GENERATIVE AI-ASSISTED MUSIC VIDEO GENERATOR

Mr. Kamalakkannan R[1], Harini Akshitha K[2], Shanmathi M[3], Sherlin Nisha A[4], Swetha J[5] ,Vinoda SRD [6]

[1]*Assistant Professor, Department of CSE (IoT & Cybersecurity including Blockchain Technology),*

[2,3,4,5,6]*Student, Department of CSE (IoT & Cybersecurity including Blockchain Technology) SNS College of Engineering, Coimbatore, India.*

ABSTRACT:

Generative AI technologies have transformed music and video creation by automating the production of high-quality material using powerful algorithms. This abstract examines major approaches such as Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), and Transformer models, focusing on their applicability in the creative industries. The study examines the history of artificial intelligence (AI) in music and video production, beginning with early efforts like David Cope's efforts in Musical Intelligence (EMI) and progressing to modern platforms like OpenAI's MuseNet and Google Magenta. It also discusses the issues of originality, variety, and the necessity for standardised evaluation procedures when evaluating AI-generated work. Future research areas include increasing user control over the generating process and investigating the integration of multimodal information for richer creative outputs. This review seeks to give a complete overview of the existing scene and future prospects for generative AI in the arts.

KEYWORDS: Generative AI, Audio Generation, Video Generation, Artificial Intelligence, Transformer Models, Deep Learning, Content Creation, Machine Learning, Multimedia, AI Ethics.

I. INTRODUCTION

The rapid development of generative artificial intelligence (AI) has opened up exciting new possibilities in multimedia content creation, notably in video and music production. One of the most fascinating uses of generative AI is its capacity to convert static images into dynamic movies, which has enormous implications for webtoon, comic, and anime artists. Artists may bring their characters to life using AI algorithms, generating animated sequences that enrich storytelling and engage viewers in unique ways (Kumar et al., 2021). The ability to collaborate not only improves the creative process, but it also allows for faster content development, saving creative people significant time.

The AI-driven solution presented in this study is intended to improve and expedite the video production process. Video generation from text prompts, dynamic video generation from static pictures, and effective subtitle generation are all made possible by the system. It also incorporates music from services like Spotify and provides features for smoothing out noise and improving audio quality. Applications like webtoon visual and verbal anime creation benefit greatly from the time savings enhanced quality that may be achieved by using AI for these jobs. The method offers a revolutionary toolkit for contemporary content producers and demonstrates the potential of AI in multimedia creation.

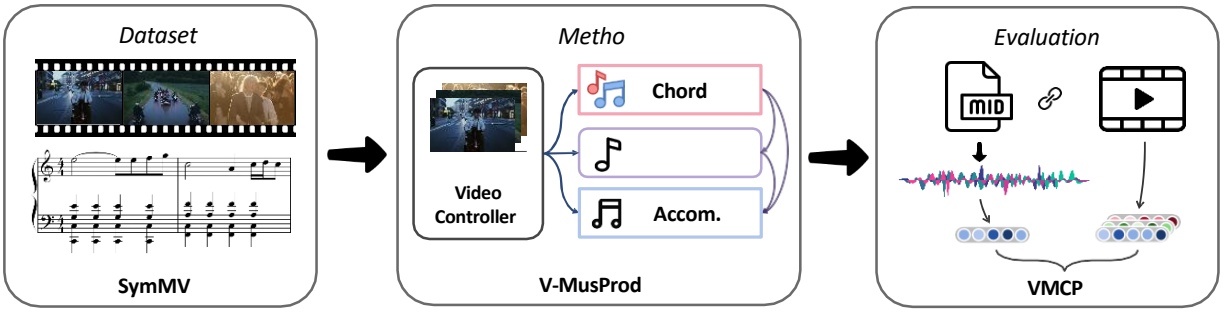


Figure 1: Our framework overview. We find the following solutions to these problems of establishing background music for videos. Left: We focus on providing the first throughly annotated dataset of videos and symbolic music. Middle: To lead distinct phases of music production, we separate different video components and split music generation in R. Kamalakkannan, Y. S. Kumar, D. S. G. R. Divya, S. M. N. Sai Monish Nithin, and S. N. Sowndheriya, "IoT Based V2V Communication Using Li-Fi Technology," YMER Journal, vol. 23, issue 1, pp. 298, January 2024.
o three progressive stages: chord, melody, and accompaniment (accom.). Right: To measure the relationship between produced music and input video, we therefore provide brand-new review metric called Video-Music CLIP Precision (VMCP).

II. RELATED WORK

Recent developments in generative AI have greatly enhanced the synthesis of audio and video, providing tools for effectively producing, editing, and improving multimedia material. An outline of the state-of-the-art in this area is provided below, emphasizing its applications, approaches.

Text-to-Audio and Text-to-Video Generation: Text-to-image frameworks like Stable Diffusion have been expanded to incorporate video and audio synthesis via generative models like diffusion-based architectures. Latent diffusion techniques are used by models such as Tune-A-Video and VideoCrafter to produce coherent films that are directed by text descriptions. In a similar vein, text-to-audio programs like Make-An-Audio and Audiobox provide control over tone, style, and emotion by producing high-quality audio from textual prompts. VideoComposer and other image-to-video transition systems concentrate on turning still pictures into moving sequences, allowing for artistic uses like turning drawings into animated films. for comics or webtoons. To guarantee consistency between frames while preserving creative freedom, these models make use of temporal dimensions in latent spaces.

Multimodal Generative AI: Text, audio, and video may now be handled simultaneously thanks to unified frameworks. By offering end-to-end solutions from text input to completely synchronized video and audio outputs, this integration streamlines processes for multimedia creation-11.

Music Production and Audio Enhancement: AI programs such as Voicebox and VALL-E improve audio quality by eliminating artifacts and noise while producing speech or music. These developments make it possible to include voiceovers or music into videos with ease, meeting a variety of use cases such as music videos, podcasts, and video narration□14□. Problems and Prospects: Although generative AI has developed quickly, there are still issues in attaining fine-grained controllability and guaranteeing temporal coherence.

Dataset	?ide o.udi o MID I ;enr e ;hor d [elod y ɔnalit y							Video Content	Size	Length (Hours)
MAESTRO [17]	?	✓	✓	?	?	?	?	-	1,276	198.7
POP909[50]	?	?	✓	✓	✓	✓	✓	-	909	70.0
HIMV-200K[20]	✓	✓	?	?	?	?	?	Music Video	200,500	-
TikTok[56]	✓	✓	?	?	?	?	?	Dance Video	445	1.5
AIST++[32]	✓	✓	?	✓	?	?	?	Dance Video	1,408	5.2
URMP[31]	✓	✓	✓	?	?	?	?	Music Performance	44	33.5
SymMV (Ours)	✓	✓	✓	✓	✓	✓	✓	Music Video	1,140	76.5

Table 1: A comparison of several music databases. The proposed SymMV dataset is the first to incorporate both video and symbolic music pairings for video background music production. Our collection also includes a variety of musical comments and metadata, such as genre, chord, and melody. In the first two rows, we provide references to prominent symbolic music datasets.

III. THE DATASET

We compile the first video-music dataset that uses symbolic representations to match musical videos with their piano versions. With a combined duration of 76.5 hours, the gathered SymMV dataset includes 1140 pop piano songs in both MIDI and audio format together with the official music video. SymMV was divided into three sets: the test set (70 pairings), validation set (70 pairs), and training set (1000 pairs). Additionally, our dataset contains a variety of annotated metadata, including chord progression, tonal-ity, as well as rhythm. An example of our dataset is displayed in Fig. 2. Comparisons with current video-music datasets are shown in Tab 1.

IV. DATA COLLECTION

There are a lot of video-music pairings on the Internet. Specifically, there is a significant creative and rhythmic relationship between music and music videos. They are also good for understanding intrinsic video-music linkages since they feature a lot of scenarios, actions, and camera angles. Therefore, our goal is to create a dataset of music videos and paired symbolic music for the purpose of creating video background music. It might be simple to locate music videos or piano covers alone, but it can be challenging to gather matching pairs. In order to overcome this difficulty, we first gather piano covers from YouTube channels that provide professional piano lessons that range in audio and melodic quality from fair to excellent. We analyze the metadata after downloading the music and its metadata, then utilize the parsed song title and vocalist as search terms.

V. DATA ANNOTATIONS

Melody and Accompaniment. Common pop music is wellstructured and can be decoupled into melody and accompa- niment. *Melody*, a combination of pitch and rhythm, consti-tutes the most memorable aspect of a song. It is easier for people to perceive melody than other music parts. Hence, melody plays an essential role in a music piece. *Accompa- niment* is correlated with the chord sequence and melody, serving as the background sound effect to harmonize the foreground melody [28], and can bring different auditory sensations. Given music note sequences, we first quantize their duration to the 16th note and implement the Skyline algorithm [47] to separate the melody and accompaniment.

Chord Progression. *Chords*, several notes in a certain ver- tical configuration that sounds harmonic, run through the whole music piece and play an important role in setting the base tone of music. Moreover, chords convey strong emotions with several unique features like color and ten- sion, *e.g.*, major chords bring a feeling of brightness, while minor chords sound relatively dim [40]. We provide chord progression to further control the music generation process. We adopt an open-source rule-based algorithm [1] to extract chords from MIDI files. We observe a long-tail distribution of chord progressions, where more than half of the chords oc- cur less than 10 times. To mitigate this problem, we narrow down the chord templates to 12 root notes and 10 qualities, which covers mostly used types in pop songs.

Tonality. *Tonality* is the general term for *tonic* and *mode* of a key. It reflects the hierarchy of stability, attractions, and directionality in music work. The tonic chord is considered to be the most stable chord in tonality, and it determines the name of the key. Mode represents a type of musical scale centered on the tonic, which can be mainly divided into two types: major and minor modes. In our dataset, we provide the tonality annotation using Krumhansl- Schmuckler algo- rithm [29] to predict tonality from MIDI files and represent music keys using 12 tonic and 2 mode types.

Rhythm. We estimate the beat and downbeat positions from audio using the RNN-based model [5], which corresponds to the fine-grained rhythm. Then, we calculate the tempo from beat and downbeat positions to represent the global rhythm.

Metadata. We also provide additional metadata of our music dataset, such as genre and lyrics. We use ShazamIO[2] to search for lyrics and genres of music in SymMV. These metadata are helpful for data analysis and may benefit future applications, *e.g.*, text-to-music generation [3, 23].

[1]https://github.com/joshuachang2311/chorder/ [2]https://github.com/dotX12/ShazamIO

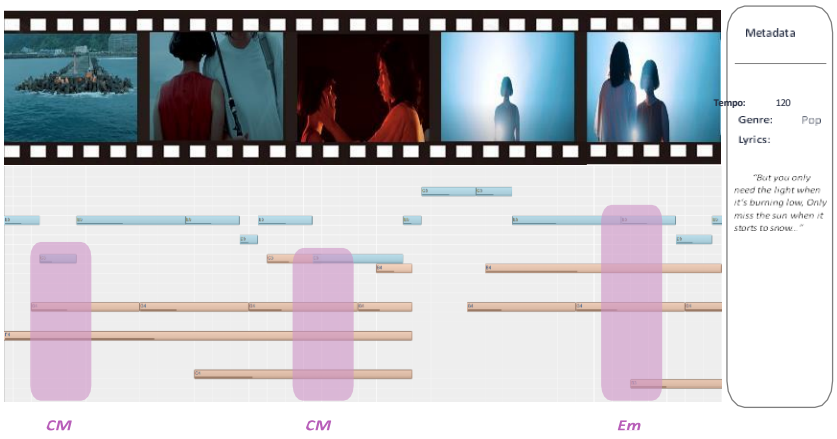


Figure 2: **Illustration of sample In SymMV.** Sample in our SymMV dataset includes paired music and video, music feature annotations, and related metadata.

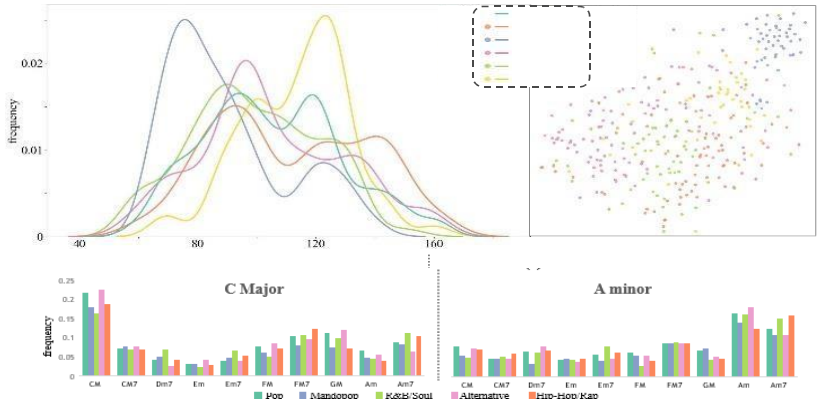


Figure 3: **Visualization of data statistics.** (a) Probability density function of beats per minute in different genres. (b) T-SNE visualization of visual features and genres. (c) Chord distribution in different genres. We show the high correlation between not only music and genre but also video and genre.

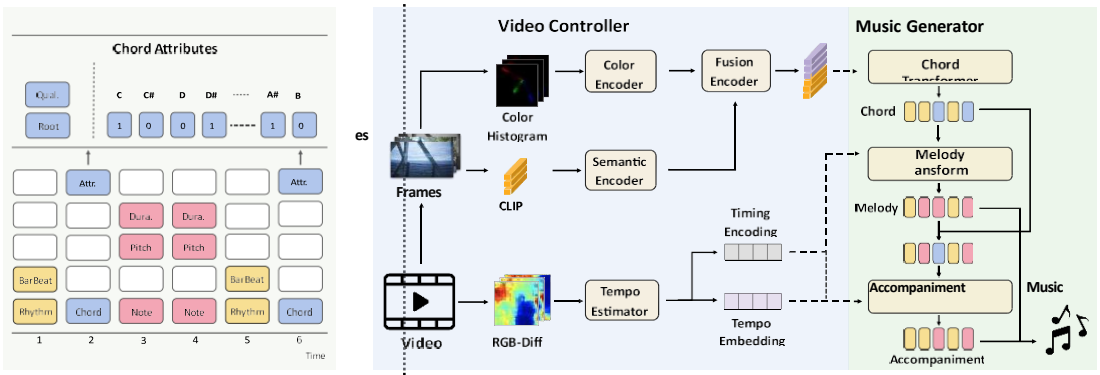
VI. DATA ANNOTATIONS

To ensure the quality of our dataset and determine video- music relationships , we provide a detailed analysis of differ- ent music and video features. Genre, an attribute shared by both modalities, is convenient for us to analyze with visual- ization tools. Since our dataset contains video-music pairs with more than 10 genres, we use genre as a bridge betweenvideo and music to explore their distinct features. We trans- pose all major music to C major (C) and minor music to A minor (a) to remove the influence of tonality on our analysis.

Chord and Genre. We count the frequency of chords in different genres of music. To ensure statistical significance,we choose the five most frequent genres and ten chord typeswith high frequency and high variance. As Fig. 3 (c) shows, the final chord distribution meets our expectations. In *pop* music, the most steady chords, *e.g.*, CM, FM, and Am, oc- cupy the largest proportion, in line with the stability of popmusic. In contrast, some rarely seen chords have a high frequency in *R&B/Soul*, such as Dm7 and Am7, in order to create a richer and more diverse harmonic palette. *Alter- native*, as a type of *Rock*, tends to favor simpler chords, so the occurrences of seventh chords are less frequent. As for *Hip-Hop/Rap*, the singing part is less melodic, and therefore, there are generally more seventh chords to provide accom- paniments with more space for expression and to fill the harmony space. We provide more analysis in the Appendix.

Rhythm and Genre. We use kernel density estimate (KDE) to visualize the distribution of beats per minute (BPM) of mu- sic in various genres. As shown in Fig. 3 (a), *Dance* and *Hip- Hop/Rap* tend to have higher BPM. Curves in other genres display two distinct peaks, representing the BPM of slow- paced and fast-paced songs, respectively. Notably, there is no obvious peak corresponding to fast songs in *R&B/Soul*.

Visual Features and Genre. As for visual features, we first extract 512-dimensional CLIP features from video frames at an FPS of 6. Then we compute the average of these features at time dimension to generate the visual feature of the whole video. We use t-SNE [48] to project visual features into a 2-dimensional space. We ignore *Pop* due to its complexity and select CLIP features of the other five genres to conduct the cluster analysis. Visual features are generally clustered by genre in Fig. 3 (b), demonstrating high correlations.



(a) Music Representation (b) Pipeline of V-MusProd

Figure 4: **Illustration of our method V-MusProd. Left**: We group multiple music attributes into one event-based token (each column). Different colors indicate different types of tokens. **Right**: We extract semantic, color, and motion features to guide the music generation process. The three types of features serve as inputs for different stages of the decoupled music generation model, which contains Chord, Melody, and Accompaniment Transformers.

Method

We propose a novel music generation framework named V-MusProd, to tackle the challenging video background mu- sic generation task. Our framework is shown in Fig. 4, which consists of a video controller and a music generator. The video controller extracts visual and rhythmic features and fuses them as the contextual input of the music generator.

The music generator decouples the music generation process into three progressive stages that are independently trained: Chord, Melody, and Accompaniment. At inference time, melody and accompaniment tracks are merged together to form a complete music piece. The progressive generation pipeline allows for the use of decoupling control on different generation stages, which improves the correspondence be- tween videos and music. We elaborate on each component in the following subsections.

Video Controller

Directly using raw video frames as conditional input is difficult for model to learn the correspondence between two different modalities. Thus, it is important to design and extract meaningful features from video as intermediate rep- resentations to simplify the learning process. Considering the style and rhythm relationship between music and video, we extract semantic, color, and motion features separately to guide the music generation model.
Semantic Features. We use pretrained CLIP2Video [13] as the extractor to encode raw video frames into semantic feature tokens without finetuning. The CLIP2Video model builds upon the CLIP encoder [36], which is pretrained on billions of image-text pairs, and further uses a temporal dif- ference block to learn the temporal context across frames. The extracted features are supposed to contain representa- tions of different video semantics, *e.g.*, scenery, sports, and crowds, which are closely related to the content of music. **Color Features.** Color in videos can reflect the underlying emotions in a given scene, corresponding with the mood of paired music. We employ color features as one of the control signals for chord generation. Specifically, we extract the color histogram of each video frame, *i.e.*, a 2D feature map proposed in [2], to represent the color distribution in a non- linear manifold. The color histogram projects an image's color into a log-chroma space, which is more robust and invariant to illumination changes.

Semantic and color features are fed into separate trans- former encoders and then concatenated together at the length dimension. We add a learnable embedding to mark whether each token is from color feature or semantic feature, and feed the sequence into a transformer encoder for inter-modality and temporal fusion. The fused output serves as keys and values of cross attention in Chord Transformer.

Motion Features. We compute RGB difference as motion features to determine the music tempo and calculate Tempo Embedding and Timing Encoding. We extract the RGB dif- ference with intervals of 5 frames (0.2 seconds) and map the mean RGB difference of a video to the music tempo. We use a linear projection from the minimum and maximum RGB difference to the tempo range of [90, 130]. The esti-mated tempo is used as the Tempo Embedding in the music generator. We also add Timing Encoding [9] in Melody and Accompaniment Transformers to synchronize the video timing and music timing, which reminds the model of the current token's position in the whole sequence.

Music Generator

The music generator G, consisting of a Chord Trans- former G_C, a Melody Transformer G_m, and an Accompa- niment Transformer G_a, is designed to generate symbolic music conditioned on the extracted video feature. The work- flow can be written as follows:

$xm = Gm(Gc(ys), yr)$,

$xa = Ga(Gc(ys), xm, yr)$, $x = xm \oplus xa$,

where style feature y_S and rhythm feature y_r is produced by video controller C, the final music piece x is composed of the melody x_m and the accompaniment x_a, represents merging the two parts into a single track. $\oplus$

Music Representation. Symbolic music comprises a set of music attributes. To encode the dependencies among different attributes, we design an event-based music rep- resentation inspired by [21, 38]. We define three types of tokens, namely Note, Rhythm, and Chord, as the red, yellow, and blue columns in Fig. 4 (a), respectively. Each token is a stack of attributes. In particular, the Rhythm token comprises the BarBeat attribute, indicating the beginning of each bar or beat; Note token contains the Pitch and Duration attributes; Chord token contains the Root and Quality attributes, *i.e.*, the root note and the quality of chords. Chord can also be represented as chromagrams, a 12D binary vector where each dimension indicates whether a pitch class is activated. An additional Type attribute is applied for all tokens to mark their types. In our implementation, the Chord Transformer only models the Rhythm and the Chord tokens, while the Melody and Accompaniment Transformers model all three token types. To align the generated music with input video with rhythmic information, we add Bar Embedding and Beat Embedding for the absolute bar and beat position of current token, and Tempo Embedding for the music tempo.

Chord Transformer. We adopt a transformer decoder architecture for Chord Transformer to learn the long-term de- pendency of input video feature sequences. The event-based token sequence is added with positional encoding and fed features from video controller are fed as keys and values. each decoder token can only attend to the contextual encoder output within the previous current or next bar.

Accompaniment Transformer Similarly, we also adopt an encoder-decoder transformer to generate the accompaniment sequence. Since accompaniment closely correlates with chords and melody, we merge the generated chord sequence with the melody and then pass the merged sequence to Ac- companiment Transformer as conditional input. We also apply the same bar-level cross-attention mask as in Melody Transformer. Eventually, the generated accompaniment is directly merged with the melody to form the final music.

Implementation Details

We train three stages separately and connect them to form a complete pipeline during inference. We construct transformers [49] based on linear transformer [26] to reduce time consumption. All three stages are trained with cross- entropy loss and teacher-forcing strategy. During inference, we use a stochastic temperature-controlled sampling [19] to increase the diversity of generated samples. We train Chord, Melody, and Accompaniment Transformers for 200, 200, and 400 epochs, respectively, on one V100 GPU. We use fluidsynth[3] to synthesize our MIDI into audio. More implementation details are in the Appendix.

Evaluation Metric

In this chapter, we first extend the vision-language CLIP [36] to the video-music domain and propose a new evaluation metric named Video-Music CLIP Precision (VMCP) to measure the video-music correspondence.

Video-Music CLIP

To build the video-music CLIP model, we adopt the de- sign choice in [46], the state-of-the-art video-music retrieval model.

Specifically,
We first split the input music and video into fixed-length segments and use CLIP [36] and music tagging model [51] to extract visual and audio features sepa- rately. Given the extracted features, we adopt a transformer encoder as the video encoder and music encoder to explore contextual relations and learn a joint multi-modal embedding space. The model is trained with the InfoNCE contrastive loss [4] to map positive video-music pairs closer while push- ing negative pairs further in the CLIP-based joint embedding space. Loss of videos v to music pieces m is defined as:

$$\mathrm{I}_{v\rightarrow m} = -\sum_{i}^{N}\sum_{l}^{L} \log \frac{\exp\left[s(v_{i,l}, m_{i,l})/\tau\right]}{\sum_{j}^{N}\sum_{k}^{L}\exp\left[s(v_{i,l}, m_{j,k})/\tau\right]}$$

where N denotes number of video-music pieces, L denotes number of segments, $s(\ )$ denotes cosine similarity, and τ is a learnable temperature parameter. The music-to-video loss $L_{m\rightarrow v}$ is defined symmetrically.

3https://github.com/FluidSynth/fluidsynth

Methods	Video-Music Correspondence				Music Quality				
	P@5	P@10	P@20	AR	SC	PE	PCE	EBR	IOI
Real (SymMV)	-	-	-	-	0.986	4.197	2.633	0.023	0.184
CMT [9]	8.9	17.7	31.0	33.4	0.990	3.920	2.444	0.074	0.246
w/o semantic	11.6	23.9	42.0	26.1	0.955	2.892	2.310	**0.019**	0.358
w/o color	15.6	**26.6**	**44.8**	**25.1**	0.956	2.732	2.200	0.011	0.330
w/o motion	12.2	22.2	37.9	26.3	0.975	3.010	2.283	0.004	0.261
Video2music	10.8	19.7	33.3	30.0	0.981	**3.990**	**2.639**	0.010	0.229
Video2chord2music	13.7	23.1	43.6	26.0	0.996	2.497	2.036	0.081	0.985
V-MusProd	**15.7**	24.6	**44.8**	25.4	**0.983**	3.940	2.607	0.004	**0.174**

Table 2: **Objective evaluation on SymMV test set.** We evaluate video-music correspondence and music quality with VMCP and music quality metrics. P indicates Precision, where higher is better. AR indicates average rank, where lower is better. For music quality metrics, **closer** to Real is better

To train this model, we need to collect plenty of video- music pairs and ensure that the training dataset should roughly cover the distribution of our dataset. Therefore, we download video clips from YouTube8M dataset [1] anno- tated as "music video" and obtain 20k video-music pairs. After training on the YouTube music video dataset, we fine- tune the model with a small learning rate on audio $\tilde{\text{converted}}$ from SymMV to improve its retrieval performance further.

To train this model, we need to collect plenty of video- music pairs and ensure that the training dataset should roughly cover the distribution of our dataset. Therefore, we download video clips from YouTube8M dataset [1] anno- tated as "music video" and obtain 20k video-music pairs. After training on the YouTube music video dataset, we fine- tune the model with a small learning rate on audio $\tilde{\text{converted}}$ from SymMV to improve its retrieval performance further.

Equipped with the pretrained video-music CLIP model, we design a retrieval-based metric similar to [52]. Given a generated music piece in MIDI format, we first synthesize it into audio and calculate the top-K retrieval accuracy from a pool of N candidate videos using the CLIP model. Specifi- cally, we rank the cosine similarity between the generated sample $m\hat{}$ and its condition video v and M 1 random sampled videos v_i. We consider a successful retrieval if the ground truth video is ranked_in the top-K place. We test the model using all generated samples and compute the success retrieval rate as the final precision score. We set $M = 70$, $K =$ 5, 10, 20. We also calculate the average rank of the ground truth video, where a lower rank implies better correspondence. Overall, the proposed metric is able to measure how well the generated music aligns with the input video. We validate that it shows a high correlation with human judgments in the experiments.

VII. EXPERIMENTS

We conduct comprehensive experiments on our V- MusProd model. In Sec. 6.1, we introduce the compared method CMT [9]. In Sec. 6.2, we evaluate video-music cor- respondence with VMCP and music quality with objective metrics. In Sec. 6.3, we conduct a thorough subjective eval- uation for video-music correspondence and music quality by user study.

In Sec. 6.4, we ablate our design choices and highlight the importance of different features used in video controller to validate the effectiveness of proposed method. In Sec. 6.5, we train V-MusProd in unconditional setting and evaluate its music quality against previous symbolic music generation methods.

We compare V-MusProd with the state-of-the-art video background music generation method CMT [9], the first and only method to generate full-length background mu- sic for general videos. CMT uses purely rule-based video- music rhythmic relationships without paired video-music data. Other video-conditional music generation methods mostly focus on specific video types (*e.g.* dance videos) and require extra annotations (*e.g.* keypoints [56, 45]), which are unavailable in the general setting. We train CMT on SymMV to provide a benchmark.

Metrics. For video-music correspondence, we use VMCP to evaluate objectively, where higher precision and lower average rank are better. For music quality, we select music objective metrics from [11, 53], including Scale Consistency (SC), Pitch Entropy (PE), Pitch Class Entropy (PCE), Empty Beat Rate (EBR), and average Inter-Onset Interval (IOI), which evaluate music by pitches and rhythm. Note that these music quality metrics are not indicated by how high or low they are but instead by their *closeness* to the real data. We use SymMV test set for evaluation.

Results. As shown in Tab. 2, V-MusProd surpasses CMT on VMCP and music quality metrics. This proves our method achieves better video-music correspondence and music qual- ity than the state-of-the-art method.

Metrics	Expert	Non-expert
Music Melody	77%	82%
Music Rhythm	63%	53%
Video Content	63%	63%
Video Rhythm	60%	57%
Chord Quality	63%	-
Accom. Quality	83%	-
Overall Ranking	73%	67%

Table 3: **Subjective evaluation for V-MusProd against CMT [9].** We show preference rates in music quality metrics, video-music correspondence metrics, and expertise metrics.

Subjective evaluation is widely adopted in previous works [9, 21, 54, 22]. We conduct a user study by send- ing out questionnaires. We invite 55 participants, including 10 *experts* with expert knowledge in music composition and 45 *non-experts*. We provide several videos from different cat- egories like scenery, city scenes, and movies. Each video has two pieces of background music generated by V-MusProd and CMT, presented randomly for blindness. The question- naire takes about 20 minutes to complete.

Metrics. Participants are required to compare two back- ground music pieces from several aspects: (1) Music Melody: melodiousness and richness of music theme; (2) Music Rhythm: structure consistency of rhythm; (3) Video Content: correspondence between video content and music; (4) Video Rhythm: correspondence between video motion and music rhythm; (5) Overall Ranking: overall preference of the two samples. Besides, we ask the expert group to evaluate two additional metrics related to music theory: (6) Chord Quality: the quality of chord progression in the music; (7) Accompa- niment Quality: the richness of music accompaniment.

Results. We provide results of preference rate, *i.e.* the per- centage of users who consider our music better than CMT, in Tab. 3. Results show that V-MusProd outperforms CMT (> 50%) in nearly all metrics and user groups, demonstrating better music quality and correspondence with videos. In par- ticular, our model outperforms CMT in both Music Melody and Accompaniment Quality by a large margin, indicating our decoupling generation of melody and accompaniment significantly improves their qualities. The subjective eval- uation results show high correlations with VMCP, which further verifies the effectiveness of our proposed metric. **Ablation on Video Controller.** We ablate the three video control features and evaluate the video-music correspon- dence with VMCP: (1) w/o semantic: no semantic feature input for video controller; (2) w/o color: no color feature. input for video controller;
(3) w/o motion: fix tempo and do not add timing encoding. As shown in Tab. 2, semantic and motion features are significant for correspondence. We observe that w/o color has similar correspondence with the full model despite lower music quality. We attribute this to the fact that music tonality is connected with the color of videos. The original keys have already recorded the infor- mation on video colors, so color features are unnecessary for video-music correspondence modeling. If we remove remove the influence of tonality by changing keys, we need color features to capture the video colors. **Ablation on Music Generator.** We further conduct ab- lation study on our music generator with VMCP. To ver- ify the necessity of the decoupled structure, we test two variants of our model: (1) Video2music: uses the out- put video features of the fusion encoder to directly gen- erate target music by a Transformer decoder without de- coupling the structure of chords, melody, and accompani- ment; (2) Video2chord2music: generate chords first and then use chords to generate music without decoupling melody and accompaniment. As shown in Tab. 2, removing any one or more components of chords, melody, and accom- paniment hurts the overall performance of correspondence while having similar music quality. The difference in corre- spondence and music quality validates that decoupled struc- ture is important for music generation and imposing video controls. The improvement of VMCP from Video2music to Video2chord2music shows the effectiveness of decou- pling chords, and the improvement from Video2chord2music to the full model V-MusProd shows the effectiveness of decoupling melody and accompaniment. We note that Video2music sometimes has better music quality. It can be explained that imposing control over music generation can hurt music quality by adding inductive biases. input for video controller; (3) w/o motion: fix tempo and do not add timing encoding. As shown in Tab. 2, semantic and motion features are significant for correspondence. We observe that w/o color has similar correspondence with the full model despite lower music quality. We attribute this to the fact that music tonality is connected with the color of videos.

The original keys have already recorded the infor- mation on video colors, so color features are unnecessary for video-music correspondence modeling. If we remove remove the influence of tonality by changing keys, we need color features to capture the video colors. **Ablation on Music Generator.** We further conduct ab- lation study on our music generator with VMCP. To ver- ify the necessity of the decoupled structure, we test two variants of our model: (1) Video2music: uses the out- put video features of the fusion encoder to directly gen- erate target music by a Transformer decoder without de- coupling the structure of chords, melody, and accompani- ment; (2) Video2chord2music: generate chords first and then use chords to generate music without decoupling melody and accompaniment. As shown in Tab. 2, removing any one or more components of chords, melody, and accom- paniment hurts the overall performance of correspondence while having similar music quality. The difference in corre- spondence and music quality validates that decoupled struc- ture is important for music generation and imposing video controls. The improvement of VMCP from Video2music to Video2chord2music shows the effectiveness of decou- pling chords, and the improvement from Video2chord2music to the full model V- MusProd shows the effectiveness of decoupling melody and accompaniment. We note that Video2music sometimes has better music quality. It can be explained that imposing control over music generation can hurt music quality by adding inductive biases. Our method can be directly used in unconditional music generation. We examine V-MusProd against previous music generation methods: (a) HAT [54]: a hierarchical model built on multiple transformer- based levels to enhance the structure of music, achieving state-of-the-art generation quality;
(b) CP Transformer [21]: transformer-based model using 2D music tokens to compress sequence length; (c) Music Trans- former [22]: the first transformer-based music generation model with improved relative attention. All the above methods are trained on POP909[50] dataset. We directly use their publicly available demos for evaluation. We train our V-MusProd on POP909 without video input, *i.e.* training Chord Transformer without cross attention with video features. The unconditionally generated results are evaluated by music quality metrics in Sec. 6.2. As shown in Tab. 4, our V-MusProd achieves closer results to POP909 training set than previous methods for most of the metrics. This indicates that unconditional music generation can bene- fit from our decoupling structure.

VIII. EXPERIMENTS

In this paper, we have introduced the SymMV dataset, which contains 1140 videos and corresponding background music with rich annotations. Based on SymMV, we devel- oped a benchmark model V-MusProd. It decouples music into chords, melody, and accompaniment, then utilizes video- music relations of semantic, color, and motion features to guide the generation process. We also introduced the VMCP metric based on video-music CLIP to evaluate video-music correspondence. With VMCP and subjective evaluation, we prove that V-MusProd outperforms baseline model CMT in correspondence both qualitatively and quantitatively.

XI. REFERENCES

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Nat- sev, George Toderici, Balakrishnan Varadarajan, and Sud- heendra Vijayanarasimhan. Youtube-8m: A large- scale video classification benchmark. arXiv preprint arXiv:1609.08675, 2016.

[2] Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. Histogan: Controlling colors of gan-generated and real im- ages via color histograms. In CVPR, 2021.

[3] Andrea Agostinelli, Timo I Denk, Zala´n Borsos, Jesse En- gel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. arXiv preprint arXiv:2301.11325, 2023.

[4] Jean-Baptiste Alayrac, Adria` Recasens, Rosalia Schneider, Relja Arandjelovic´, Jason Ramapuram, Jeffrey De Fauw, Lu- cas Smaira, Sander Dieleman, and Andrew Zisserman. Self- Supervised MultiModal Versatile Networks. In NeurIPS, 2020.

[5] Sebastian Bo¨ ck, Filip Korzeniowski, Jan Schlu¨ ter, Florian Krebs, and Gerhard Widmer. madmom: a new Python Audio and Music Signal Processing Library. In *MM*, 2016.

[6] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

[7] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zis- serman. Vggsound: A large- scale audio-visual dataset. In *ICASSP*, 2020.

[8] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A gen- erative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[9] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video back- ground music generation with controllable music transformer. In *MM*, 2021.

[10] Tan M Dinh, Rang Nguyen, and Binh-Son Hua. Tise: Bag of metrics for text-to-image synthesis evaluation. In *ECCV*, 2022.

[11] Hao-Wen Dong, Ke Chen, Julian McAuley, and Taylor Berg- Kirkpatrick. Muspy: A toolkit for symbolic music generation. In *ISMIR*, 2020.

[12] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi- track sequential generative adversarial networks for symbolic music generation and accompaniment. In *AAAI*, 2018.

[13] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021.

[14] Chuang Gan, Deng Huang, Peihao Chen, and Joshua B Tenen- baum. Foley music: Learning to generate music from videos. In *ECCV*, 2020.

[15] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[16] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Dou- glas Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

[17] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the maestro dataset. In *ICLR*, 2019.

[18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[19] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020.

[20] Sungeun Hong, Woobin Im, and Hyun S Yang. Content- based video-music retrieval using soft intra-modal structure constraint. *arXiv preprint arXiv:1704.06761*, 2017.

[21] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *AAAI*, 2021.

[22] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *ICLR*, 2019.

[23] Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023.

[24] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *MM*, 2020.

[25] Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive sur- vey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

[26] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Trans- formers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[27] A Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *ICASSP*, 2020.

[28] Stefan Kostka and Dorothy Payne. *Tonal harmony*. McGraw- Hill Higher Education, 2013.

[29] Carol L Krumhansl. *Cognitive foundations of musical pitch*. Oxford University Press, 2001.

[30] Sangho Lee, Jiwan Chung, Youngjae Yu, Gunhee Kim, Thomas Breuel, Gal Chechik, and Yale Song. Acav100m: Automatic curation of large-scale datasets for audio-visual video representation learning. In *ICCV*, 2021.

[31] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music per- formance dataset for multimodal music analysis: Challenges, insights, and applications. *TMM*, 2018.

[32] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021.

[33] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019.

[34] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[35] .Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to- image synthesis. In *NeurIPS Datasets and Benchmarks Track*, 2021.

[36] R. Kamalakkannan, B. N. K. Bharathi, K. S. Karthik, C. R. Chandan, and J. S. Jagadish, "Artificial Intelligence Based USB Drive Scanner: Integrating AI with Security," YMER Journal, vol. 23, issue 11, pp. 987, November 2024.

[37] K. R. Kamalakkannan, B. S. Balaji, M. S. Mathan, A. K. ArunKumar, and K. C. Karthikeyan, "Locating Smartphones Using Seeker Tool," YMER Journal, vol. 23, issue 11, pp. 980, November 2024.

[38] R. Kamalakkannan, A. Anantha Krishnan, R. Arjun, and B. Deepak, "Retina Touch: A Seamless HCI Experience with Eye and Hand Integration," YMER Journal, vol. 23, issue 4, pp. 907, April 2024.

[39] R. Kamalakkannan, Y. S. Kumar, D. S. G. R. Divya, S. M. N. Sai Monish Nithin, and S. N. Sowndheriya, "IoT Based V2V Communication Using Li-Fi Technology," YMER Journal, vol. 23, issue 1, pp. 298, January 2024.

[40] R. Kamalakkannan, M. K. R. Mohana Karthikeyan, S. H. G. Shree Harish, S. T. Someshwaran, and V. S. Harikrishna Sai, "Revolutionizing Legal Practice: The Transformative Power of Artificial Intelligence," YMER Journal, vol. 2, issue 2, pp. 17.

[41] R. Kamalakkannan, E. Ajaykumar, S. Hariprasanth, H. A. Dakshanapriya, and R. Bhuvanika, "Design Thinking Based Device to Detect Motion of Trespassers of the Territory Using Arduino Uno & GSM Module," Industrial Engineering Journal, vol. 52, issue 12, no. 2, pp. 174, December 2023.

[42] R. Kamalakkannan, R. Ajay Surya, S. Pranesh, A. Purosh Khan, and K. K. Vinay Kousigan, "Design Thinking Based Automatic Railway Gate Controller Using IoT," Industrial Engineering Journal, vol. 52, issue 12, no. 2, pp. 178, December 2023.

[43] R. Kamalakkannan, V. Dineshkumar, V. Karthick Saran, C. Karthikeyan, and U. Dharshini, "Design Thinking Based Accident Prevention System Using Eye Blink Sensor," Industrial Engineering Journal, vol. 52, issue 8, no. 3, pp. 168, August 2023.

[44] R. Kamalakkannan, N. Vinai, A. Mugundhan, S. Suresh, and G. Rasiga, "Design Thinking Approach and Implementation of IoT Based Gas Detection System," Industrial Engineering Journal, vol. 52, issue 8, no. 3, pp. 112, August 2023.