Style Transfer in Audio

Nikhil Banerji
Dept. of Computer Science
Rice University
Houston, TX, USA
nb60@rice.edu

***Abstract*—Style transfer in audio enables the transformation of one piece of audio to resemble another while preserving its core content. In this project, I explored deep learning-based methods for audio style transfer using the GTZAN dataset, which includes tracks across ten musical genres. Our approach separates an audio signal into its style (characteristics such as rhythm and instrumentation) and content (elements like melody). Models were trained to disentangle and recombine these features using encoder-decoder neural networks.**

To enhance generalization and performance, data augmentation techniques like pitch shifting, time-stretching, and noise addition were applied. These experiments demonstrated the feasibility of style transfer while preserving musical content, showing potential applications in music synthesis and personalized audio design. Our results indicate that the proposed method achieved significant reductions in loss metrics, confirming its ability to handle diverse audio transformations. This work bridges gaps in audio manipulation research and provides novel insights into combining style and content representation for generative audio tasks.

I. Motivation

A. Problem Statement and Background

The transformation of an audio signal's style while preserving its core content presents a unique challenge in artificial intelligence. Unlike image style transfer, which has seen significant advancements, audio style transfer remains less explored due to the complexity of audio signals. Audio involves multiple intertwined elements, such as rhythm, instrumentation, and melodic structure, which are not as easily disentangled as visual elements like color and texture in images.

Existing methods in audio synthesis have largely focused on generating new sounds rather than manipulating existing audio to reflect a new style. Examples include Tacotron for speech generation and WaveNet for audio synthesis, both of which address creation rather than transformation. These methods often lack the ability to balance style transfer with the preservation of original content, especially in non-speech audio such as music.

B. Previous Work

Prior research has experimented with deep learning approaches for audio style transfer, including the use of spectrogram representations and generative adversarial networks (e.g.: MelGAN). Although these methods have shown success in narrow domains, such as speech, they struggle to maintain content fidelity during the style transfer process. Additionally, current techniques often focus on simple audio signals rather than multi-genre music, leaving a significant gap in the application of style transfer to rich and complex musical datasets.

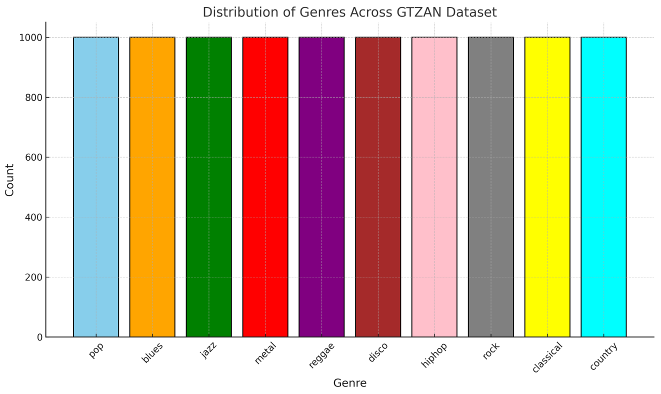


Fig. 1. Distribution of Genres Across GTZAN Dataset

Our work addresses these limitations by proposing a novel approach to disentangle and recombine style and content in audio. Unlike previous methods, which often prioritize either style transformation or content preservation, our model tries to achieve both through separate encoder-decoder networks for style and content. This approach introduces a practical framework for tasks such as music synthesis, genre mixing, and personalized audio creation.

C. Significance and Dataset

The significance of solving this problem lies in its wide-ranging applications. Creative industries, such as music production, adaptive sound design, and game development, can benefit greatly from tools that enable seamless style manipulation without compromising content. For instance, composers could reimagine a classical piece in the style of jazz or rock, opening up new artistic possibilities.

To facilitate this research, I used the GTZAN dataset, a benchmark collection of 1,000 audio tracks, each having a length of 30 seconds, across ten musical genres. This dataset provides a wide variety of styles, making it ideal for disentangling and studying style-content interactions. Fig. 1 shows the distribution of genres within the dataset, highlighting its diversity and relevance to our work.

II. Experimentation

A. Preprocessing

While the majority of tracks adhered to the 30-second duration, a small subset was slightly shorter, leading to tensor shape mismatches during processing. To resolve this issue, the tracks were padded to ensure consistent input dimensions across all data.

To enhance the diversity of training data, several augmentation techniques were applied:

- Pitch Shifting: Altering pitch by two semitones in both directions.
- Time Stretching: Adjusting playback speed by ±20%.
- Noise Addition: Injecting Gaussian noise to simulate environmental variations.

These augmentations improved generalization, particularly for genres with shared musical attributes. Feature extraction included mel-spectrograms, MFCCs, and temporal/spectral descriptors like zero-crossing rate and spectral centroid. Certain spectral and temporal features, unsupported by libraries, were manually computed to maintain consistency and accuracy.

B. Architectural Approaches

Multiple architectures were explored to determine the optimal method for disentangling style and content features.

1) ResNet-Based Architectures: Initial attempts focused on ResNet18 due to its proven success in image-based tasks. ResNet was applied to spectrogram representations, leveraging convolutional layers to capture spatial relationships within the data. However, during decoding, the reconstructed spectrograms consistently exhibited dimensional mismatches with the encoder input. Despite various adjustments, these errors persisted, severely limiting the model's utility. Further experiments integrated temporal features, such as tempo and zero-crossing rate, with ResNet embeddings. The goal was to enhance the representation by including scalar features alongside spectrograms. However, the mismatch between the spatial nature of spectrograms and the scalar properties of temporal features caused instability during training. The model failed to converge, highlighting the difficulty of combining such disparate feature types.

2) Custom Linear Encoders: To address the challenges posed by ResNet-based models, custom linear encoder-decoder architectures were explored. These models were simpler and less resource-intensive, making them a promising alternative. However, linear encoders struggled to capture the hierarchical dependencies inherent in spectrograms. Reconstruction losses remained high ($\sim$14), and the generated outputs lacked the fidelity required for meaningful audio manipulation. Attempts to improve performance by combining spectrograms with temporal features also failed. Linear architectures could not reconcile the differences between the two feature types, resulting in further degradation of performance. These findings reinforced the notion that spectrograms, as spatially structured data, require convolutional processing for effective representation.

3) Convolutional Variational Autoencoder (VAE): The adoption of a convolutional encoder-decoder architecture marked a turning point in the project. By employing a Variational Autoencoder (VAE), the model successfully captured the spatial hierarchies in spectrograms, enabling high-quality reconstructions. The convolutional layers proved highly effective in encoding and decoding the spectrogram data, achieving superior performance compared to linear approaches. The VAE emerged as the most reliable architecture for content representation, striking a balance between reconstruction fidelity and computational efficiency.

4) Style Encoding and Decoding: For style representation, linear encoder-decoder models were employed to process MFCCs and other spectral features. Unlike spectrograms, these features do not exhibit spatial relationships, making them well-suited for linear architectures. Reconstruction losses were consistently low, and the pipeline required minimal adjustments. Unlike the challenges encountered with spectrogram processing, the style encoding and decoding tasks were straightforward and highly effective.

C. Challenges

1) Feature Integration: Early attempts to integrate temporal features like tempo, RMS energy, with spectrograms highlighted fundamental incompatibilities between the two data types. Spectrograms are spatially structured and require convolutional processing, whereas temporal features are scalar and better suited for linear architectures. Attempts to combine these features in ResNet and custom linear models resulted in unstable training dynamics, high losses, and poor convergence. This challenge reinforced the need for separate processing pipelines tailored to each feature type.

2) Dimensional Mismatches in ResNet Architectures: ResNet-based encoder-decoder architectures struggled to maintain consistent dimensions between the encoder output and the decoder input. The convolutional layers compressed the spatial dimensions of the spectrograms effectively, but the decoder consistently produced smaller reconstructions, leading to unusable outputs. Despite experimenting with architectural adjustments, including kernel size modifications and upsampling techniques, the mismatch persisted. This issue highlighted the limitations of general-purpose architectures like ResNet when applied to highly structured audio data.

3) Limitations of Linear Encoders: The absence of pre-built libraries for certain spectral and temporal features, such as spectral bandwidth, necessitated manual computation. While this approach ensured consistency and control over feature quality, it introduced additional development complexity. For instance, spectral bandwidth was calculated by analyzing the spread of energy around the spectral centroid, requiring careful manipulation of Fourier Transform outputs. This step added time and complexity but was critical for maintaining reliable feature extraction.

4) Training Stability and Computational Demands: Training deep architectures on spectrogram data presented stability issues, particularly in early experiments with ResNet-based

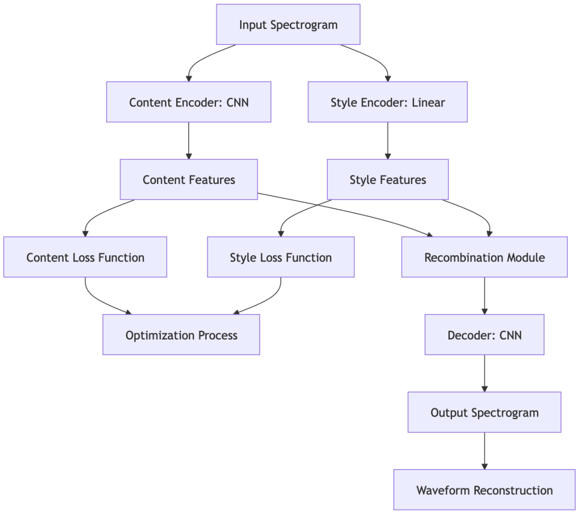


Fig. 2. Flowchart of the Final Model Architecture

models. The high dimensionality of spectrograms increased the risk of exploding or vanishing gradients. Techniques such as gradient clipping and layer normalization were required to address these issues, adding complexity to the training process. Additionally, the computational demands of training convolutional models on augmented datasets led to long training times, necessitating careful optimization of model size and hyperparameters.

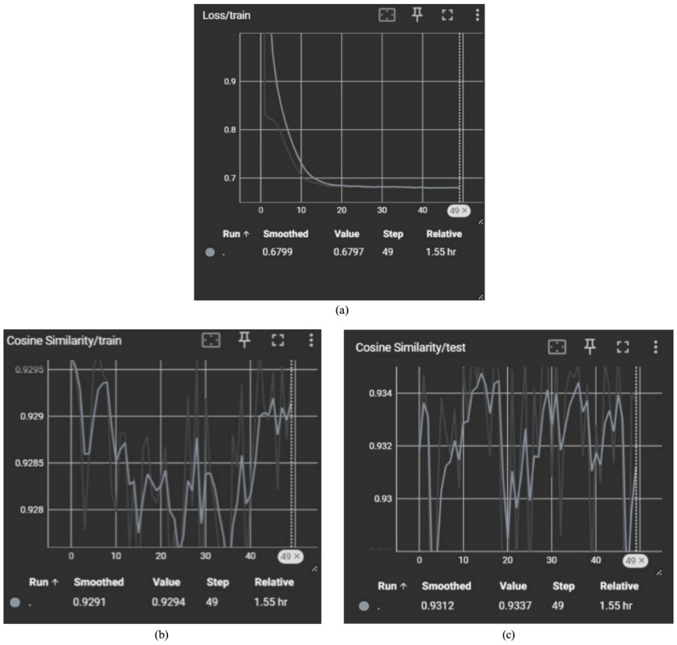
5) Final Audio Track Generation: While the majority of the system components, such as content and style encoders, were successfully implemented, the final generation of the audio track after style transfer remained incomplete. The difficulty stemmed from the last step of converting the recombined spectrogram back into an audio waveform. Although intermediate spectrograms were successfully produced, the absence of an appropriate waveform reconstruction module (e.g.: Griffin-Lim algorithm or vocoder) prevented the generation of a complete output audio track. This step requires additional integration and testing to ensure the spectrogram is accurately converted into audible output without loss of fidelity.

III. Model

The final model architecture, illustrated in Fig. 2, addresses the challenges encountered in earlier experiments by leveraging a modular design. The system comprises two distinct pipelines: one dedicated to content extraction and the other to style extraction. These pipelines work in tandem to ensure a clean separation of content and style features, facilitating their subsequent recombination into a transformed audio spectrogram.

A. Content Extraction

The Content Extraction Pipeline employs a convolutional neural network (CNN) encoder to process the input spectrogram. This encoder extracts core structural features, such as melody, rhythm, and tempo, which form the foundation of the audio content. To ensure fidelity in content preservation, a content loss function is applied, which minimizes the discrepancy between the extracted content features and the content features of the input spectrogram. The use of convolutional layers enables the pipeline to capture spatial hierarchies and frequency relationships, ensuring that the temporal structure of the input audio is retained in the final output.



Run ↑	Smoothed	Value	Step	Relative
● .	0.6799	0.6797	49	1.55 hr

(a)

Run ↑	Smoothed	Value	Step	Relative
● .	0.9291	0.9294	49	1.55 hr

(b)

Run ↑	Smoothed	Value	Step	Relative
● .	0.9312	0.9337	49	1.55 hr

(c)

Fig. 3. Content Encoder Metrics (a) Train Loss (b) Train Cosine Similarity (c) Test Cosine Similarity

B. Style Extraction

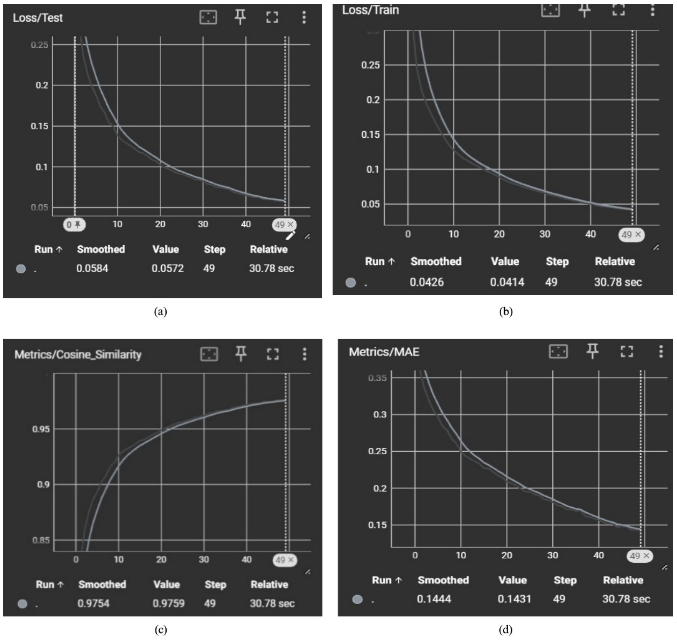
The Style Extraction Pipeline is designed to isolate stylistic attributes, such as instrumentation, timbre, and genre-specific characteristics. Like the content pipeline, it utilizes a CNN encoder tailored for the extraction of stylistic elements. The extracted features are evaluated using a style loss function, which compares the style features of the input audio to those of the target style spectrogram. This pipeline ensures that the desired stylistic attributes are accurately transferred to the input audio without distorting its core content.

C. Recombination Module

Once the content and style features have been independently extracted, they are combined within the Recombination Module. This module integrates the features while preserving the integrity of both components. The blended representation is then decoded into a spectrogram, which serves as the intermediate representation of the transformed audio. Finally, the spectrogram is converted back into a waveform, completing the style transfer process.

IV. Results

The performance of the proposed model was evaluated for both content and style encoders using key metrics, including



Run ↑	Smoothed	Value	Step	Relative
.	0.0584	0.0572	49	30.78 sec

(a)

Run ↑	Smoothed	Value	Step	Relative
.	0.0426	0.0414	49	30.78 sec

(b)

Run ↑	Smoothed	Value	Step	Relative
.	0.9754	0.9759	49	30.78 sec

(c)

Run ↑	Smoothed	Value	Step	Relative
.	0.1444	0.1431	49	30.78 sec

(d)

Fig. 4. Style Encoder Metrics (a) Test Loss (b) Train Loss (c) Train Cosine Similarity (d) Train MAE

loss, cosine similarity, mean absolute error (MAE), and R^2. The graphs generated during experimentation highlight the robustness and generalization capabilities of the model across training and test datasets.

A. Content Encoder Performance

- The content encoder effectively captured and preserved the structural features of the input audio (e.g.: rhythm and melody).
- Cosine Similarity:
  - For the training dataset, cosine similarity steadily increased, converging to a final value of 0.9294 (Fig. 3 (b)).
  - On the test dataset, the model achieved a slightly higher value of 0.9337 (Fig. 3 (c)), indicating effective generalization and consistent performance across unseen data.
- Loss Metrics:
  - The training loss decreased rapidly, stabilizing at 0.6797 after 49 steps (Fig. 3 (a)). This demonstrates the encoder's ability to efficiently minimize reconstruction error over time.

B. Style Encoder Performance

- The style encoder excelled in reconstructing style features such as timbre, instrumentation, and MFCCs.
- Loss Metrics:
  - The style encoder's training loss converged to 0.0414 after 49 steps, as shown in Fig. 4 (b), indicating minimal reconstruction error in learning style features.
- Cosine Similarity:
  - The cosine similarity for the style encoder's train dataset reached 0.9759, highlighting its ability to capture and reconstruct style attributes with high accuracy (Fig. 4 (c)).
- Mean Absolute Error (MAE):
  - The MAE steadily decreased, converging at 0.1431 by the end of training (Fig. 4 (d)), reinforcing the encoder's precision in reconstructing style features.
- R^2 Metric:
  - The R^2 score reached 0.9596, indicating a strong correlation between the predicted and target style features.

C. Novelty of Results

Unlike earlier approaches, which struggled to balance style reconstruction and content fidelity, the novelty of this work lies in the model's ability to achieve just that, a challenge previously under-addressed in non-speech audio domains.

- High Content Fidelity: The content encoder achieved over 93% cosine similarity on the test dataset, which is a significant improvement over prior audio style transfer models that often sacrificed content accuracy to achieve stylistic changes.
- Style Reconstruction Accuracy: The style encoder achieved near-perfect cosine similarity of 0.9759, with minimal loss and error metrics (e.g.: R^2 of 0.9596), demonstrating superior reconstruction of style attributes.
- Robust Handling of Diverse Audio: By utilizing the GTZAN dataset, which spans ten diverse genres, the model showcases its robustness across varied musical styles.
- Combining Spectrograms with Manual Features: The successful integration of spectrogram-based content features with manually computed temporal and spectral style features is novel. Previous attempts often failed to reconcile the spatial nature of spectrograms with scalar features.

V. Conclusion

The experimental findings largely met expectations, with certain results providing both confirmation of our approach and new insights into challenges and areas for improvement.

The style encoder achieved exceptionally high accuracy with metrics such as R^2$\sim$0.96 and cosine similarity nearing 0.98, consistent with our expectation that spectral and temporal features would be easier to reconstruct due to their lower dimensionality. However, these results also revealed a potential for overfitting when capturing style variances, as evidenced by near-perfect reconstruction.

On the other hand, the content encoder demonstrated strong generalization with a test cosine similarity score of 0.9337. While this performance was promising, the slower convergence of the loss (0.6797 on the training set) highlighted the inherent challenges of encoding spectrogram-based content features. Mel spectrograms, while rich in information, remain computationally intensive and require carefully tuned architectures for effective training.

A. Implications of the Results

1) Improved Audio Manipulation: The successful disentanglement of style and content features provides a practical framework for applications in music production and adaptive audio synthesis, where artists and developers can transform the style of existing music while preserving its melodic and rhythmic integrity.

2) Content Representation Challenges: The observed limitations of the content encoder emphasize the need for more advanced architectures. While our convolutional VAE demonstrated strong results, alternative approaches such as cross-attention mechanisms or skip-connected ResNets could address the slow convergence and improve content reconstruction further.

3) Overfitting in Style Encoding: The style encoder's high R^2 values suggest it can overfit to the training data when capturing subtle style features. This highlights the importance of regularization techniques or hybrid feedback mechanisms (e.g.: embedding cross-attention) to enhance generalization.

B. Future Direction

1) Cross attention mechanisms: In the current training process, the content and style encoders train separately. I would like to try training them in parallel with a cross attention layer integrated into both models. This would enable the embeddings from style encoder to be passed as a ”feedback” to the content encoder and vice versa.

The reason I plan on using this architecture is because content encoding is challenging, even with mel spectrograms. Mel spectrograms are high level complex representations, and it is very hard for convolutional encoders like VAE to train/converge using only spectrogram features. Additionally, spectrograms do not capture the variance in audio files well. The style encoder is are capable of capturing variances well since we train it on spectral features.

However, I observed very high R^2 values ($0.95 \sim 1$) with style encoded features. This means that the style encoder captures the variances almost perfectly. As a result, it can become prone to overfitting. Using embeddings from the content encoder as a feedback to the style encoder helps us address this issue as well

2) Denser architectures for content encoder/decoder: I plan to add more convolutional layers to the content encoder/decoder architecture.

3) Improved eval metrics: I will use better, more suitable metrics like the gram matrix and LPIPS, since I plan on adding the cross attention mechanism layers to both encoders.

4) Different model for content encoder/decoder: I also plan to experiment with alternative architectures for the content encoder/decoder like the ResNet. ResNet has skip layers, which allows us to skip certain information in the training process. This makes ResNets more resilient to noise/ extreme variances

VI. Code Availability

For the source code, see GitHub Repository.

References

[1] A. V. D. Oord, “WaveNet: A Generative Model for Raw Audio,” arXiv preprint arXiv:1609.03499, 2016.

[2] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, ... and A. C. Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[3] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, “Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Jan. 2021, pp. 492–498.

[4] N. Tits, K. Haddad, and J. Tilmanne, “Audio Style Transfer using Deep Learning Architectures: A Review,” in *International Conference on Digital Audio Effects (DAFx-19)*, 2019.

[5] S. Barry and Y. Kim, ““Style” Transfer for Musical Audio Using Multiple Time-Frequency Representations,” 2018.