Video-to-Music Generation

Scott Merill
Computer Science Department
smerrill@unc.edu

Titus Spielvogel
Computer Science Department
tis@cs.unc.edu

Abstract

Music plays a vital role in enhancing video content, yet automatically generating background music that aligns with a video's pacing, tone, and narrative remains a complex challenge. This work tackles the task of video-to-music generation by proposing a novel method that balances fine-grained temporal synchronization with overall musical coherence. Our approach combines optical flow–based motion embeddings with CLIP-derived visual embeddings through a cross-attention mechanism, enabling the model to capture both dynamic motion and high-level semantic context. These fused representations condition an autoregressive transformer decoder to generate music tokens aligned with the visual input. Remarkably, our method remains competitive with prior work such as VidMuse, despite using five times fewer parameters and being trained on less than one-eighth the amount of data. These findings highlight the critical importance of high-quality, tightly aligned training data, demonstrating that a well-curated subset of the V2M dataset can compensate for smaller dataset size and model scale.

1. Introduction

Music plays a crucial role in enhancing video content, making it more immersive and emotionally engaging. This is evident on platforms like TikTok, where well-synchronized music and choreography often capture massive viewer attention. However, choosing background music that aligns seamlessly with a video's pacing, tone, and narrative is a complex task—one that typically demands both creative skill and technical expertise. In this work, we address the challenge of video-to-music generation: automatically composing background music that not only synchronizes with visual content but also elevates the overall viewing experience.

The automatic generation of video-aligned music has applications in film, advertising, gaming, and social media. Recent advances in multimodal machine learning have made it possible to generate dynamic, context-aware soundtracks. These systems can analyze visual cues—such as motion, scene transitions, and emotional tone—to produce music that adapts in real time.

Despite encouraging progress, significant challenges persist. A major difficulty lies in achieving fine-grained temporal alignment between visual events and musical elements—for example, synchronizing beat drops with action sequences or crescendos with dramatic climaxes. However, synchronization alone is not sufficient. Equally critical is maintaining musical coherence; the generated music must retain internal structure and emotional flow, even as it adapts to shifting visual content. These two goals can often be at odds, making video-to-music generation a fundamentally challenging task.

To balance synchronization with musical coherence, we combine fast-paced motion embeddings with lower-frequency visual embeddings. The motion embeddings allow our model to respond rapidly to short-term visual cues such as quick movements or scene changes, while the lower-frequency embeddings capture longer-range contextual information, including mood, setting, and narrative arcs. By integrating these complementary perspectives, our model generates music feels musically intentional while remaining tightly aligned with the visual narrative.

In this work, we make several key contributions. First, we propose a novel architecture that fuses optical flow–based motion embeddings with CLIP-derived visual embeddings using a cross-attention mechanism, enabling the model to capture both fine-grained temporal dynamics and high-level semantic context. We demonstrate our approach achieves strong alignment between video and music while maintaining musical coherence, outperforming prior work on key metrics like chroma similarity. Finally, we show that high-quality, tightly aligned training data can compensate for smaller dataset size and model scale, suggesting a data-centric path forward for improving video-to-music generation systems.

2. Related Work

2.1. Music Generation

Early approaches to Music Generation relied on symbolic sequence modeling with Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) architectures, which captured short-term melodic and harmonic dependencies but often failed to maintain global coherence or structural richness, leading to repetitive or disjointed outputs [1, 2].

The introduction of transformer-based architectures marked a major breakthrough in music generation. By enabling models to capture long-range dependencies, transformers allowed for the creation of more coherent and structurally complex compositions. Models like *MusicGen* and *STEMGEN* [3] were at the forefront of this innovation, demonstrating the full potential of transformers with fine-grained control over aspects such as style, instrumentation, and overall structure.

More recently, diffusion-based models such as *MusicLDM* [4] and hybrid architectures like *MeLoDy* [5] have redefined the state of the art in audio fidelity and dynamic expressiveness. These models combine the temporal precision of diffusion sampling with the contextual modeling strengths of transformers.

Additional efforts have focused on emotional control and personalization. Emotion-conditioned systems like *ECMusicLM* [6] incorporate affective cues into the generative process, while reinforcement learning models such as *MusicRL* [7] adapt generation based on listener feedback, allowing for dynamic, user-aligned musical outcomes.

2.2. Video-to-Music Generation

Video-to-music generation extends music generation into the multimodal domain. Early work such as *Pseudo Song Prediction* [8] explored visual-to-audio mappings based on pseudo-acoustic similarity, but lacked flexibility and expressive control.

Attention-based models such as *GVMGen* [9] and *Vid-Musician* [10] improved alignment between visual and musical structures using hierarchical attention mechanisms. These systems demonstrated the importance of synchronizing video motion and structure with rhythm and emotion in music.

Flow-matching and contrastive learning approaches like *MuVi* [11] and beat-aware models such as *VMAS* [12] and *VMB* [13] enhanced emotional congruence and rhythm alignment by modeling temporal and perceptual structures more effectively.

Specialized models for dance, such as *D2MNet* [14] and *D2M-GAN* [15], employed beat tracking and adversarial training to tightly couple musical beats with bodily motion. Affect-sensitive and genre-aware systems like *Video2Music* [16] and *CMT* [17] added further nuance through multimodal conditioning and transformer-based fusion.

Our approach is primarily inspired by advancements in transformer-based music generation, exemplified by *MusicGen* [3]'s ability to generate coherent and structured music. We take a similar approach to *VidMuse* [10] by constructing representations of visual and motion information from the input video and using these to guide the music generation process. To incorporate detailed motion information, we also draw inspiration from the field of computer vision, where dense optical flow techniques [18, 19] have been widely used to capture fine-grained temporal dynamics.

3. Methods

Our approach builds on VidMuse [20], but introduces a more efficient and context-aware mechanism for capturing both long-range semantics and fine-grained temporal dynamics. VidMuse predicts music tokens using a rolling attention window over short- and long-term video embeddings. While effective, this method is computationally expensive and constrained by a fixed lookback window.

In contrast, we seed *MusicGen* [21] with a fused video representation that integrates both high-level visual context and detailed motion information. Specifically, we use CLIP embeddings to encode semantic content at a low frame rate, and optical flow to capture motion at a higher frame rate. Visual embeddings $\{v_1, \ldots, v_{N_v}\}$ prioritize long-range semantics, while motion embeddings $\{f_1, \ldots, f_{N_m}\}$ are sampled more densely to capture short-term dynamics, where $N_m > N_v$.

We then apply a cross-attention mechanism, enabling each visual embedding to attend over the motion sequence, fusing spatial and temporal information into a set of representations $\{o_1, \ldots, o_{N_v}\}$. These are projected into *MusicGen*'s token embedding space and used to condition the model for audio generation. We adopt the same training objective as *MusicGen*, predicting the next audio token given previous tokens and the video context. An overview of our framework is presented in Figure 1, where components highlighted in blue denote trainable parameters, and those in light gray are kept frozen. The following subsections provide a detailed breakdown of each component in our architecture.

3.1. Visual Embeddings

To extract semantic visual features, we uniformly sample N_v video frames with a frozen CLIP image encoder [22]. Each frame produces Q patch embeddings of dimension D, yielding a tensor of shape $N_v \times Q \times D$.

Since CLIP is not trained for music generation, we refine these features using a lightweight transformer encoder.

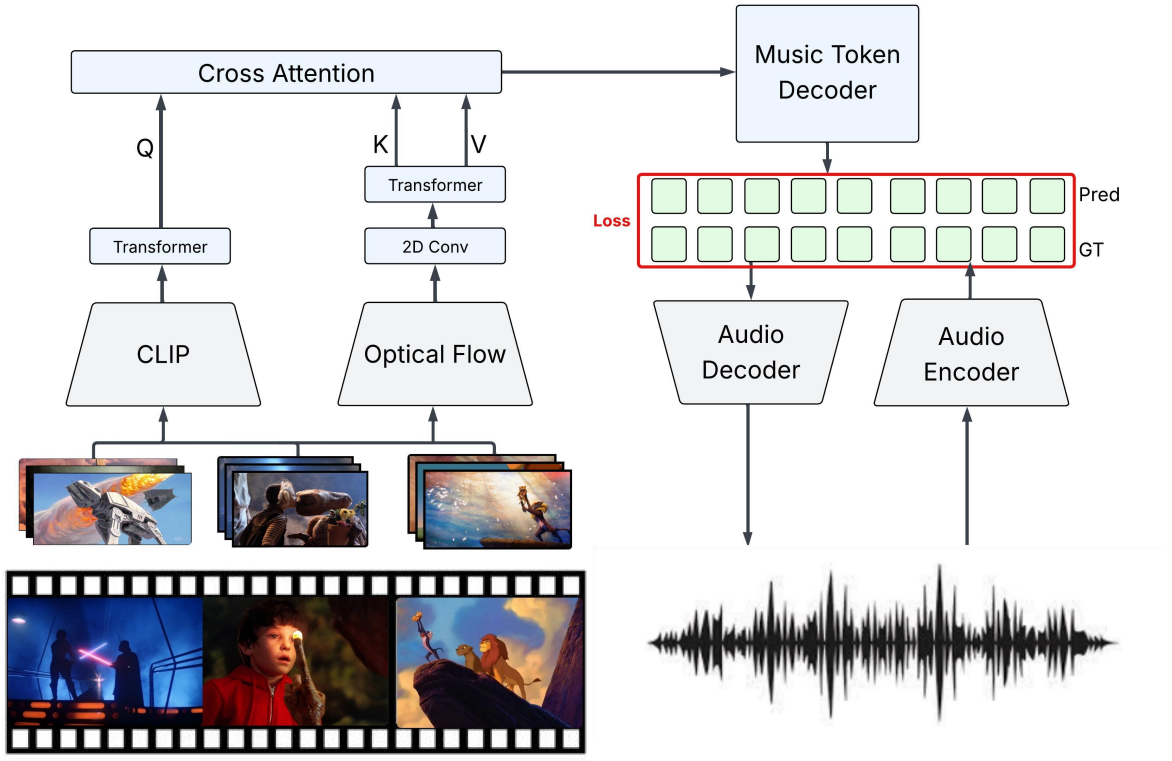

Figure 1. Overview of our framework. CLIP embeddings capture long-range visual semantics at a low frame rate, while optical flow features capture fine-grained motion at a higher frame rate. A cross-attention module fuses the two modalities before projection into *MusicGen*'s embedding space for audio generation. The model is trained as a next token prediction problem with trainable parameters are highlighted in blue.

We first add learnable spatial positional embeddings, then apply the transformer over patches within each frame. To summarize each frame, we apply CLS pooling, resulting in a sequence of N_v visual embeddings $\{v_1, \ldots, v_{N_v}\}$, where each $v_i \in \mathbb{R}^D$.

3.2. Motion Embeddings

To capture short-term motion, we sample N_m grayscale frames at a higher frame rate, such that $N_m > N_v$. Let $X_m \in \mathbb{R}^{N_m \times H \times W}$ denote the frame sequence. We compute dense optical flow between each consecutive frame pair using the Gunnar Farneback algorithm [23], yielding per-pixel displacements $(\Delta u, \Delta v)$.

We convert these displacements to polar coordinates and retain only the magnitude, which reflects motion speed. This results in N_m motion maps $\{m_1, \ldots, m_{N_m}\}$, where each $m_i \in \mathbb{R}^{H \times W}$. We embed each map using a learnable, non-overlapping 2D convolution with a $D \times D$ kernel, producing Q patches per frame. Similar to the visual embeddings, we flatten these patch embeddings and pass them through a transformer encoder to capture spatial context. Finally, we apply mean pooling across patches to produce N_m motion embeddings $\{f_1, \ldots, f_{N_m}\}$, with $f_j \in \mathbb{R}^D$.

3.3. Cross-Attention Fusion

To combine long-range visual context with fine-grained motion dynamics, we apply a cross-attention mechanism between the visual and motion feature sequences. Let $\{v_1, \ldots, v_{N_v}\}$ be the visual queries and $\{f_1, \ldots, f_{N_m}\}$ the motion keys and values, with $v_i, f_j \in \mathbb{R}^D$.

Cross-attention produces a fused sequence $\{o_1, \ldots, o_{N_v}\}$, where each o_i combines the spatial content of frame i with motion information from the full sequence. We project each o_i into the *MusicGen* token embedding space and use these fused representations to condition audio token generation.

3.4. Music Token Decoder

To generate music conditioned on video input, we use an autoregressive transformer decoder that predicts a sequence of discrete music tokens $\{\bar{y}_1, \ldots, \bar{y}_T\}$. At each time step t, the decoder outputs logits for the current token $\bar{y}_t \in \mathbb{R}^{K \times C}$,

where K denotes the number of codebooks and C is the vocabulary size per codebook, following the tokenization strategy used in *MusicGen*.

The decoder consists of a transformer architecture with latent dimension M, allowing for scalable model size depending on computational constraints. It incorporates a cross-attention mechanism over the fused video representations $\{o_1, \ldots, o_{N_v}\}$, which are projected into the decoder's embedding space as $Z \in \mathbb{R}^{N_v \times P \times M}$, where P is the number of projected features per frame. This allows the decoder to attend to both long-range semantic and short-term motion cues from the video.

During training, the model predicts the next token given all previous tokens and the full visual context Z, using a standard next-token prediction objective. At inference, the decoder autoregressively generates music tokens, which are later converted into audio via the audio codec.

3.5. Audio Codec

To convert between continuous audio and discrete tokens, we use a pretrained audio codec that encodes audio into a sequence of quantized codebook indices and reconstructs it back into waveform. The codec represents each audio segment as a grid of size $K \times T$, where T corresponds to the number of time steps and K is the number of codebooks used for quantization.

Formally, let $\mathcal{C}_{\text{encode}}(\cdot)$ denote the audio encoder and $\mathcal{C}_{\text{decode}}(\cdot)$ the decoder. During training, the ground truth audio segment A is passed through the encoder $\mathcal{C}_{\text{encode}}(A)$ to obtain discrete supervision tokens. These serve as training targets for the music token decoder. During inference, the predicted token sequence $\{\bar{y}_1, \ldots, \bar{y}_T\}$ is passed to $\mathcal{C}_{\text{decode}}(\cdot)$ to synthesize the final audio output.

3.6. Training

We train our model using a next-token prediction objective to align audio generation with video content. Given a video segment and corresponding ground-truth audio A, we extract fused video features via the visual encoder and cross-attention module. These are projected and passed to the music token decoder, which outputs predicted logits $\bar{Y} \in \mathbb{R}^{K \times T \times C}$, where K is the number of codebooks, T is the number of timesteps, and C is the vocabulary size.

The ground-truth audio A is encoded into one-hot token representations $Y \in \mathbb{R}^{K \times T \times C}$ using the audio codec encoder $\mathcal{C}_{\text{encode}}(A)$. Each $Y_{k,t,c}$ is 1 if token c is the target at codebook k and timestep t, and 0 otherwise.

The model is optimized using cross-entropy loss between $\bar{Y}$ and Y:

$$\mathcal{L} = -\frac{1}{KT} \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{c=1}^{C} Y_{k,t,c} \log \bar{Y}_{k,t,c} \qquad (1)$$

This loss guides the decoder to predict the correct audio token at each timestep, conditioned on the video input. The audio codec remains frozen during training.

4. Dataset

To train and evaluate our model, we require high-quality datasets composed of video–music pairs. To obtain these pairings, we utilized publicly available YouTube-based datasets. We employed yt-dlp to download videos at scale. However, large-scale data acquisition from YouTube posed challenges due to detection mechanisms and rate limiting. To address this, we modified several parameters within yt-dlp to emulate typical user behavior and reduce the likelihood of being flagged as automated traffic.

Additionally, we routed all download requests through publicly available proxies. While such proxies are often unstable, their abundance made our strategy effective: when one proxy failed, we seamlessly moved to the next in a precompiled list. This proxy rotation mechanism allowed us to successfully download large-scale datasets from YouTube while avoiding rate limits and throttling.

With this strategy we downloaded a subset of the V2M Dataset [20], which comprises approximately 360,000 high-quality video–music pairs. Due to the scale of the full dataset, we selected a 20,000-sample subset, extracting only the first 30 seconds of each video. V2M encompasses a wide range of media types—including movie trailers, advertisements, and documentaries. This diversity ensures broad coverage of stylistic and temporal patterns across various audiovisual domains. Each video–audio pair was carefully curated to maximize audiovisual alignment, making V2M particularly well-suited for training dense video-to-music generation models [24, 20].

Following the download, we extracted raw video frames and applied music source separation to isolate instrumental tracks by removing vocals. This step was necessary, as our model is designed to generate background music without lyrics. We show a basic outline of this data extraction in Figure 2

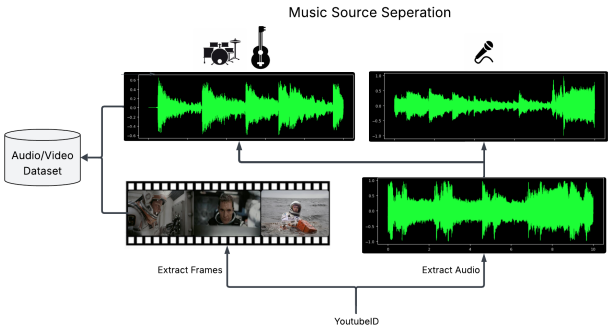

Figure 2. Data Extraction Pipeline.

5. Experiments

5.1. Implementation Details

We trained our model on 13293 samples from the V2M20K dataset. For optical flow computation, videos were sampled at 4 frames per second, while raw video frames were sampled at 2 frames per second to generate visual embeddings. These embeddings were extracted using CLIP ViT-B/32, which divides each frame into 32×32 pixel patches. To decode music from the visual input, we utilized the *MusicGen* Small variant, which has 300 million parameters—significantly fewer than the Medium and Large variants, which contain 1.5 billion and 3.3 billion parameters, respectively. Audio was compressed using Encodec [25] at 32 kHz for monophonic output. The model was trained with the AdamW optimizer and a batch size of 64 samples for 10 epochs with 416 updates per epoch. The initial learning rate was set to 3.5×10^{-5}, with a linear warmup of 4000 steps and cosine decay to zero each cycle. An EMA with decay 0.99 was applied on the GPU. While training was conducted for 10 epochs, we observed that both training and validation losses plateaued after 10 epochs. The entire training process took 17 hours on 8 L40s GPUs.

5.2. Evaluation Metrics

While we consider human evaluation as the gold standard for assessing subjective information like music, large-scale human studies can be prohibitively time-consuming and costly, especially within the limited timeframe of a single academic semester. As a more efficient and scalable alternative, we evaluate model performance using three key quantitative metrics: Kullback-Leibler Divergence (KLD), Fréchet Audio Distance (FAD), and Chroma Cosine Similarity.

KLD measures the divergence between the distributions of generated and ground-truth audio samples. It is important to note that KLD is not a true metric as it does not satisfy the triangle inequality. Specifically, the Kullback-Leibler divergence between $KL(P||Q)$ is not equal to $KL(Q||P)$. As a result, we compute KLD in both directions, where P represents the generated audio and Q represents the ground truth.

FAD, on the other hand, quantifies how closely the generated audio resembles real audio by comparing feature distributions in the VGGish embedding space [26]. This allows for a direct comparison of the audio at a feature level, providing insight into the overall quality of the generated samples.

The Chroma Cosine Similarity [3] compares the chroma feature vectors of two audio samples. A chroma feature vector has 12 dimensions, each corresponding to one of the pitch classes in music (e.g., C, C#, D, D#, etc.), representing the harmonic content of the audio at a given moment. For instance, a chroma vector might quantify how much of the note "C" is present, along with the intensities of the other 11 pitch classes. By measuring the cosine of the angle between chroma feature patterns [27], this metric effectively captures the similarity between the harmonic structures of two audio clips, allowing for an assessment of their tonal and chordal alignment.

Together, these metrics provide a comprehensive and scalable framework for evaluating generative music models, offering a quantitative alternative to human judgment.

5.3. Results

We compare our model against VidMuse, as it shares the core idea of seeding *MusicGen* with visual context. While VidMuse was trained across all variants of *MusicGen*, our primary comparison is with the version trained on *MusicGen* Small. The authors, however, only provided a model checkpoint for VidMuse-M so we will compare against this model. It's also important to note that this model was trained on 360k samples while our model was only trained on 20k. With this in mind, we show the results for our model in Table 1 and use bold font to highlight the best values in each row.

From Table 1, we find that the VidMuse model that is 5 times larger and trained on 8 times more data only marginally outperforms our model. In fact, our model is able to achieve higher chroma similarities. Furthermore, the chroma simiilarities are arguably the most important metric in this table since they are the only metric that incorporate musical domain knowledge.

We hypothesize that several key factors contribute to the performance of our model. First, the VidMuse framework utilizes a fusion of both short- and long-term features through spatial and temporal cross-attention mechanisms. In contrast, our approach focuses exclusively on cross-attention along the temporal axis. By narrowing the attention scope to the temporal dimension, our model may find it easier to identify and align the critical features required for synchronizing video and audio. This reduction in attention complexity likely helps eliminate irrelevant spatial information, allowing the model to more effectively capture the time-based correlations between video and audio signals, which are crucial for high-quality audio-visual alignment. Future work will explore this hypothesis by incorporating spatial attention into our model to assess its impact on performance.

Second, the integration of CLIP features alongside optical flow offers a more robust mechanism for capturing both long-range dependencies and fine-grained contextual information compared to VidMuse's use of rolling windows for temporal modeling. CLIP, with its ability to encode semantic visual information into embeddings, provides a rich and contextually meaningful representation of video con-

Model	Params	Training Samples	FAD ↓	KLD(P$\|Q$)↓	KLD(Q$\|P$)↓	Chroma ↑
VidMuse-M	1.9B	360k	**2.13**	**1.32**	**0.98**	0.056
Our Model-S w/o motion	487M	20k	2.89	1.48	1.08	**0.059**
Our Model-S	487M	20k	2.87	1.48	1.08	0.058

Table 1. Comparison of our model against VidMuse-M across various evaluation metrics. While VidMuse-M benefits from a significantly larger parameter count and training dataset, our model demonstrates competitive performance—particularly in Chroma similarity, which reflects musical coherence and alignment..

tent. This enhances the model's ability to align audio and video in a semantically coherent manner. Additionally, optical flow, specifically designed for detecting motion between consecutive frames, has proven highly effective in a range of computer vision tasks, particularly in modeling temporal visual dynamics. By leveraging these two powerful features, our model gains a superior understanding of motion and scene transitions, which are critical for maintaining synchronization between audio and video over time. The combination of motion and semantic information encoded directly into the model's features allows it to capture both long-term dependencies and fine-grained details in the alignment process, potentially improving performance over VidMuse's reliance on less targeted temporal windowing.

Third, we believe the high quality of the V2M dataset plays a big role in enhancing our model's performance. The VidMuse team carefully curated the dataset to ensure that video and audio components are tightly aligned, significantly reducing label noise. This data quality enables our model to focus on learning the essential mapping between video and audio, without the distractions of noise or misalignments. Given the meticulously aligned nature of the dataset, there is less need for an excessively large training dataset to achieve strong performance. This is further supported by our observation that both training and validation accuracies plateaued rapidly, after only 10 epochs of training.

5.4. Ablations

To evaluate the role of fine-grained motion features, we conduct an ablation study by removing the optical flow embeddings from our model. In this configuration, the model relies solely on CLIP embeddings, which are processed by a transformer and subsequently projected into the *MusicGen* embedding space. This ablation allows us to isolate and assess the contribution of motion-specific information to the alignment between video and music modalities. The results, presented in Table 1, demonstrate a marginal impact when motion features are omitted.

One potential explanation for this limited impact is the nature of our training objective, which focuses on matching ground truth music clips given a video input. In this setup, the model without motion features is already capable of achieving strong performance, and the addition of motion features offers minimal incremental benefit under the current supervision signal. This is reflected in the similar FAD scores between the full model and the ablated variant.

To more effectively leverage motion features, we believe that the training objective should explicitly incorporate a motion-sensitive alignment component, akin to beat alignment scheme in [12]. However, introducing such an objective would require a shift in evaluation strategy. Traditional metrics such as FAD and KLD, which are designed to measure similarity to ground truth audio, may not adequately capture the benefits of motion-aware alignment. In fact, optimizing for motion synchronization could degrade performance on these metrics, as the model may deviate from the literal ground truth in favor of better audiovisual coherence. Furthermore, adding such a component would require alternative evaluation metrics—such as AV-ALIGN [24]—that reflect the temporal and semantic congruence between video and music. Exploring such alignment-aware training objectives and complementary evaluation protocols is a promising direction for future work. It opens the door to models that go beyond literal reconstruction, aiming instead for perceptually meaningful synchronization between modalities.

6. Conclusion

Our work introduces a novel approach to video-to-music generation by effectively combining both visual and motion cues. By leveraging a cross-attention mechanism to fuse CLIP embeddings for high-level visual semantics and optical flow for fine-grained motion dynamics, we demonstrate that this integrated approach captures essential temporal needed for good video-to-music alignment. Notably, our method outperforms VidMuse on certain evaluation metrics, such as chroma similarity, which directly reflects the model's capacity to understand and align the harmonic structure of the music with the visual input.

One key observation from our research is the role of high-quality data in achieving strong model performance. The V2M dataset's careful curation, ensuring tight audio-visual alignment. We believe this tight alignment reduces label noise and allows our model to model to remain competitive VidMuse despite the smaller dataset and fewer parameters. The rapid plateauing of training and validation losses further supports the idea that well-aligned data can

drastically reduce the need for extensive training samples.

Looking ahead, a promising direction for future research is the incorporation of explicit video-audio alignment objectives during training. This would necessitate moving beyond standard evaluation metrics such as FAD and KLD toward alignment-specific metrics that better capture perceptual and temporal coherence between modalities. Additionally, incorporating human evaluations could provide a more nuanced understanding of the perceptual quality and emotional relevance of the generated music.

Finally, we aim to explore the integration of textual inputs—such as video titles, descriptions, or user-provided prompts—as an additional modality. Incorporating such semantic information could enrich the contextual understanding of the video content, enabling the model to generate music that is not only temporally and visually synchronized but also semantically aligned. This multimodal extension holds significant potential to enhance the expressiveness, relevance, and interpretability of the generated music, thereby expanding the practical utility of video-to-music generation in diverse real-world applications.

References

[1] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," *ICML*, 2012.

[2] C. Waite, "Generating long-term structure in songs and stories." `https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn`, 2016.

[3] Q. Kong *et al.*, "Musicgen: Controllable music generation with text and chords," *arXiv preprint arXiv:2306.05284*, 2024.

[4] Q. Kong *et al.*, "Musicldm: Music generation with latent diffusion models," *arXiv preprint arXiv:2401.00001*, 2024.

[5] B. Lam *et al.*, "Melody: A diffusion model for melodic multi-instrument music generation," in *NeurIPS*, 2023.

[6] S. Dash and K. Agres, "Ecmusiclm: Emotion-conditioned music generation via language modeling," *arXiv preprint arXiv:2403.01895*, 2024.

[7] Y. Zhang *et al.*, "Musicrl: Reinforcement learning for personalized music generation," *arXiv preprint arXiv:2403.04000*, 2024.

[8] C. Lin *et al.*, "Generating pseudo songs from images using deep neural networks," in *ACMMM*, 2016.

[9] T. Zuo *et al.*, "Gvmgen: Generative video-music model via hierarchical attention," *CVPR*, 2025.

[10] Z. Li *et al.*, "Vidmusician: Video-to-music generation with multi-level alignment," *arXiv preprint arXiv:2402.00065*, 2024.

[11] Y. Li *et al.*, "Muvi: Emotionally aligned video-to-music generation with flow-matching," *arXiv preprint arXiv:2402.01432*, 2024.

[12] Y.-B. Lin, Y. Tian, L. Yang, G. Bertasius, and H. Wang, "Vmas: Video-to-music generation via semantic alignment in web music videos," *arXiv preprint arXiv:2409.07450*, 2024.

[13] R. Wang *et al.*, "Vmb: Video-music bridging with retrieval-augmented generation," *arXiv preprint arXiv:2401.08345*, 2024.

[14] J. Huang *et al.*, "D2mnet: Dance-to-music generation using cross-modal transformers," in *CVPR*, 2024.

[15] X. Zhu *et al.*, "D2m-gan: Dance-to-music generation with adversarial training," in *ACMMM*, 2022.

[16] H. Kang *et al.*, "Video2music: Affective multimodal music generation from video," *arXiv preprint arXiv:2310.11245*, 2023.

[17] W. Di *et al.*, "Cmt: Cross-modal transformer for video-to-music generation," in *ICCV*, 2021.

[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, vol. 81, pp. 674–679, 1981.

[19] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[20] Z. Tian, Z. Liu, R. Yuan, J. Pan, Q. Liu, X. Tan, Q. Chen, W. Xue, and Y. Guo, "Vidmuse: A simple video-to-music generation framework with long-short-term modeling," 2024.

[21] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 47704–47720, 2023.

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.

[23] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pp. 363–370, Springer, 2003.

[24] G. Yariv, I. Gat, S. Benaim, L. Wolf, I. Schwartz, and Y. Adi, "Diverse and aligned audio-to-video generation via text-to-video model adaptation," 2023.

[25] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," 2021.

[26] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," 2017.

[27] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.