Audio-Visual Generation from Text input

A Project Report Submitted to the
Department of Computer Science of
Ramakrishna Mission Vivekananda Educational and Research Institute, Belur,
in partial fulfilment of the requirements for the degree of
MSc in Computer Science.

Submitted by
Biswajit Rana
ID No. B2330026

Supervisor:
Champak Dutta
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute

Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, Howrah 711202, West Bengal, India
December 24, 202

Audio-Visual Generation from Text input

By

Biswajit Rana

Declaration by student:

“I hereby declare that the present dissertation is the outcome of my project work under the guidance of Champak Dutta and I have properly acknowledged the sources of materials used in my project report."

(Biswajit Rana, ID No. B2330026)

A project report in the partial fulfilment of the requirements of the degree of MSc in Computer Science

Examined and approved on

by

Champak Dutta (supervisor)
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute

Countersigned by

Registrar
Ramakrishna Mission Vivekananda Educational and Research Institute

Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, Howrah 711202, West Bengal, India
December 24, 2024

Acknowledgement

The present project work is submitted in partial fulfilment of the requirements for the degree of Master of Science of Ramakrishna Mission Vivekananda University (RKMVU). I express my deepest gratitude to my supervisor Champak Dutta of Ramakrishna Mission Vivekananda Educational and Research Institute for his inestimable support, encouragement, profound knowledge, largely helpful conversations and also for providing me a systematic way for the completion of my project work. His ability to work hard inspired me a lot. I am also extremely grateful to the Vice-Chancellor of this University for his encouragement and support throughout the course. Last but not the least, this work would not have been possible without support of my fellow classmates.

Belur
December 24, 2024

(Biswajit Rana)
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The demand for creative media assets, such as music and poster layouts, is ever-growing in industries like advertising, social media, and entertainment. However, the process of acquiring these assets is riddled with challenges. Many creators and businesses struggle with high costs, licensing restrictions, or the necessity of giving credit for purchased music and poster ideas. These barriers can impede innovation and limit accessibility for small-scale creators and startups.

Music generation has traditionally required expertise in composition, access to professional recording equipment, or reliance on pre-existing soundtracks. Models like *MusicGen* alleviate these challenges by enabling the creation of high-quality, customizable music from simple textual inputs. This democratization of music production caters to creators who need bespoke soundtracks without the logistical hurdles of traditional methods.

Similarly, layout design—essential for marketing, branding, and content presentation—often involves substantial time and cost investments. *PosterLlama* addresses these challenges by automating the generation of content-aware layouts using advanced language and vision models. This not only streamlines the creative process but also ensures output quality comparable to human designers.

The integration of these models into a single application offers transformative potential. Such an application provides users with a unified platform for generating both music and layout assets. This innovation is especially beneficial for small businesses, individual creators, and educators who need high-quality outputs without the associated costs.

By combining *MusicGen* and *PosterLlama*, the proposed application fosters creativity, accessibility, and efficiency. It empowers users to produce customized media assets, catering to their specific requirements, while bypassing licensing and crediting issues. This solution is poised to redefine the landscape of creative media production.

The integration of MusicGen and PosterLlama into a single application represents a transformative step forward in creative media production. By offering a unified platform for generating both music and layout assets, this solution addresses the diverse needs of creators across industries. Such an application is particularly beneficial for small businesses, independent creators, educators, and marketers who require cost-effective, high-quality outputs. It eliminates logistical hurdles while empowering users to customize their creative assets with ease and precision.

This integrated approach fosters creativity, accessibility, and efficiency, revolutionizing the way media assets are produced. By bypassing licensing restrictions and crediting requirements, the application removes long-standing barriers in the creative process. The combined power of MusicGen and PosterLlama is poised to redefine the landscape of creative media production, empowering users to bring their ideas to life in unprecedented ways.

Chapter 2

Literature Survey

2.1 PosterLlama

" *PosterLlama: Bridging Design Ability of Language Model to Contents-Aware Layout Generation* "

PosterLlama is a groundbreaking model that reformats layout elements into HTML sequences, effectively leveraging the capabilities of language models to generate content-aware layouts. It integrates both visual and textual encoding to ensure high-quality outputs that align with user-provided prompts. The model employs a two-stage training approach, focusing first on alignment and then fine-tuning to enhance performance. Additionally, it incorporates depth-guided augmentation strategies, making it robust to various input conditions.

The technical contributions of PosterLlama are noteworthy. It introduces a unique depth-based poster augmentation strategy, which enhances the robustness of layout generation by simulating varied depth conditions. The use of DINOv2 as a visual encoder ensures seamless text-visual alignment, enabling the model to produce layouts that are both visually appealing and semantically rich. Representing layouts in HTML further adds to its versatility, ensuring semantic richness and compatibility across platforms.

In terms of performance, PosterLlama consistently outperforms competitors such as RADM and LayoutPrompter across established benchmarks. Its ability to generate superior layout quality and maintain strong content-awareness makes it a valuable tool in the field of automated design generation.

Key Features:

- Integration of visual and textual encoding.
- Use of a two-stage training approach for alignment and fine-tuning.
- Depth-guided augmentation for enhanced robustness.

Technical Contributions:

- Introduces a unique depth-based poster augmentation strategy.
- Employs DINOv2 as a visual encoder to bridge text-visual alignment.
- Leverages HTML-based layout representation to ensure semantic richness.

Performance: PosterLlama outperforms competitors like RADM and Layout-Prompter across benchmarks, demonstrating superior layout quality and content-awareness.

2.2 MusicGen

Simple and Controllable Music Generation

Framework Description: MusicGen employs a single-stage transformer-based architecture that eliminates the need for multi-stage dependencies commonly found in traditional music generation models. This streamlined architecture allows for more efficient audio tokenization, achieved through residual vector quantization (RVQ). MusicGen also introduces innovative codebook interleaving patterns, striking a balance between computational cost and output fidelity.

The model supports both text and melody-based conditioning, enabling users to input textual prompts or melodies for generating high-quality stereo audio. Notably, it achieves this without additional computational overhead, making it highly efficient. Extensive evaluations against baselines like Riffusion and Mousai highlight its robustness and superiority in terms of fidelity and alignment metrics.

MusicGen is particularly strong in handling diverse textual and melodic inputs. Its comprehensive evaluations showcase its ability to produce high-fidelity audio that aligns closely with user prompts, setting a new benchmark for transformer-based music generation models.

Key Features:

- Supports both text and melody-based conditioning.

- Produces stereo audio without additional computational overhead.
- Conducts extensive evaluations, surpassing baselines like Riffusion and Mousai.

Strengths:

- Robust to varying textual and melodic inputs.
- Comprehensive evaluations highlight its superiority in fidelity and alignment metrics.

2.3 Natural TTS Synthesis

Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions

Natural TTS Synthesis leverages the WaveNet model to synthesize natural speech from mel spectrograms. This approach emphasizes smooth transitions and high-fidelity outputs, making it a cornerstone of advancements in text-to-speech synthesis. WaveNet's ability to generate highly realistic audio inspires control mechanisms for fine-tuned audio generation in MusicGen.

The conditioning strategies demonstrated in Natural TTS Synthesis serve as a foundation for improving the generation quality of models like MusicGen. These strategies highlight the potential of leveraging fine-grained control to enhance the alignment and fidelity of audio outputs, particularly in scenarios requiring high degrees of customization.

Core Ideas:

- Utilizes WaveNet for synthesizing natural speech from mel spectrograms.
- Emphasizes smooth transitions and high-fidelity outputs.

Applications to Proposed Models:

- Inspires control mechanisms for fine-tuned audio generation in MusicGen.
- Demonstrates the potential of conditioning strategies for improving generation quality.

These three papers collectively underscore the advancements in generative models for audio and visual tasks, forming the backbone of the proposed application.

Chapter 3

Proposed Algorithm

3.1 Methods in MusicGen model

3.1.1 Audio Tokenization

MusicGen employs EnCodec, a convolutional autoencoder, to transform audio into discrete tokens. Residual Vector Quantization (RVQ) compresses audio signals into parallel streams of discrete tokens, preserving fidelity while reducing data complexity. Each frame is quantized across multiple codebooks, with the first codebook encoding the most significant features. Subsequent codebooks encode finer details, allowing for nuanced audio generation.

3.1.2 Codebook Interleaving

MusicGen introduces novel interleaving patterns for managing parallel token streams. These include *"delay"* and *"parallel"* patterns, which balance the complexity of token dependencies.

- The delay pattern offsets tokens from different streams, reducing overlap and enhancing model efficiency.

- Parallel patterns allow simultaneous token processing, significantly speeding up generation without compromising quality.

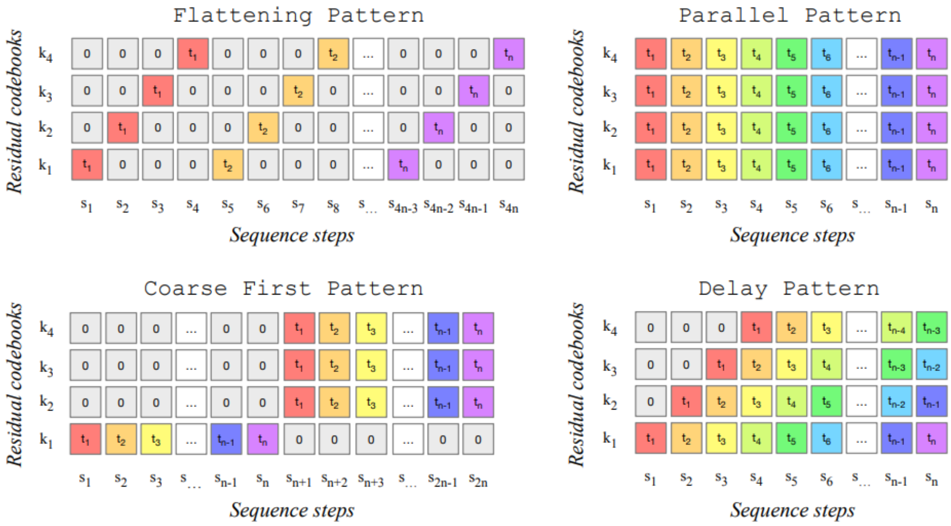


Figure 3.1: Codebook interleaving patterns

3.1.3 Codebook Interleaving Patterns (See Figure 3.1)

Exact Flattened Autoregressive Decomposition

An autoregressive model requires a discrete random sequence $U \in \{1, \dots, M\}^S$ with S as the sequence length. By convention, we define $U_0 = 0$, a deterministic special token indicating the beginning of the sequence. We model the distribution as:

$$\forall t > 0, \quad p_t(U_{t-1}, \dots, U_0) P[U_t \mid U_{t-1}, \dots, U_0]. \tag{1}$$

Let us build a second sequence of random variables $\tilde{U}$ using the autoregressive density p, recursively defining $\tilde{U}_0 = 0$ and:

$$\forall t > 0, \quad P\left[\tilde{U}_t \mid \tilde{U}_{t-1}, \dots, \tilde{U}_0\right] = p_t\left(\tilde{U}_{t-1}, \dots, \tilde{U}_0\right). \tag{2}$$

The sequences U and $\tilde{U}$ follow the same distribution. If we perfectly estimate $\hat{p}$ of p using a deep learning model, we can fit the distribution of U exactly.

The main issue with the representation Q obtained from the EnCodec model is having K codebooks for each time step. A solution is flattening Q, resulting in $S = d \cdot fr \cdot K$, where we first predict the first codebook of the first time step, then the second codebook, and so on. Using equations (1) and (2), we can theoretically model the distribution of Q exactly. However, this increases complexity, reducing some of the gains from the lowest sample rate fr.

Multiple flattening approaches exist. For example, MusicLM [**?**] uses two models: one for the first $K/2$ codebooks and another for the remaining $K/2$, conditioned on the first model. The number of autoregressive steps remains $d \cdot fr \cdot K$.

Inexact Autoregressive Decomposition

An alternative is using an inexact autoregressive decomposition where some codebooks are predicted in parallel. Define a sequence $V_0 = 0$, and for $t \in \{1, \dots, T\}, k \in \{1, \dots, K\}$, let $V_{t,k} = Q_{t,k}$. Dropping the codebook index k, V_t represents the concatenation of all codebooks at time t.

$$p_{t,k}(V_{t-1}, \dots, V_0) P[V_{t,k} \mid V_{t-1}, \dots, V_0]. \tag{3}$$

Recursively, define $\tilde{V}_0 = 0$ and:

$$\forall t > 0, \forall k, \quad P\left[\tilde{V}_{t,k}\right] = p_{t,k}\left(\tilde{V}_{t-1}, \dots, \tilde{V}_0\right). \tag{4}$$

Unlike equation (2), $\tilde{V}$ may not follow the same distribution as V, even with an exact distribution $p_{t,k}$. Errors can accumulate as t increases, causing divergence. This approach, while inexact, maintains the original frame rate, significantly speeding up training and inference for long sequences.

Arbitrary Codebook Interleaving Patterns

To experiment with different decompositions, we introduce codebook interleaving patterns. Let $\Omega = \{(t,k) : t \in \{1, \dots, d \cdot fr\}, k \in \{1, \dots, K\}\}$ be the set of all time steps and codebook indices. A codebook pattern is a sequence $P = (P_0, P_1, P_2, \dots, P_S)$, with $P_0 = \emptyset$, where each $P_s \subset \Omega$ forms a partition of Ω. We predict all positions in P_s conditioned on $P_0, P_1, \dots, P_{s-1}$.

For practical use, we restrict patterns so each codebook index appears at most once in any P_s. Examples include:

$$P_s = \{(s,k) : k \in \{1, \dots, K\}\} \tag{5}$$

for a parallel pattern and:

$$P_s = \{(s-k+1, k) : k \in \{1, \ldots, K\}, s-k \geq 0\} \tag{6}$$

for a delay pattern [**?**].

Through empirical evaluations, we compare the benefits and drawbacks of different codebook patterns, highlighting the importance of exact modeling for parallel sequences.

3.1.4 Transformer Architecture (See Figure 3.2)

The autoregressive transformer decoder forms the core of MusicGen. It predicts tokens sequentially, conditioned on textual or melodic inputs. Positional embeddings and cross-attention layers integrate context from inputs, ensuring coherence and relevance in generated outputs. Transformer layers are equipped with residual connections and normalization to stabilize training and improve convergence.

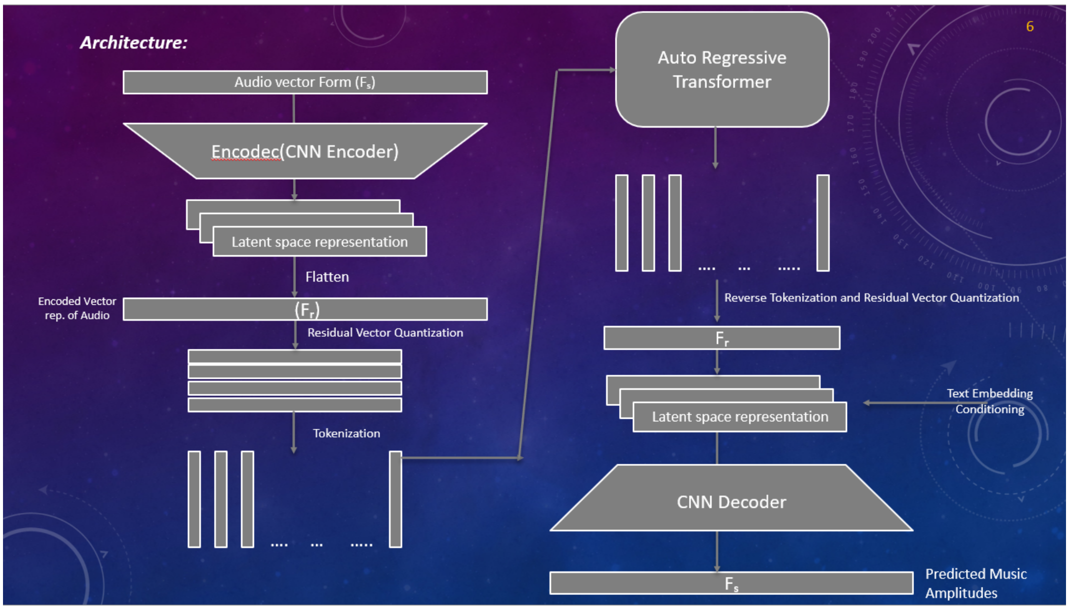

Architecture:
Audio vector Form (F_s)
Encodec(CNN Encoder)
Latent space representation
Flatten
Encoded Vector rep. of Audio
(F_r)
Residual Vector Quantization
Tokenization
....
Auto Regressive Transformer
....
Reverse Tokenization and Residual Vector Quantization
F_r
Latent space representation
Text Embedding Conditioning
CNN Decoder
F_s
Predicted Music Amplitudes
6

Figure 3.2: Transformer Architecture in the MusicGen model.

3.1.5 Conditioning Mechanisms

- **Text Conditioning:** Achieved using T5-based encoders, embedding textual descriptions into latent spaces.
- **Melody Conditioning:** Introduces chromagrams, capturing harmonic structures from audio inputs. This allows for iterative refinement and melody alignment.

3.1.6 DataSet

- **Original dataset :**20K hours of licensed music to train MUSICGEN. Specifically an internal dataset of 10K high-quality music tracks, and on the ShutterStock and Pond5 music data collections2 with respectively 25K and 365K instrument-only music tracks. All datasets consist of full-length music sampled at 32 kHz with metadata composed of a textual description and information such as the genre, BPM, and tags. We downmix the audio to mono unless stated otherwise. But this dataset is not available publically.
- **Used Dataset :** Used a small subset of Maestro-v3.0.0-midi dataset which is originally of size 120GB.

3.2 Methods in PosterLlama model

3.2.1 HTML-Based Representation (Figure: 3.3(left))

PosterLlama employs an HTML-like sequence representation for layout generation. Each layout element, such as rectangles, text, and other graphical components, is reformatted into structured sequences with tags like `<rect>` and `<svg>`. These tags capture essential semantic relationships and spatial constraints, providing a comprehensive framework for defining positions, sizes, and categories of layout elements. This representation ensures the layout's semantic integrity while supporting precise spatial arrangement, enabling efficient and consistent layout synthesis.

3.2.2 Two-Stage Training Process

PosterLlama adopts a two-stage training process to balance feature alignment and generation adaptability:

Stage 1: Adapter Module Training

- The first stage focuses on aligning visual features with textual representations using an adapter module.
- The adapter is optimized via cross-entropy loss, which ensures that visual features extracted from design elements are effectively encoded into textual representations.
- This alignment bridges the gap between visual layouts and textual encoding, creating a robust foundation for layout understanding and generation.

Stage 2: Fine-Tuning for Layout Generation

- In the second stage, the model is fine-tuned for generating layouts using HTML-formatted datasets.
- LoRA (Low-Rank Adaptation) is utilized to maintain knowledge acquired during the first stage while fine-tuning the model for specific layout generation tasks.
- This approach enhances the model's ability to synthesize complex, content-aware layouts while preserving the foundational visual-textual alignment.

3.2.3 Depth-Guided Augmentation

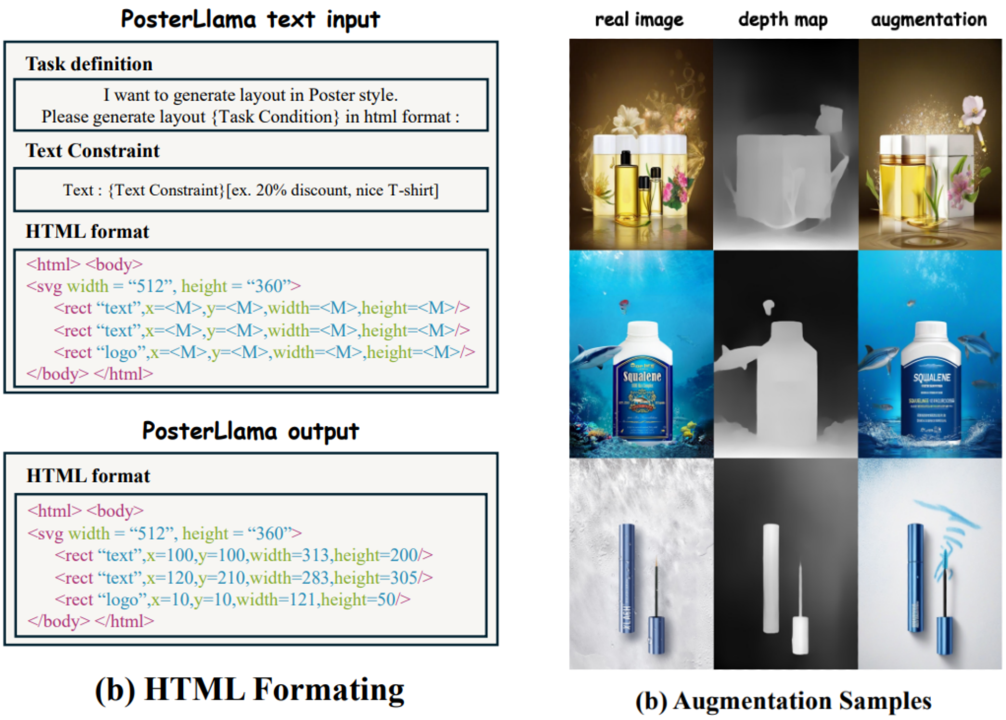
PosterLlama incorporates depth-guided augmentation to mitigate data scarcity issues and enhance generalization across diverse layout scenarios:

ControlNet Integration

Depth maps are generated using ControlNet to highlight salient regions within the layout. These maps guide the model in identifying and focusing on critical areas during layout synthesis.

Augmented Datasets (Figure: 3.3(right))

By augmenting the training datasets with depth-based variations, the model learns to handle a wider range of layout complexities and configurations.



PosterLlama text input

Task definition

I want to generate layout in Poster style.
Please generate layout {Task Condition} in html format :

Text Constraint

Text : {Text Constraint}[ex. 20% discount, nice T-shirt]

HTML format

```
<html> <body>
<svg width = “512”, height = “360”>
    <rect “text”,x=<M>,y=<M>,width=<M>,height=<M>/>
    <rect “text”,x=<M>,y=<M>,width=<M>,height=<M>/>
    <rect “logo”,x=<M>,y=<M>,width=<M>,height=<M>/>
</body> </html>
```

PosterLlama output

HTML format

```
<html> <body>
<svg width = “512”, height = “360”>
    <rect “text”,x=100,y=100,width=313,height=200/>
    <rect “text”,x=120,y=210,width=283,height=305/>
    <rect “logo”,x=10,y=10,width=121,height=50/>
</body> </html>
```

(b) HTML Formating

(b) Augmentation Samples

Figure 3.3: left: HTML formatting , right: Augmentation samples .

Improved Generalization

The depth-guided augmentation enables the model to produce visually coherent layouts that adapt seamlessly to diverse design requirements and constraints. This augmentation strategy ensures that PosterLlama excels in generating high-quality layouts, even in scenarios with limited or imbalanced training data.

3.3 Integration into the Application

The application seamlessly integrates MusicGen and PosterLlama algorithms, leveraging the strengths of shared transformer architectures. By utilizing a unified transformer backbone, the system ensures interoperability between the two models, enabling efficient cross-modal interactions.

3.3.1 Cross-Modal Conditioning

The integration employs cross-modal conditioning mechanisms, where the output of one modality (e.g., music) influences the other (e.g., layout). This ensures that the generated music and layout harmonize seamlessly, creating outputs that complement each other both aesthetically and contextually.

3.3.2 Unified Interface

A streamlined, user-friendly interface consolidates all functionalities into a single workflow. This design allows users to interact with both MusicGen and PosterLlama seamlessly, facilitating efficient asset generation without needing to navigate between separate tools or processes.

3.3.3 Enhanced User Experience

By combining these models into one application, users can create cohesive assets, such as music and visual layouts, with minimal effort, ensuring an intuitive and productive experience.

Chapter 4

Results

4.1 Results of MusicGen

Some Evaluation Metrics for Text-to-Music Evaluation.

- **FAD (Fréchet Audio Distance, FAD_{vgg}) $\downarrow$**
  - **Definition:** Measures the similarity between the generated music and ground-truth music by comparing statistical distributions of feature embeddings.
  - **Formulation:**

    $$FAD = \|\mu_g - \mu_r\|^2 + \mathrm{Tr}(\Sigma_g + \Sigma_r - 2(\Sigma_g \Sigma_r)^{1/2})$$

    where μ_g, Σ_g are the mean and covariance of the generated music features, and μ_r, Σ_r are those of the real music features. Lower values indicate better alignment.

- **KL (Kullback-Leibler Divergence, KL) $\downarrow$**
  - **Definition:** Quantifies how much the probability distribution of generated music $p_g(x)$ diverges from the ground-truth distribution $p_r(x)$.
  - **Formulation:**

    $$KL(p_r \| p_g) = \sum_x p_r(x) \log \frac{p_r(x)}{p_g(x)}$$

    where $p_r(x)$ and $p_g(x)$ are the ground-truth and generated distributions, respectively. A lower KL divergence indicates better similarity.

- **CLAP Score (Contrastive Language-Audio Pretraining Score, $CLAP_{scr}$) ↑**
  - **Definition:** Measures how well the generated music aligns semantically with the given text prompt. It uses a pretrained CLAP model to compute the similarity score.
  - **Formulation:**

    $$CLAP_{scr} = \cos(E_{\text{text}}, E_{\text{audio}})$$

    where E_{text} and E_{audio} are the text and audio embeddings, respectively. Higher cosine similarity indicates better alignment.

- **OVL (Overlap Score, OVL) ↑**
  - **Definition:** Quantifies the temporal overlap between the generated and real music by measuring alignment in rhythmic or temporal patterns.
  - **Formulation:**

    $$OVL = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}[f_g(t) = f_r(t)]$$

    where $f_g(t)$ and $f_r(t)$ are the features of generated and real music at time t, and $\mathbb{I}[\cdot]$ is an indicator function. Higher values indicate better temporal consistency.

- **REL (Relevance Score, REL) ↑**
  - **Definition:** Captures how well the generated music adheres to the thematic or stylistic intent specified in the text prompt. This is often computed as a subjective or aggregated score.
  - **Formulation:** No explicit mathematical formulation exists, but it typically involves aggregating expert or automated scoring for thematic alignment. Higher scores indicate better relevance to the prompt.

4.2 Comparison with other models

Table 4.1: Performance Comparison on the MUSICCAPS Test Set

MODEL	**FADvgg** ↓	**KL** ↓	**CLAPscr** ↑	**OVL.** ↑	**REL.** ↑
Riffusion	14.8	2.06	0.19	79.31 ± 1.37	74.20 ± 2.17
Mousai	7.5	1.59	0.23	76.11 ± 1.56	77.35 ± 1.72
MusicLM	4.0	-	-	80.51 ± 1.07	82.35 ± 1.36
Noise2Music	2.1	-	-	-	-
MUSICGEN w. Maestro	3.9	1.40	0.45	77.40	85.22

4.3 Results of PosterLlama

Some Evaluation matric of Text-to-Layout generation.

- **val (Validation Accuracy)** ↑: Measures the proportion of correctly predicted samples in the validation set.

$$\text{val} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

- **ove (Overlap Error)** ↓: Evaluates the extent of overlap mismatch between predicted and ground truth regions.

$$\text{ove} = 1 - \frac{\text{Intersection Area}}{\text{Union Area}}$$

- **ali (Alignment Error)** ↓: Assesses the misalignment of key points or edges in predictions.

$$\text{ali} = \frac{1}{N}\sum_{i=1}^{N} \|\text{Pred}_i - \text{GT}_i\|$$

- **und_l (Large Undetected Regions)** ↑: Proportion of large ground truth regions correctly detected.

$$\text{und\_l} = \frac{\text{Correctly Detected Large Regions}}{\text{Total Large Regions}}$$

- **und_s (Small Undetected Regions)** ↑: Proportion of small ground truth regions correctly detected.

$$\text{und\_s} = \frac{\text{Correctly Detected Small Regions}}{\text{Total Small Regions}}$$

- **rea (Reassignment Error)** ↓: Quantifies incorrect reassignment of regions or labels.

$$\text{rea} = \frac{\text{Misassigned Regions}}{\text{Total Regions}}$$

- **occ (Occlusion Error)** ↓: Evaluates the error due to occluded or overlapping regions.

$$\text{occ} = \frac{\text{Error in Occluded Regions}}{\text{Total Occluded Regions}}$$

4.4 Comparison with other models

Table 4.2: Model Performance Comparison

Model	**val** ↑	**ove** ↓	**ali** ↓	**und_l** ↑	**und_s** ↑	**rea** ↓	**occ** ↓
DS-GAN	0.8451	0.0336	0.0039	0.8848	0.5969	0.1169	0.0597
RADM	1.0000	0.0079	0.0026	0.9029	0.6817	0.0973	0.0528
RALF	1.0000	0.0156	0.0044	0.9820	0.9666	0.1126	0.0595
PosterLlama	1.0000	0.0006	0.0001	1.0000	1.0000	0.1142	0.0513

Chapter 5

Conclusion & Future Work

MusicGen, a controllable music generation model conditioned on both text and melody, offering high-quality stereo output with fewer autoregressive steps compared to other models. Unlike previous methods that rely heavily on complex conditioning, our approach uses simple codebook interleaving strategies to achieve efficient generation. While our model provides powerful control, it still lacks fine-grained adherence to conditioning, particularly with audio. Ethical considerations were addressed through legal agreements and by mitigating dataset diversity concerns. By incorporating features like melody conditioning, we aim for MUSICGEN to be useful for both amateurs and professionals, setting it apart from more complex, traditional models.

In other hand , PosterLlama, a content-aware layout generation method that combines visual and textual understanding using LLMs. By utilizing Visual Question-answering training and depth-guided augmentation, PosterLlama overcomes data scarcity and inpainting artifacts. Our experiments show that it outperforms existing models, achieving diverse, conditional generation and demonstrating robustness with small datasets, making it suitable for real-world applications.

5.1 Future Development

5.1.1 Future Development: Enhanced Multimodal Interactivity

Looking ahead, several key areas provide exciting opportunities for enhancing the capabilities of the application. One potential direction for future development is the introduction of **Enhanced Multimodal Interactivity**. This feature would allow users to make real-time adjustments to both music and layout outputs, enabling an interactive experience where modifications to one modality—such as altering the music—could directly influence the design and layout, or vice versa. This would offer a more dynamic and fluid creative process, where users can experiment and iterate quickly, fostering a more intuitive and efficient workflow.

5.1.2 Future Development: Cross-Modal Conditioning

Another promising area is **Cross-Modal Conditioning**, which would involve using the generated music to influence the layout designs and vice versa. By making use of advanced machine learning techniques, the system could create even more harmonized outputs, where the characteristics of the music (such as tempo, mood, or instruments) could dynamically inform the visual layout design, and the visual elements could shape the musical composition. This would lead to a more synergistic relationship between music and design, producing a more cohesive and integrated final product.

5.2 Conclusion

In conclusion, the integration of **MusicGen** and **PosterLlama** paves the way for a new era in media creation—one that is accessible, customizable, and efficient. With future developments focusing on enhanced interactivity, cross-modal conditioning, and scalability, this platform holds great potential for revolutionizing the way creative assets are produced across a wide range of industries.

Bibliography

[1] Seol, J., Kim, S., Yoo, J.: PosterLlama: Bridging Design Ability of Language Model to Contents-Aware Layout Generation. arXiv (2024), `https://arxiv.org/abs/2404.00995`

[2] Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., Défossez, A.: Simple and Controllable Music Generation. arXiv (2024), `https://arxiv.org/abs/2306.05284`

[3] Shen, J., Pang, R., Weiss, R.J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R.J., Saurous, R.A., Agiomyrgiannakis, Y., Wu, Y.: Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. arXiv (2018), `https://arxiv.org/abs/1712.05884`

[4] Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z.A., Dieleman, S., Elsen, E., Engel, J., Eck, D.: Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. arXiv (2019), `https://arxiv.org/abs/1810.12247`