Extending Visual Dynamics for Video-to-Music Generation

Xiaohao Liu
xiaohao.liu@u.nus.edu
National University of Singapore
Singapore, Singapore

Teng Tu
teng.tu@u.nus.edu
National University of Singapore
Singapore, Singapore

Yunshan Ma*
ysma@smu.edu.sg
Singapore Management University
Singapore, Singapore

Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore
Singapore, Singapore

ABSTRACT

Music profoundly enhances video production by improving quality, engagement, and emotional resonance, sparking growing interest in video-to-music generation. Despite recent advances, existing approaches remain limited in specific scenarios or undervalue the visual dynamics. To address these limitations, we focus on tackling the complexity of dynamics and resolving temporal misalignment between video and music representations. To this end, we propose DyViM, a novel framework to enhance dynamics modeling for video-to-music generation. Specifically, we extract frame-wise dynamics features via a simplified motion encoder inherited from optical flow methods, followed by a self-attention module for aggregation within frames. These dynamic features are then incorporated to extend existing music tokens for temporal alignment. Additionally, high-level semantics are conveyed through a cross-attention mechanism, and an annealing tuning strategy benefits to fine-tune well-trained music decoders efficiently, therefore facilitating seamless adaptation. Extensive experiments demonstrate DyViM's superiority over state-of-the-art (SOTA) methods.

CCS CONCEPTS

• **Information systems** → **Multimedia information systems**; • **Applied computing** → **Sound and music computing**.

KEYWORDS

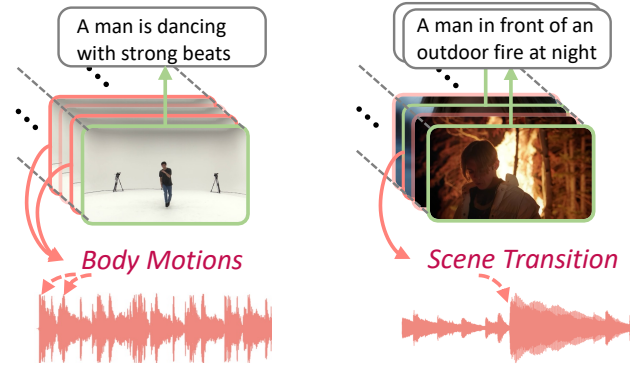Video-to-music generation, Dynamics modeling

*Corresponding author.

Figure 1: Illustrative examples of video-music pairs of dance video (left) and music video (right). Visual dynamics (*e.g.*, body motions or scene transitions) demonstrates temporal synchronization with music rhythm and changes (*e.g.*, beats).

1 INTRODUCTION

Video-to-music generation, the task of creating music that temporally aligns with visual content, has garnered growing interest due to its potential to enrich user experience in media consumption [9, 22, 48]. This task requires capturing not only the visual semantics (*e.g.*, general mood or themes of a video) but also fine-grained dynamics (*e.g.*, camera movements or scene transitions) in real-time, ensuring that the music reflects nuanced shifts within the video flow [11, 24]. Notably, visual dynamics play a substantial role in establishing video-music correlations [32]. For instance, videos may share similar semantics, yet their visual dynamics differ, making them an indispensable factor for music cues [34, 57, 63, 64].

Unfortunately, existing methods in video-to-music generation exhibit limitations from the perspective of visual dynamics. We group them into three main approaches. One approach leverages human-centric dynamics, like body motions [57, 63, 64] or gestures [17, 26], which work limited in specific scenarios like dance or instrumental videos. Another approach maps quantitative visual dynamics, such as motion magnitude or speed, to musical concepts, like beats and note density, with rule-based methods [11, 24], while this relies heavily on expert knowledge. A third approach utilizes various dynamics features extracted from pre-trained models [48], such as I3D [5]; however, it struggles to achieve precise temporal

synchronization. Overall, despite dynamics being involved, these methods remain limited by their inability to generalize nuanced dynamics features and to finely control music generation in sync with temporal changes. These unresolved limitations hinder the generation of music that dynamically adapts to diverse visual cues.

Addressing the challenges in video-to-music generation requires tackling two core issues: effectively capturing complex visual dynamics and resolving the representational misalignment between video and music. First, visual dynamics are intricate and multifaceted, serving as essential cues for musical changes. For instance, as shown in Figure 1, visual dynamics can manifest as body movements in dance videos, directly indicating musical beats, or as scene transitions and camera movements in music videos (MVs) that enhance storytelling. Music aligns with these nuanced visual dynamics, rather than merely reflecting high-level semantics, making it more distinct and contextually relevant to each video. However, recent approaches to handle visual dynamics tend to be superficial, often combining various video encoders without adapting them specifically for music generation, limiting both the generalization for various scenarios and effectiveness [48, 57]. Second, video and music possess inconsistency in their representations, posing challenges for achieving fine-grained temporal synchronization. Video typically operates at frame rates of around 24 frames per second, whereas music is sampled at a much higher rate, such as 32 kHz. Despite sharing the same duration in a video-music pair, these differences in data density and structure complicate alignment. This inconsistency leads recent methods to rely on coarse-level (*e.g.*, short-term context [32]) alignment, resulting in suboptimal synchronization. Although some video-to-audio methods [13, 59] suggest using onset detection to enforce synchronization, they sacrifice flexibility for music generation (*i.e.*, not every dynamics necessarily indicates a musical change and vice versa). Encoding informative visual dynamics flexibly, along with its fine-grained conditioning for music generation across diverse scenarios, remains largely unexplored.

To this end, we introduce a novel framework for enhancing **Dy**namics modeling for **Vi**deo-to-**M**usic generation, named as **DyViM** (/ˌdaɪ.vɪm/). At its core, we propose to model visual dynamics by 1) encoding nuanced frame-wise dynamics features using an optical flow-based method and 2) decoding these dynamics onto music tokens with a fine-grained temporal alignment. DyViM captures subtle and variable dynamics cues by adapting an optical flow-based method [45, 50], without requiring any domain knowledge or specific deign catering to the type of dynamics in the video. Music waveforms are encoded into discrete tokens with a residual vector quantizer (RVQ) model (*i.e.*, Encodec [10]). To achieve temporal synchronization, DyViM interpolates the dynamics features onto these tokens, creating a continuous alignment between dynamics shifts and the music generation. This approach allows the music to respond in real time to the visual dynamics, providing a finer token-level synchronization. In parallel to dynamics modeling, DyViM typically utilizes a pre-trained image encoder (*i.e.*, CLIP [41]) to extract high-level video semantics from keyframes. These features guide the music generation via cross-attention, ensuring thematic coherence. Additionally, an annealing tuning strategy is introduced to reduce over-constraint to the pre-trained music decoder, allowing

more seamless adaptation. Extensive experiments on three datasets validate DyViM's effectiveness, demonstrating its ability to generate music that aligns closely with both dynamic and semantic cues, advancing the field of video-to-music generation. Overall, our contributions are threefold:

- We underscore the crucial role of fine-grained visual dynamics in video-to-music generation, addressing an unexplored gap in leveraging nuanced cues for temporally synchronized music.
- We introduce a novel framework, DyViM, for video-to-music generation that enhances dynamics modeling through a specialized dynamics encoder and token-level dynamics-conditioned generation. Moreover, an annealing tuning strategy is introduced to optimize DyViM.
- We conduct extensive experiments across multiple datasets demonstrate DyViM's superiority, with code and demos provided to support future research.

2 RELATED WORK

We review the literature on video understanding, music generation, and video-to-music generation.

2.1 Video Understanding

Video understanding, a prerequisite for video-to-music generation, has advanced with improvements in representation learning and Multimodal Large Language Models (MLLMs) [2, 44, 53, 54]. In video representation learning, early approaches [5, 52] utilize 3D CNNs to extract spatiotemporal features and capture different levels of dynamics via dual-pathway architectures [14]. Concurrently, other works utilize optical flow prediction as a self-supervised objective to model object dynamics [39, 45, 50]. More recently, transformer-based models [3, 33, 37–39, 46, 47] capture spatiotemporal dependencies and learn robust representations without labels. With the development of MLLMs, through integrating video features with language models, video language models have significant advances in video captioning [2, 36, 44, 55] and QA [2, 53, 54].

2.2 Music Generation.

Music generation has evolved from unconditional models to conditional models with diverse architectures and various conditions. Different paradigms, like GANs [12], transformers [8, 20], and diffusion models [15, 56, 60, 61], are utilized to generate music without auxiliary guidance. Recently, researchers have started to leverage conditional models regarding various inputs. Some models tackle continuation and inpainting tasks [8, 40] by given music itself. Others condition on text or visual inputs to align music generation with user-provided textual input [1, 8], or to leverage visual-music correlations [7, 43]. Notably, recent works incorporate temporal dynamics into visual conditioning to study video-conditioned music generation, which is detailed in the next subsection.

2.3 Video-to-Music Generation.

The video-to-music generation task can be divided into distinct sub-fields depending on the video content, where each type of video is separately formulated as a specific task. Early studies concentrate on generating music from instrument performance videos by
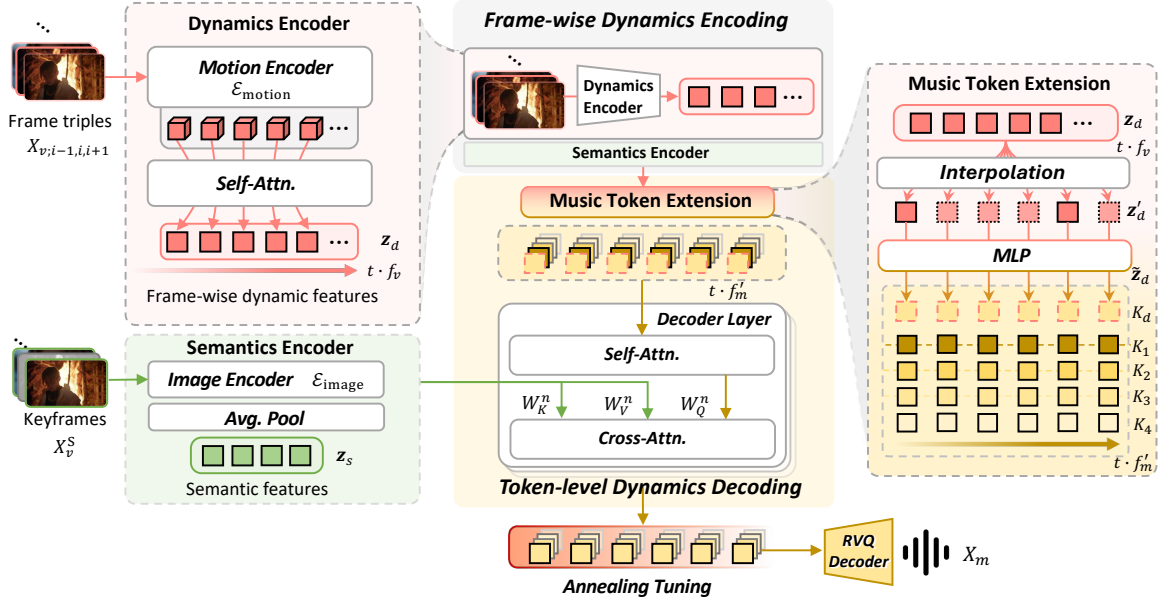
Figure 2: The overall framework of DyViM. We enhance the dynamics modeling by extracting frame-wise dynamics features and interpolating them to extend music tokens to achieve token-level conditioning. Additionally, keyframes provide semantics with cross-attention. An annealing tuning strategy is employed to optimize the model in an effective and efficient manner.

mapping visual cues from playing actions to corresponding musical scores [17, 26, 49] or spectrograms [6]. Subsequent research utilizes motion cues from dance videos to generate synchronized music that reflects the dynamics and emotions of the input [57, 63, 64]. Recently, research has increasingly shifted its focus to music videos or trailers. For example, CMT [11] utilizes predefined rules to extract visual cues from videos, which is then leveraged to guide the music generation process. VidMuse [51] introduces a long-short-term module, which only extracts the overall video embeddings without leveraging the alignment between music and video in the dataset. Similarly, V2Meow [48] utilizes key visual semantics of sparsely sampled video frames, but potentially does not sufficiently exploit the video's temporal features. VMAS [34] emphasizes beat synchronization between video and music by utilizing a rule-based beat extraction method and weighted autoregressive loss. However, this method is limited when the video-music alignment is of low quality. In summary, even though various methods have been proposed for video-to-music generation, they are still limited to handle nuanced dynamics and cannot achieve fine-grained temporal synchronization during generation.

3 PRELIMINARY

We present the problem formulation for video-to-music generation and introduce the backbone of our approach, an autoregressive music generation model.

3.1 Problem Formulation

Given a video consisting of a list of image frames $\mathbf{X}_v \in \mathbb{R}^{T_v \times C \times H \times W}$, where C is the number of channels and H and W denotes the height

and width of each frame, respectively, our goal is to learn a mapping function M that transforms the video into a piece of music, represented as $\mathbf{x}_m \in \mathbb{R}^{T_m}$. Here, T_v and T_m represents the number of frames for the video and the music, respectively. The video and music have the same length of duration, while they have different frame rates, with $T_v = t \cdot f_v$ and $T_m = t \cdot f_m$, where t is the duration, f_v is the frame rate for the video, and f_m is the frame rate of the music. The mapping function is formally presented as $M : \mathbb{R}^{T_v \times C \times H \times W} \to \mathbb{R}^{T_m}$.

3.2 Autoregressive Music Generation

We focus on generating the music tokens autoregressively. Here music generation models focus on using a convolutional autoencoder to generate quantized music codes [1], followed by an autoregressive decoder that generates new music tokens. The generation of the tokens are conditioned on either textual descriptions or a sequence of provided tokens. Hence, we separate the music generation process into subsequet phases: 1) Music Tokenization, and 2) Autoregressive Generation.

3.2.1 Music Tokenization. In this work, we utilize a convolutional autoencoder, specifically EnCodec [10], where the latent representation is processed through Residual Vector Quantization (RVQ) [58]. For an input music signal $\mathbf{x}_m \in \mathbb{R}^{t \cdot f_m}$, sampled at a high frequency, EnCodec transforms it into a compact latent vector with a reduced frame rate f'_m, where $f'_m \ll f_m$. This latent vector is subsequently quantized into a discrete set $Q \in \{1, \ldots, M\}^{t \cdot f'm \times K}$, with K denoting the number of codebooks and M indicating the codebook size in

---

[1]In language modeling, codes are generally defined as tokens, serving as discrete representations.

the RVQ framework. Consequently, the i-th music token is derived as $\mathbf{z}_i = \sum_{k=1}^{K} \mathbf{z}_{i;k}$, aggregating contributions across all codebooks. Leveraging the residual tokens being summed per timestep in RVQ, we extend it by incorporating visual dynamics as additional tokens. Details are provided in Section 4.1.1.

3.2.2 Autoregressive Generation. The autoregressive music generator models the conditional probability distribution of the next music token given the preceding tokens, formally represented as $p(\tilde{Q}_i \mid \tilde{Q}_{i-1}, \dots, \tilde{Q}_0)$, where $\tilde{Q}_0$ is initialized as 0 and $i > 0$. To model this distribution, we adopt a music decoder $\mathcal{D}_m$, a single-stage language model, for music generation. To preserve musical harmony and consistency, we take advantage of the pre-tained MusicGen [8] and use it to initialize our model, instead of training the generation model from scratch.

4 OUR APPROACH

We introduce DyViM to enhance dynamics modeling in video-to-music generation, as illustrated in Figure 2. DyViM consists of two main modules: frame-wise dynamics feature encoding and token-level dynamics decoding. In addition, we integrate semantics features via a cross-attention within the decoder layers, and we employ an annealing tuning strategy to optimize the overall framework.

4.1 Frame-wise Dynamics Encoding

We aim to enhance the frame-wise dynamics by designing a dynamics encoder inspired by an optical-flow method. A self-attention module is utilized to aggregate features within frames to produce dense dynamic features. Additionally, we follow the previous methods to extract semantics features from pre-trained image encoder [48], to complements the dynamics features.

4.1.1 Dynamics Encoder. We encode dynamic features on a frame-wise basis to capture extensive visual motions across diverse video scenarios, including both dance and music videos. Inspired by previous works [39, 45, 50] that estimate optical flow by comparing neighboring frames, we incorporate a dynamic encoder to capture the dynamic details across frames, denoted as:

$$\mathbf{z}_d^i = \mathrm{DE}(\mathbf{X}_{v;i-1,i,i+1}), \ \mathrm{DE} := \mathcal{E}_{\mathrm{motion}} \circ \mathrm{Self\text{-}Attn}, \qquad (1)$$

where $\mathbf{X}_{v;i-1,i,i+1} \in \mathbb{R}^{3 \times C \times H \times W}$ represents the triplet of frames centered on the i-th frame. This setup enables us to exploit bi-directional dynamics in both forward and backward directions, following [45]. Specifically, we elaborate on dynamics decoding with three steps.

Encoding motions from frame triplets. We calculate the correlation volumes ($\mathbf{Corr}_{i,i-1}, \mathbf{Corr}_{i,i+1}$) to measure pixel-wise visual similarity between image pairs via dot-product in downsized height H' and width W'. Correlation features $\mathbf{F}_{corr}^l \in \mathbb{R}^{H' \times W' \times d_{corr}}$ and flow features $\mathbf{F}_{flow}^l \in \mathbb{R}^{H' \times W' \times d_{flow}}$ are encoded from the retrieved multi-scale correlation values and predicted bi-directional flows in l-th refinement iteration step, respectively. A motion encoder is then implemented to generate motion features $\mathbf{z}_m^i$ through a fusion module that combines both correlation and flow information, formulated as $f : \mathbb{R}^{H \times W \times d_c} \times \mathbb{R}^{H \times W \times d_f} \to \mathbb{R}^{H \times W \times d_m}$.

Extracting music relevance with attentive aggregation. While maintaining extensive motion details, our goal is to extract concise features that focus on music-relevant information. Therefore, we aggregate the generated motion features via a self-attention mechanism within frames, followed by average pooling to obtain the attentive dynamics $\mathbf{z}_d^i \in \mathbb{R}^{d_m}$ for each frame.

Organizing. Ultimately, the dynamic features are represented as $\mathbf{z}_d \in \mathbb{R}^{t \cdot f_v \times d_m} := [\mathbf{z}_d^0, \dots, \mathbf{z}_d^{t \cdot f_v}]$.

4.1.2 Semantics Encoder. Additionally, we introduce the semantics, which can be attributed to the visual content extracted from keyframes. A straightforward approach to capture the semantics from video is to utilize a pre-trained video captioning model, which summarizes the video content into textual descriptions []. However, such methods often ignore the visual nuances when translating visual features into text. To extract the semantic features from video, we leverage a pre-trained 2D visual encoder $\mathrm{SE}(\cdot)$, such as CLIP [41], to encode visual content from keyframes. Specifically, we first employ the MPEG-4 [16] compression technique to extract keyframes, where I-frames in MPEG-4 are used [23]. These frames are denoted as $\mathbf{X}_v^S \in \mathbb{R}^{N_s \times C \times H \times W}$, where N_s represents the number of keyframes. Subsequently, we employ a semantic encoder to obtain semantic features, denoted as:

$$\mathbf{z}_s = \mathrm{SE}(\mathbf{X}_v^S), \ \mathrm{SE} := \mathcal{E}_{\mathrm{image}} \circ \mathrm{Avg\text{-}Pool}, \qquad (2)$$

where $\mathcal{E}_{\mathrm{image}} : \mathbb{R}^{N_s \times C \times H \times W} \to \mathbb{R}^{N_s \times N_h \times d_s}$ transforms the N_s images into d_s-dimensional latent features, with N_h being the number of hidden states. We then apply a simple average pooling operation, $\mathrm{Avg\text{-}Pool} : \mathbb{R}^{N_s \times N_h \times d_s} \to \mathbb{R}^{N_h \times d_s}$, to efficiently aggregate these features.

4.2 Token-level Dynamics Decoding

To effectively generate music from nuanced visual features, we interpolate dynamics features to extend music tokens, achieving fine-grained temporal synchronized conditioning. And semantics are integrated via cross-attention modules.

4.2.1 Music Token Extension. Visual dynamics offer detailed conditions for music generation, introducing nuanced variations that facilitate rhythmic and temporal synchronization, providing low-level control. However, recent methods either apply visual dynamics as a global control [57, 63] or utilize direct matching (*e.g.*, onsets [34] or optical flow magnitude [24]) for strict synchronization, which limits its fine-grained and flexible conditioning. Inspired by the intrinsic structure of music codes, where four codes represent a single music token, where each code complements the prior ones, as outlined in Section 3.2—we propose integrating dynamic shifting into these existing tokens with the following two alignments.

Frame-level alignment. We first interpolate the dynamics features to match the number of music frames, thus yielding $\mathbf{z}_d' \in \mathbb{R}^{t \cdot f_m \times d_m}$. We adopt a fast and straightforward way of nearest-neighbor interpolation to find the closest dynamics for new points, where $\tilde{\mathbf{z}}_{d;i'}' = \tilde{\mathbf{z}}_{d;i}, i' = \mathrm{round}(t \cdot \frac{f_m}{f_v} \cdot i)$.

Dimensional alignment. We transform these features to align with the dimensionality of the music features using a Multilayer Perceptron (MLP), producing $\tilde{\mathbf{z}}_d \in \mathbb{R}^{t \cdot f_m \times d} = \mathrm{MLP}(\mathbf{z}_d')$. To incorporate the transformed dynamic features, we extend the existing

music codes derived from RVQ codebooks with interpolated dynamic features and sum them up to obtain the music tokens, defined as:

$$\mathbf{z}_i := \text{sum}(\alpha \cdot \tilde{\mathbf{z}}_{d;i}, \{(1 - \alpha) \cdot \mathbf{z}_{i;k}; k \in [K]\}), \quad (3)$$

where $\tilde{\mathbf{z}}_{d;i}$ represents the dynamic feature, and $\{\mathbf{z}_{i;k}\}$ represents the set of music codes from the codebooks, with k indexing over the set $[K]$; and α controls the strength of dynamics to the music codes.

Ultimately, music codes extension enables fine-grained control over music generation by embedding dynamic visual features, allowing the generated music to reflect subtle visual changes accurately.

4.2.2 Semantics Condition. Visual semantics provide fundamental guidance for music generation, maintained throughout the entire process. To this end, we propose incorporating visual semantics as high-level signals, which are temporally invariant and control the foundational tones of the music. Similarly, we adopt an MLP projector, to transform visual semantics into music conditions, denoted as $\tilde{\mathbf{z}}_s \in \mathbb{R}^{N_d \times d} = \text{MLP}(\mathbf{z}_s)$, thus maintaining dimensional consistency [35, 62]. To condition the music generation, we employ a cross-attention mechanism, denoted as:

$$\mathbf{A}^n = \frac{1}{\sqrt{d}} \mathbf{z}^{n-1} \mathbf{W}_{\mathbf{q}}^{\mathbf{n}} (\tilde{\mathbf{z}}_s \mathbf{W}_{\mathbf{k}}^{\mathbf{n}})^\top,$$
$$\mathbf{z}^n = \text{softmax}(\mathbf{A}^n) \tilde{\mathbf{z}}_s \mathbf{W}_{\mathbf{v}}^{\mathbf{n}}, \quad (4)$$

where $\mathbf{W}_{\mathbf{q}}^{\mathbf{n}}$, $\mathbf{W}_{\mathbf{k}}^{\mathbf{n}}$, and $\mathbf{W}_{\mathbf{v}}^{\mathbf{n}} \in \mathbb{R}^{d \times d}$ are attention weight matrices that transform the music features $\mathbf{z}^{n-1}$ and semantic conditions into query, key, and value spaces at the n-th layer. $\mathbf{A}^n$ is the corresponding attention matrix, followed by a softmax function to normalize attention scores, resulting in new representations at the next layer.

4.3 Annealing Tuning

To leverage the effectiveness of pre-trained music generation models while preserving their inherent musical knowledge, we employ an annealing fine-tuning strategy. Specifically, we fine-tune the music decoder with only a small set of parameters by adopting LoRA [19] with an annealing schedule. Following an autoregressive approach, we define the annealing tuning formally as:

$$\max_{\Theta} \sum_{\tau=1}^{|Q|} \log \mathbf{a}_\tau p(\tilde{Q}_\tau | Q_{<\tau}; \mathbf{z}_s, \mathbf{z}_d), \quad (5)$$

where Θ represents all trainable parameters, including the LoRA weights and modules from the decomposition and composition processes. The annealing schedule $\mathbf{a}_\tau \in \mathbb{R}^{t \cdot f_m}$ controls the weight of each token, with more emphasis placed on the initial tokens, and gradually reducing the weight over time. The rationale is that the musical theme or structure is primarily established during the critical early stages of music generation. Reducing the weight on later tokens allows the model to rely on its well-trained understanding of musical structure, enabling coherent music generation without being overly constrained by the given samples. We employ a cosine decay schedule, defined as $\mathbf{a}_\tau = a_{\max} \cdot (1 + \cos(\frac{\pi \cdot \tau}{|Q|}))/2 + \epsilon$, where $a_{\max}$ and ϵ denote the maximum and minimum weight values, respectively. Moreover, we analyze various schedules, like linear or step decays in Section 6.3.5 for a comprehensive evaluation.

5 EXPERIMENTAL SETUP

We elucidate the experimental setup, covering implementation details, datasets, metrics, and selected baselines to ensure a fair and comprehensive evaluation.

5.1 Implementation Details

5.1.1 Visual Encoder. We adapt the official VideoFlow library [2] and extract the motion features (d_m = 1024) for dynamic feature aggregation, and employ the official CLIP library [3] and extract the last hidden states (N_h = 50 and d_s = 768) for semantic feature aggregation.

5.1.2 Music Decoder. We adopt MusicGen [8] as the music decoder, and EnCodec [10] as the compression model that tokenize waveforms, where the dimension of music features is 1536, *i.e.*, d=1536. Their official library [4] is adapted for further modification. MusicGen is equipped with 48 transformer layers, followed by 4 linear layer to map music features to the indexes of music codes. While EnCodec is a convolutional neural network, compresses 1-second waveform to 50 discrete audio tokens within four codebooks with the size of 2048.

5.1.3 Hyper-parameter Setting. We split the video in to 10-second clips to form the datasets. And we utilize the AdamW optimizer (β_1=0.9 and β_2=0.95) with a batch size of 8 and warming up for 100 steps in a cosine learning schedule. Notably, our training is efficient with a single 48GB A40 GPU device.

5.2 Datasets

We follow the prior video-to-music methods [11, 32, 57, 63], while evaluating on more diverse datasets to evaluate the efficacy of proposed methods. Specifically, the collected dataset incorporates two categories: dance and music videos. The dance videos are curated from AIST++ [30], with clear visual pacing accompanies rhythmic songs. The music videos refers to the a short film, mostly focusing on the the content to visual storytelling. We collect this type of videos from SymMv [65] and BGM909 [5] [32]. And we split the dataset for training and testing with the ratio of 9:1 to ensure fairness and avoids the information leakage.

5.3 Evaluation Metrics

For a comprehensive evaluation, we utilize both objective and subjective metrics to evaluate the overall performance.

5.3.1 Objective Evaluation. contains the estimations between generated music to the real music in associated video. Following previous work [21, 48, 51], we compute the Fréchet Distance (FD), Fréchet Audio Distance (FAD) [25] and Kullback-Leibler Divergence (KL) [29]. Wherein, FAD compares the statistical distribution between generated and real music via their extracted high-level feature from VGGish [18], while FD uses PANNs [27] as the feature

---

[2]https://github.com/XiaoyuShi97/VideoFlow
[3]https://github.com/openai/CLIP
[4]https://github.com/facebookresearch/audiocraft
[5]The BGM909 dataset changes the original music tracks of the videos using midi music retrieved from POP909. In our case, we choose to keep the original version to maintain fidelity.

Table 1: The overall performance comparison between our DyViM and various baselines on three datasets regarding both subjective (Obj. Eval.) and objective (Subj. Eval.) evaluation metrics. Bold and underlined indicate the best and the second-best performance.

Dataset	AIST++					SymMV					BGM909				
	Obj. Eval.			Subj. Eval.		Obj. Eval.			Subj. Eval.		Obj. Eval.			Subj. Eval.	
Metrics	FD↓	FAD↓	KL↓	OVL↑	REL↑	FD↓	FAD↓	KL↓	OVL↑	REL↑	FD↓	FAD↓	KL↓	OVL↑	REL↑
CMT [11]	63.65	15.59	1.24	62.05	49.23	54.86	7.13	<u>0.87</u>	53.91	42.61	51.09	8.56	1.02	45.21	30.91
D2MGAN [8] [63]	48.37	11.40	0.94	46.67	58.33	-	-	-	-	-	-	-	-	-	-
CDCD [8] [64]	54.80	7.20	**0.40**	50.62	<u>76.25</u>	-	-	-	-	-	-	-	-	-	-
Video2Music [24]	90.15	31.53	1.54	64.10	43.59	81.06	20.70	1.03	60.87	<u>60.21</u>	86.25	24.84	1.29	61.73	51.30
DiffBGM [32]	88.12	23.72	1.53	<u>66.40</u>	37.6	71.44	16.23	0.93	<u>64.37</u>	47.50	75.63	9.90	1.22	<u>61.87</u>	44.37
M^{2}UGen [21]	62.89	18.39	0.98	26.67	36.41	<u>47.91</u>	5.49	1.07	56.52	43.64	45.70	5.75	1.23	49.56	44.34
VidMuse [51]	<u>46.50</u>	<u>6.03</u>	0.99	58.46	53.84	50.19	<u>4.78</u>	1.03	60.21	50.43	<u>32.41</u>	**2.80**	**0.93**	60.86	<u>61.73</u>
DyViM (ours)	**26.86**	**4.11**	<u>0.75</u>	**74.35**	**77.43**	**31.99**	**3.24**	**0.77**	**71.30**	**76.52**	**24.19**	<u>3.40</u>	<u>0.94</u>	**62.61**	**63.48**

extractor. And KL compute the divergence of labels between generated and original music. These evaluation can be implemented via well-established libraries [6].

5.3.2 Subjective Evaluation. Inspired by recent text/image-to-music methods [7, 28], we assess the generated samples using two metrics: overall quality (OVL) and relevance to the input video (REL), each rated on a range from 1 to 10 [7].

5.4 Baselines

To make an exhaustive evaluation, we choose several well performed and accessible methods as baselines. **CMT** [11] pre-defines visual features to connect video to music via motion and timing and generate symbolic music. **D2MGAN** [63] is specialized for dance videos with a GAN-based discriminator for generating waveforms with human body motions. **CDCD** [64] advances D2MGAN by combining diffusion and constrative learning for dance-to-music generation. **Video2Music** [24] leverages semantic, motion and emotional features to predict symbolic music by training a specialized transformer decoder. **DiffBGM** [32] segments input video for frames and captions, followed with visual and language encoders and output piano roll conditioned with a diffusion model. **M^{2}UGen** [21] employs a LLM to understand multiple modalities (*e.g.*, video and text) and generates waveform music via a frozen pretrained music decoder. **VidMuse** [51] utilizes a long-short-term visual module to obtain visual features and trains a pre-trained music decoder with extensive data. Notably, we implement these baselines rigidly following their officially provided code to ensure their promised performance.

6 RESULTS AND ANALYSIS

To demonstrate the effectiveness and rationale of DyViM, we conduct extensive experiments, including an overall comparison with baselines, ablation study on conditioning and tuning strategies, and model analyses on dynamics control, visual feature selection, and annealing schedules.

6.1 Overall Performance

To demonstrate the effectiveness of DyViM, we present both objective and subjective comparisons across three datasets, as shown in Table 1. DyViM consistently outperforms the baselines, indicating its superiority in generating both high-fidelity and visually synchronized music. Specifically, we have multiple observations. First, DyViM significantly improves upon the baselines, even surpassing specialized models on certain metrics, such as D2MGAN and CDCD for dance-to-music generation, and VidMuse, which was trained on 200K music videos. Second, symbolic music generation yields lower objective scores due to substantial distributional gaps with test datasets, but provides a relatively comfortable listening experience in subjective evaluations due to its generation policy based on symbolics, as exemplified by Video2Music and Diff-BGM. Third, both M^{2}UGen and VidMuse leverage pre-trained music decoders (*e.g.*, MusicGen), leading to improvements in objective scores. Finally, the paradigm of freezing the music decoder (*i.e.*, M^{2}UGen) causes disharmony, significantly disrupting the listening experience and leading to lower subjective scores. Both objective and human evaluations compared with baselines demonstrate the effectiveness of DyViM.

6.2 Ablation Study

6.2.1 Impact of different conditions. To verify the importance of different conditions (*i.e.*, dynamics, denoted as **D**, and semantics, denoted as **S**), we perform ablations to create different model variants, as shown in Table 2. The results showcases a significant performance drop when dynamics are excluded, compared to the case without semantics. This might indicate that visual dynamics provide more informative guidance than high-level semantics in the music generation process.
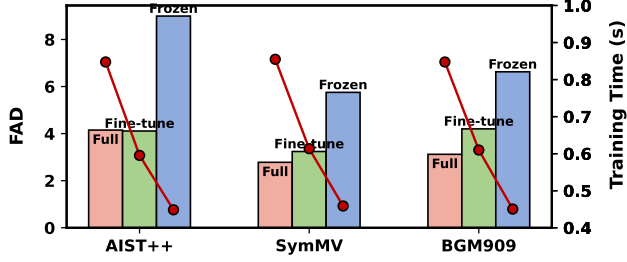
6.2.2 Impact of different condition methods. To explore various methods for conditioning music generation from video, we also replace the conditioning methods in the two-level composition process as shown in Table 2. Specifically, we test: 1) cross, which uses a cross-attention mechanism; 2) prepend, which adds the features as prefix tokens; and 3) extend, representing our proposed method of extending music codes. Among these conditioning methods, our

---

[6]https://github.com/haoheliu/audioldm_eval
[7]The reported scores are scaled up to 100 for intuitive presentation.

Table 2: Performance comparison *w.r.t.* different visual conditions and strategies.

Dataset	Visual conditions		Condition strategies			
	w/o-D	w/o-S	$\mathbf{D}_{cross}$	$\mathbf{D}^*_{extend}$	$\mathbf{S}_{prepend}$	$\mathbf{S}^*_{cross}$
AIST++	4.63	4.59	4.41	4.11	4.45	4.11
SymMV	4.02	3.67	3.57	3.24	4.16	3.24
BGM909	4.39	4.29	4.26	3.40	4.49	3.40

Figure 3: Performance comparison (bar) and the time cost during training (line) *w.r.t.* different tuning strategies.
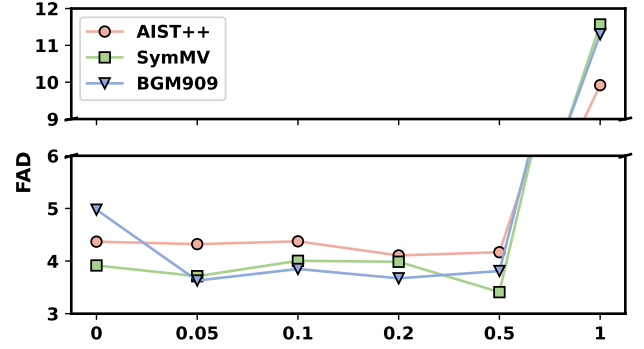
proposed approach of extending music codes combined with semantic conditioning via cross-attention achieves the best performance. This result further supports the rationale behind our design, demonstrating its effectiveness in adapting to different levels of visual cues.

6.2.3 Performance regarding to different tuning strategies. We conduct performance and training time comparisons across different tuning strategies, as shown in Figure 3, to demonstrate the effectiveness and efficiency of our fine-tuning strategy. Wherein, **Full** denotes updating all parameters, **Fine-tune** represents our approach that trains only a subset of parameters, and **Frozen** keeps all parameters fixed. The results reveal a clear performance hierarchy: **Full ≥ Fine-tune > Frozen**, and an efficiency hierarchy: **Full < Fine-tune < Frozen**. The fine-tuning approach achieves an effective trade-off between performance and efficiency. Notably, the full tuning setting incorporates our annealing schedule, which helps mitigate overfitting, leading to enhanced performance.

6.3 Model Study

6.3.1 Impact of α to control dynamics. The parameter α, introduced in Equation 3, represents the strength of the dynamic feature when extending the music tokens. To illustrate its impact, we vary its value within the range [0, 1] and present a performance comparison in Figure 4. Incorporating visual dynamics generally results in lower FAD scores (*i.e.*, improved performance), while the extreme cases of $\alpha = 0$ and $\alpha = 1$ yield poorer results. In our experiments, a larger value of α tends to destabilize the learning process and is prone to convergence to a suboptimal state. Therefore, we emphasize tuning α below 0.5.

---

[8]D2MGAN and CDCD specialize in dance videos with only processed features provided, so we report results on their self-split AIST++ dataset.

Figure 4: Performance comparison *w.r.t.* different α to control the strength of visual dynamics.

Table 3: Performance comparison *w.r.t.* different visual encoders.

Dataset	Dynamic encoder			Semantic encoder	
	$DE^{OpenPose9}$	DE^{I3D}	DE^{FwDF*}	SE^{T5}	SE^{CLIP*}
AIST++	4.48	4.23	4.11	4.09	4.11
SymMV	-	5.31	3.24	4.29	3.24
BGM909	-	3.85	3.40	4.28	3.40

6.3.2 Impact of different visual encoders. We replace different visual encoders for both dynamic and semantic encoding to investigate their impacts, with performance comparisons shown in Table 3. For dynamics, we compare our proposed method (*i.e.*, $\mathbf{D}^{FwDF}$) with OpenPose [4], which is specifically used for dance videos [57, 63, 64], and I3D [5], which also leverages optical flow information but with frame compression. Compared to other dynamic encoders, our method demonstrates superior performance, which can be attributed to its capability to capture nuanced visual information and attentively guide music generation. For semantics, we employ a video captioning model (*i.e.*, PLLaVA [54]) to obtain descriptive captions, followed by a T5 model [42]. The transformation from video → language → music may introduce information loss, which limits its performance compared to a direct image encoder like CLIP [41], especially on datasets like SymMV and BGM909, which contain complex semantics. However, for simpler semantics, such as in AIST++, which consists mainly of dance scenes, T5-based approach slightly outperforms visual encoder methods.

6.3.3 Dynamics attention. We illustrate the attention matrix generated by the self-attention modules in the dynamics encoder, as shown in Figure 5. There are distinct attention patterns for different types of videos. For intuitive visualization, we select 10 frames. For dance videos, the dynamics attention is relatively focused, which is reasonable since dance videos typically feature a fixed object within the scene producing motions. In contrast, music videos, which exhibit more diverse dynamics, display varying attention patterns across frames. Our proposed simple self-attention module, trained
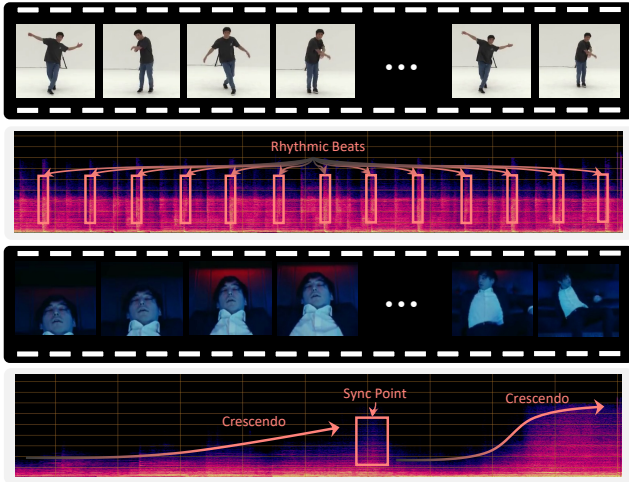
---

[9]OpenPose specializes in extracting human body motions, making it suitable only for the AIST++ dataset.

with token-level dynamics decoding, effectively captures these nuanced dynamic changes, thereby enhancing finer video-to-music generation.

Figure 5: An illustration of the dynamics attention matrix for dance videos (top) and music videos (bottom), where the size of each point represents the quantitative value of attention.

6.3.4 Case studies on visual dynamics. We analyze the dynamic visual correlations between the generated music and associated videos, as illustrated in Figure 6. Notably, DyViM produces visually relevant music that aligns with visual dynamics (*e.g.*, body motions corresponding to rhythmic beats and camera movements to music crescendos). More cases are available in the Demo and supplementary materials.

Figure 6: Case study on generated music samples from dance and music videos.

Table 4: Performance comparison *w.r.t.* different tuning schedules.

Dataset	w/o-annealing		w/-annealing		
	$a^{constant}$	a^{random}	a^{step}	a^{linear}	$a^{cosine*}$
AIST++	4.87	4.76	4.18	4.24	4.11
SymMV	4.11	4.43	3.39	3.32	3.24
BGM909	4.64	4.71	3.59	3.64	3.40

6.3.5 Different annealing schedules. Annealing tuning is central to adapting the autoregressive music decoder for video conditioning in DyViM. The main idea is to prioritize the initial tokens while gradually reducing the weights of later tokens. We examine five different annealing schedules in Table 4: 1) $a_\tau^{constant} = 1$; 2) $a_\tau^{random} \sim \text{Uniform}(\epsilon, a_{max})$; 3) $a_\tau^{step} = \mathbb{I}(\tau < |Q|/2) \cdot a_{max} + \epsilon$; 4) $a_\tau^{linear} = \frac{(|Q|-\tau) \cdot a_{max}}{|Q|} + \epsilon$; 5) $a_\tau^{cosine} = \frac{a_{max} \cdot (1 + \cos(\frac{\pi \cdot \tau}{|Q|}))}{2} + \epsilon$. We consider $a^{constant}$ and a^{random} as non-annealing tuning variants, with $a^{constant}$ representing the most conventional fine-tuning method. The results clearly showcase that incorporating an annealing schedule significantly improves performance, further validating the rationale and effectiveness of our approach.

7 CONCLUSION

In this paper, we emphasized the significance of visual dynamics for video-to-music, while exhibiting an unexplored gap in leveraging nuanced dynamic cues for temporally synchronized music. To this end, we presented DyViM, a novel framework that enhances dynamics modeling in video-to-music generation. Specifically, we encoded the frame-wise dynamics features by adapting optical-flow based methods to capture motion features across frame triples, followed with a self-attention to aggregate music-relevant dynamics within frames. These features are then decoded onto music tokens, adding dynamics shifts to achieve fine-grained temporal synchronization. Moreover, semantics are incorporated with cross-attention for high-level conditions. We proposed an annealing tuning strategies to facilitate the model optimization. We conducted extensive experiments, comparing DyViM with SOTA methods across three datasets and evaluating results through both objective and subjective metrics. We hope to push the boundaries of this field, especially by delving into the complex dynamics between video and music and providing capabilities that work with it out of the box. In future work, we will explore multiple meaningful directions, including: 1) Introducing comprehensive evaluation dimensions, including new metrics, novel benchmarks, etc., for holistic video-to-music assessment; 2) Handling long-sequence video-to-music generation that maintains temporal consistency while incorporating narrative progression and structural variation over extended durations; 3) Investigating interactive video-to-music generation systems that allow diverse forms of user instructions and customization.

REFERENCES

[1] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse H. Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Havnø Frank. 2023. MusicLM: Generating Music From Text. *CoRR* abs/2301.11325 (2023).

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. In *NeurIPS*.

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. 2021. ViViT: A Video Vision Transformer. In *ICCV*. IEEE, 6816–6826.

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *CVPR*. IEEE Computer Society, 1302–1310.

[5] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*. IEEE Computer Society, 4724–4733.

[6] Lele Chen, Sudhanshu Srivastava, Zhiyao Duan, and Chenliang Xu. 2017. Deep Cross-Modal Audio-Visual Generation. In *ACM Multimedia (Thematic Workshops)*. ACM, 349–357.

[7] Sanjoy Chowdhury, Sayan Nag, K. J. Joseph, Balaji Vasan Srinivasan, and Dinesh Manocha. 2024. MELFuSION: Synthesizing Music from Image and Language Cues Using Diffusion Models. In *CVPR*. IEEE, 26816–26825.

[8] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2023. Simple and Controllable Music Generation. In *NeurIPS*.

[9] Adyasha Dash and Kathleen Agres. 2024. AI-Based Affective Music Generation Systems: A Review of Methods and Challenges. *ACM Comput. Surv.* 56, 11 (2024), 287:1–287:34.

[10] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2023. High Fidelity Neural Audio Compression. *Trans. Mach. Learn. Res.* 2023 (2023).

[11] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. 2021. Video Background Music Generation with Controllable Music Transformer. In *MM*, Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo César, Florian Metze, and Balakrishnan Prabhakaran (Eds.). ACM, 2037–2045.

[12] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2018. MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In *AAAI*. AAAI Press, 34–41.

[13] Yuexi Du, Ziyang Chen, Justin Salamon, Bryan Russell, and Andrew Owens. 2023. Conditional Generation of Audio from Video via Foley Analogies. In *CVPR*. IEEE, 2426–2436.

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast Networks for Video Recognition. In *ICCV*. IEEE, 6201–6210.

[15] Seth* Forsgren and Hayk* Martiros. 2022. Riffusion - Stable diffusion for real-time music generation. (2022). https://riffusion.com/about

[16] Didier Le Gall. 1991. MPEG: A Video Compression Standard for Multimedia Applications. *Commun. ACM* 34, 4 (1991), 46–58.

[17] Chuang Gan, Deng Huang, Peihao Chen, Joshua B. Tenenbaum, and Antonio Torralba. 2020. Foley Music: Learning to Generate Music from Videos. In *ECCV (Lecture Notes in Computer Science, Vol. 12356)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 758–775.

[18] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. 2017. CNN architectures for large-scale audio classification. In *ICASSP*. IEEE, 131–135.

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[20] 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. 2019. Music Transformer: Generating Music with Long-Term Structure. In *ICLR (Poster)*. OpenReview.net.

[21] Atin Sakkeer Hussain, Shansong Liu, Chenshuo Sun, and Ying Shan. 2023. M^{2}UGen: Multi-modal Music Understanding and Generation with the Power of Large Language Models. *CoRR* abs/2311.11255 (2023).

[22] Shulei Ji, Jing Luo, and Xinyu Yang. 2020. A Comprehensive Survey on Deep Music Generation: Multi-level Representations, Algorithms, Evaluations, and Future Directions. *CoRR* abs/2011.06801 (2020).

[23] Yang Jin, Zhicheng Sun, Kun Xu, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, Kun Gai, and Yadong Mu. 2024. Video-LaVIT: Unified Video-Language Pre-training with Decoupled Visual-Motional Tokenization. In *ICML*.

[24] Jaeyong Kang, Soujanya Poria, and Dorien Herremans. 2024. Video2Music: Suitable music generation from videos using an Affective Multimodal Transformer model. *Expert Syst. Appl.* 249 (2024), 123640.

[25] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. 2019. Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms. In *INTERSPEECH*. ISCA, 2350–2354.

[26] A. Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman. 2020. Sight to Sound: An End-to-End Approach for Visual Piano Transcription. In *ICASSP*. IEEE, 1838–1842.

[27] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. *Trans. Audio Speech Lang. Process.* 28 (2020), 2880–2894.

[28] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2023. AudioGen: Textually Guided Audio Generation. In *ICLR*. OpenReview.net.

[29] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86. https://doi.org/10.1214/aoms/1177729694

[30] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. 2021. Learn to Dance with AIST++: Music Conditioned 3D Dance Generation. *CoRR* abs/2101.08779 (2021).

[31] Ruiqi Li, Siqi Zheng, Xize Cheng, Ziang Zhang, Shengpeng Ji, and Zhou Zhao. 2024. MuVi: Video-to-Music Generation with Semantic Alignment and Rhythmic Synchronization. arXiv:2410.12957 [cs.SD] https://arxiv.org/abs/2410.12957

[32] Sizhe Li, Yiming Qin, Minghang Zheng, Xin Jin, and Yang Liu. 2024. Diff-BGM: A Diffusion Model for Video Background Music Generation. *CVPR* (2024).

[33] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. SwinBERT: End-to-End Transformers with Sparse Attention for Video Captioning. In *CVPR*. IEEE, 17928–17937.

[34] Yan-Bo Lin, Yu Tian, Linjie Yang, Gedas Bertasius, and Heng Wang. 2024. VMAS: Video-to-Music Generation via Semantic Alignment in Web Music Videos. *CoRR* abs/2409.07450 (2024).

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. In *NeurIPS*.

[36] Hui Liu and Xiaojun Wan. 2021. Video Paragraph Captioning as a Text Summarization Task. In *ACL/IJCNLP (2)*. Association for Computational Linguistics, 55–60.

[37] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. 2022. Swin Transformer V2: Scaling Up Capacity and Resolution. In *CVPR*. IEEE, 11999–12009.

[38] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. Video Swin Transformer. In *CVPR*. IEEE, 3192–3201.

[39] Yawen Lu, Qifan Wang, Siqi Ma, Tong Geng, Yingjie Victor Chen, Huaijin G. Chen, and Dongfang Liu. 2023. TransFlow: Transformer as Flow Learner. In *CVPR*. IEEE, 18063–18073.

[40] Xingwei Qu, Yuelin Bai, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, Xinrun Du, Shuyue Guo, Yiming Liang, Yizhi Li, Shangda Wu, Junting Zhou, Tianyu Zheng, Ziyang Ma, Fengze Han, Wei Xue, Gus Xia, Emmanouil Benetos, Xiang Yue, Chenghua Lin, Xu Tan, Stephen W. Huang, Wenhu Chen, Jie Fu, and Ge Zhang. 2024. MuPT: A Generative Symbolic Music Pretrained Transformer. *CoRR* abs/2404.06393 (2024).

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, Marina Meila and Tong Zhang (Eds.), Vol. 139. PMLR, 8748–8763.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[43] Ivan Rinaldi, Nicola Fanelli, Giovanna Castellano, and Gennaro Vessio. 2024. Art2Mus: Bridging Visual Arts and Music through Cross-Modal Generation. *CoRR* abs/2410.04906 (2024).

[44] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. 2022. End-to-end Generative Pretraining for Multimodal Video Captioning. In *CVPR*. IEEE, 17938–17947.

[45] Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. 2023. VideoFlow: Exploiting Temporal Cues for Multi-frame Optical Flow Estimation. In *ICCV*. IEEE, 12435–12446.

[46] Jiajie Su, Chaochao Chen, Zibin Lin, Xi Li, Weiming Liu, and Xiaolin Zheng. 2023. Personalized Behavior-Aware Transformer for Multi-Behavior Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia* (Ottawa ON, Canada) *(MM '23)*. Association for Computing Machinery, New York, NY, USA, 6321–6331. https://doi.org/10.1145/3581783.3611723

[47] Jiajie Su, Chaochao Chen, Weiming Liu, Fei Wu, Xiaolin Zheng, and Haoming Lyu. 2023. Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation. In *Proceedings of the ACM web conference 2023*. 165–176.

[48] Kun Su, Judith Yue Li, Qingqing Huang, Dima Kuzmin, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, and Timo I. Denk. 2024. V2Meow: Meowing to the Visual Beat via Video-to-Music Generation. In *AAAI*, Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (Eds.). AAAI Press, 4952–4960.

[49] Kun Su, Xiulong Liu, and Eli Shlizerman. 2020. Audeo: Audio Generation for a Silent Performance Video. In *NeurIPS*.

[50] Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV (2) (Lecture Notes in Computer Science, Vol. 12347)*. Springer, 402–419.

[51] Zeyue Tian, Zhaoyang Liu, Ruibin Yuan, Jiahao Pan, Xiaoqiang Huang, Qifeng Liu, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. 2024. VidMuse: A Simple Video-to-Music Generation Framework with Long-Short-Term Modeling. *CoRR* abs/2406.04321 (2024).

[52] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*. IEEE Computer Society, 4489–4497.

[53] Paul Voigtlaender, Soravit Changpinyo, Jordi Pont-Tuset, Radu Soricut, and Vittorio Ferrari. 2023. Connecting Vision and Language with Video Localized Narratives. In *CVPR*. IEEE, 2461–2471.

[54] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. PLLaVA : Parameter-free LLaVA Extension from Images to Videos for Video Dense Captioning. arXiv:2404.16994 [cs.CV]

[55] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning. In *CVPR*. IEEE, 10714–10726.

[56] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2023. Diffsound: Discrete Diffusion Model for Text-to-Sound Generation. *IEEE ACM Trans. Audio Speech Lang. Process.* 31 (2023), 1720–1733.

[57] Jiashuo Yu, Yaohui Wang, Xinyuan Chen, Xiao Sun, and Yu Qiao. 2023. Long-Term Rhythmic Video Soundtracker. In *ICML (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara

Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 40339–40353.

[58] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2022. SoundStream: An End-to-End Neural Audio Codec. *IEEE ACM Trans. Audio Speech Lang. Process.* 30 (2022), 495–507.

[59] Yiming Zhang, Yicheng Gu, Yanhong Zeng, Zhening Xing, Yuancheng Wang, Zhizheng Wu, and Kai Chen. 2024. FoleyCrafter: Bring Silent Videos to Life with Lifelike and Synchronized Sounds. *CoRR* abs/2407.01494 (2024).

[60] Zhenglin Zhou, Fan Ma, Hehe Fan, Zongxin Yang, and Yi Yang. 2024. Headstudio: Text to animatable head avatars with 3d gaussian splatting. In *European Conference on Computer Vision*. Springer, 145–163.

[61] Zhenglin Zhou, Xiaobo Xia, Fan Ma, Hehe Fan, Yi Yang, and Tat-Seng Chua. 2025. DreamDPO: Aligning Text-to-3D Generation with Human Preferences via Direct Preference Optimization. *arXiv preprint arXiv:2502.04370* (2025).

[62] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. In *ICLR*. OpenReview.net.

[63] Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. 2022. Quantized GAN for Complex Music Generation from Dance Videos. In *ECCV (Lecture Notes in Computer Science, Vol. 13697)*, Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, 182–199.

[64] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. 2023. Discrete Contrastive Diffusion for Cross-Modal Music and Image Generation. In *ICLR*. OpenReview.net.

[65] Le Zhuo, Zhaokai Wang, Baisen Wang, Yue Liao, Chenxi Bao, Stanley Peng, Songhao Han, Aixi Zhang, Fei Fang, and Si Liu. 2023. Video Background Music Generation: Dataset, Method and Evaluation. In *ICCV*. IEEE, 15591–15601.

A DATA PRE-PROCESSING DETAILS

We summarize the statistics of the datasets in Table 5. The datasets were downloaded from their official repositories: AIST++[8], SymMV[9], and Diff-BGM[10]. To preserve the original video-music correlations, we retain the waveform tracks provided in the datasets rather than adopting symbolic music (*e.g.*, MIDI) or substituted tracks (*e.g.*, POP909 in Diff-BGM).

Table 5: Statistics of datasets.

	Video Content	Size	Length (Hours)
AIST++	Dance Video	1,408	5.2
SymMV	Music Video	1,140	76.5
BGM909	Music Video	909	62.6

For training consistency and generation quality, we apply the following pre-processing steps: 1) Vocal Removal: Using vocal-remover [11], we remove vocals to focus on instrumental tracks. 2) Silence Detection: Prolonged silences are detected and removed with pydub[12] to ensure consistent audio energy. These steps maintain the integrity of the datasets while facilitating training and generation quality.
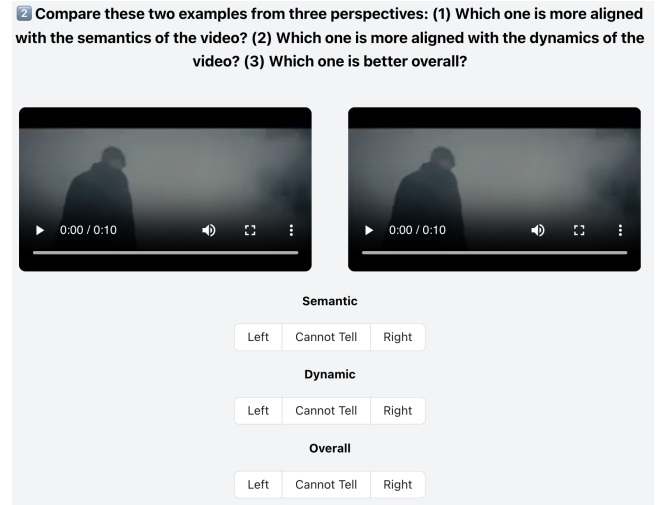
B IMPLEMENTATION DETAILS OF BASELINES

To ensure a rigorous and fair comparison, we carefully implement and adapt the baseline models following the instructions and publicly available codebases. **CMT** [11] and **Video2Music** [24] are models that generate symbolic music based on pre-defined visual features and rule-based broken chord approach, respectively, both using official implementations and pre-trained weights[13][14]. **Diff-BGM** [32][3] uses diffusion-based generation with visual and language encoders, relied on multiple open-source feature extractors: VideoCLIP[15] for video features, BLIP[16] for video captions, Bert-base-uncased[17] as the language encoder, and TransNetV2[18] for shot detection. We integrated these extractors with their default setting. Since the three aforementioned models produce symbolic music as outputs, we synthesized the results into audio using FluidSynth[19] and the FluidR3_GM soundfont[20]. Notably, DiffBGM's audio output was trimmed to match the video duration by retaining the initial segment. **D2MGAN** [63] and **CDCD** [64] are Dance2Music approaches heavily relying on OpenPose features, limiting their application to datasets like SymMV and BGM909. Both of them are trained on AIST++ and provide only pre-segmented clips and extracted features without preprocessing scripts; we utilized their provided model weights and calculated scores only on

their test sets to avoid inconsistencies in dataset splits, using the 2-second segmentation for D2MGAN and 6-second segmentation for CDCD, as recommended in their repositories[21]. **M²UGen** [21][22] and **VidMuse** [51][23] are end-to-end models capable of generating raw music directly from video inputs. Although their official repositories did not provide batch inference scripts, they included Gradio-based[24] demo interfaces. We modified these demos to develop custom inference scripts to evaluate these models on our test datasets.

For **VMAS** [34][25], **LORIS** [57][26], **MuVi** [31][27], and **V2Meow** [48][28], which introduce innovative frameworks such as Transformer-based, diffusion-based, multi-stage autoregressive models, the absence of publicly available code or pre-trained weights precluded their inclusion in our experiments.

C HUMAN EVALUATION DETAILS

Figure 7: The human evaluation platform for comparing two generated samples. Participants can choose from three options (*i.e.*, left, cannot tell, and right) to express their evaluations.

To conduct the subjective evaluation for video-to-music generation, we designed a questionnaire that asks raters: "Rate the generated music from two perspectives: 1) Overall quality (OVL) and 2) Visual relevance (REL)." Notably, we randomly selected samples for raters to evaluate, collected their ratings, and then averaged the ratings at the end. Moreover, we compared DyViM with Vid-Muse (which is trained on extensive video-music pairs) by asking participants: "Compare these two examples from three perspectives: (1) Which one is more aligned with the semantics of the video?

---

[8]https://github.com/L-YeZhu/D2M-GAN
[9]https://github.com/zhuole1025/SymMV
[10]https://github.com/sizhelee/Diff-BGM
[11]https://github.com/tsurumeso/vocal-remover
[12]https://github.com/jiaaro/pydub
[13]https://github.com/CMT-repository
[14]https://github.com/Video2Music-repository
[15]https://github.com/CryhanFang/CLIP2Video
[16]https://github.com/salesforce/BLIP
[17]https://huggingface.co/google-bert/bert-base-uncased
[18]https://github.com/soCzech/TransNetV2
[19]https://www.fluidsynth.org/
[20]https://musical-artifacts.com/artifacts/738

[21]https://github.com/D2MGAN-repository, https://github.com/L-YeZhu/CDCD
[22]https://github.com/M2UGen-repository
[23]https://github.com/VidMuse-repository
[24]https://www.gradio.app/docs
[25]https://genjib.github.io/project_page/VMAS/index.html
[26]https://github.com/OpenGVLab/LORIS
[27]https://muvi-v2m.github.io
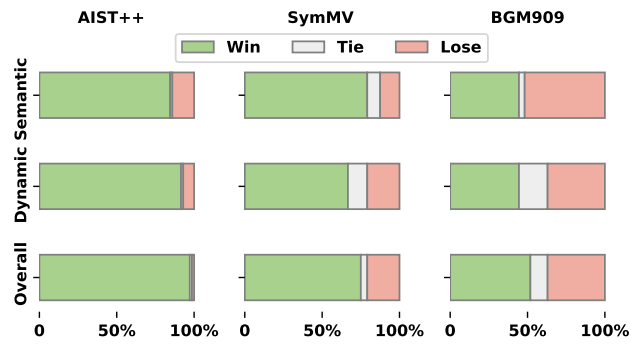[28]https://tinyurl.com/v2meow

Figure 8: Human voting by comparing music qualities between DyViM (green) and baseline VidMuse (red) from three perspectives: semantic alignment, dynamic alignment, and overall quality.

(2) Which one is more aligned with the dynamics of the video? (3) Which one is better overall?" This evaluation was conducted on the platform shown in Figure 7. The results are presented in Figure 8. In general, DyViM outperforms VidMuse, showcasing significant improvements, especially on the AIST++ and SymMV datasets.