resultados mostraron características intermedias entre ambas, con transiciones suaves en algunos casos.

En conclusión, este método permitió manipular vectores latentes para generar nuevos audios, con resultados variables según la modificación realizada. Si bien algunos experimentos resultaron en sonidos poco coherentes, otros, como la interpolación, destacaron por su capacidad para combinar características y explorar creativamente el espacio latente aprendido por el modelo.

6 Conclusión

En este trabajo práctico, logramos desarrollar un autoencoder eficiente para representar canciones en un espacio latente compacto, con un tamaño de 13824 (64x216). A través de diversos experimentos, determinamos que una arquitectura simple con una capa convolucional y una deconvolucional resultó en el mejor balance entre tamaño del espacio latente y calidad de las reconstrucciones.

El análisis del espacio latente mostró que, aunque el modelo es efectivo en la reconstrucción de audio, su simplicidad limita su capacidad para capturar relaciones más profundas entre las canciones en el espacio latente. Si bien el clustering identificó correctamente canciones repetidas, la mayoría de los datos se agruparon en un único cluster, sin patrones significativos. Por otro lado, PCA resaltó outliers asociados a características particulares, como alta velocidad, evidenciando que el espacio latente captura algunos atributos individuales pero carece de complejidad para representar otro tipo de información como los géneros. Estos resultados sugieren que un modelo más robusto podría explorar relaciones más profundas en las canciones.

En pruebas con canciones fuera del dataset de entrenamiento, si bien el autoencoder muestra una capacidad notable para encodear y reconstruir música no utilizada durante el entrenamiento, enfrenta limitaciones con entradas muy diferentes o ruidosas.

Por último, experimentos de manipulación del espacio latente demostraron la capacidad del modelo para generar nuevos sonidos. Aunque algunos resultados estaban distorsionados, la capacidad del modelo para conservar características originales con algunas modificaciones destaca el poder del espacio latente para explorar nuevas posibilidades.

En síntesis, este trabajo permitió no solo comprimir y reconstruir canciones con éxito, sino también explorar cómo las representaciones latentes pueden ser utilizadas para análisis, generalización y creación.