TOWARDS FLEXIBLE AND CONTROLLABLE MUSIC GENERATION FOR UNIVERSAL VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

Video-to-Music (V2M) generation task creates music that matches given visual content to improve audience experience. However, current methods have limitations in editing flexibility, conditional controllability, and generalizability. To meet varied user needs for general video use cases, we first construct Universe-MV, a music-video dataset for general scenarios, using large-scale web videos. We then train CueMuse, a general-purpose V2M framework supporting multi-condition control and editing, on Universe-MV. The framework includes V-Beater, a visual rhythm extraction module, and a conditional music generation backbone. Our model detects visual rhythm frames at 12 fps and can achieve precise alignment between video and music. At the same time, CueMuse can take text and editable video rhythm timestamps as inputs to satisfy additional professional requirements. Both subjective and objective experiments suggest that CueMuse achieves competitive performance and generally outperforms prior models in music-video alignment, music generation quality, and musicality. See our demo at `https://anonymous.4open.science/r/demo-CueMuse`.

1 INSTRUCTION

In recent years, with the development of short-video creation and social media platforms, as well as the sharp surge in AI-generated videos Seedance et al. (2026); Shao & Li (2025), integrating suitable background music to infuse warmth and emotion into silent footage has become an urgent creative demand. However, adding music to videos is no an easy task: music creation has a high entry barrier, while relying on existing music might trigger copyright and consistency issues, and may also cause audience aesthetic fatigue. At the same time, some video creations require millisecond-level alignment Levitin et al. (2000); Eg et al. (2015) between music beats and visual rhythms to delight audiences, which is extremely time-consuming and labor-intensive Sun et al. (2023). Therefore, automatic Video-to-Music (V2M) generation holds significant application value across a wide range of video domains, such as films, advertisements, dance videos, edited clips, landscape videos and so on.

Despite its wide range of application scenarios, this also places high demands on V2M models. An ideal universal V2M model should be adaptable to a broad spectrum (or even all) of visual scenarios and creations, while meeting the creative requirements of diverse scenarios. Specifically, this involves three key requirements: 1) High-fidelity: The quality, musicality and aesthetics of generated music are good enough to satisfy audiences. 2) Semantic alignment: The generated music can accurately understand and conform to the theme, elements, emotions and imagery of the video. 3) Temporal synchronization: The music can adapt to the semantic changes of the video over time, and synchronize certain prominent elements in the video with musical beats for emphasis. It is worth noting that demands for these three aspects vary across different video categories and creators. A landscape-themed video may only require beautiful music, while dance videos place greater emphasis on rhythm alignment. In some extreme cases, the emotions conveyed by the video and the music can even be diametrically opposed. Therefore, the V2M model also requires a high degree of flexibility and controllability to meet the needs of different creators and scenarios.

Motivated by this observation, contrary to previous approaches that focused on in-domain V2M generation task Yu et al. (2023); Su et al. (2024); Tian et al. (2025); Tong et al. (2026), we leverage large-scale freely accessible internet video data to learn general representations for various types

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of videos, namely Universe-MV. Universe-MV features video creation of various styles, including films, vlogs, editing, dance and so on. On this basis, we developed a video-to-music generation framework called **CueMuse**. CueMuse focuses on the flexibility and controllability in V2M generation and integrates multiple optional control conditions, including text instructions and beat timestamps. Different from existing video-to-music generation methods that utilize video cut detection model to detect visual beats Tong et al. (2026); Ji et al. (2026), we further develop an efficient **V**isual **Beat** Align**er** (V-Beater), which can end-to-end identify prominent actions and shot transitions in videos at a fine-grained level (around 80ms) to capture the temporal cues required for beat alignment in music generation. To further improve music quality, we also modified the architecture of MusicGen with reference to LLaMA3 Grattafiori et al. (2024) and retrained it on a million of internal music data.

We further fine-tune MusicGen-LLaMA on our newly acquired Universe-MV dataset to obtain CueMuse, which generates video-consistent pure music in an end-to-end manner. Editable video rhythm information, together with music descriptions, serves as optional generation conditions to achieve precise customization of music generated for videos. In addition, we integrated existing evaluation data to construct Universe-Bench and introduced new objective metrics to more comprehensively assess the quality of video-to-music generation. Experimental results show that our method produces higher-quality music on Universe-Bench compared with previous video-based music generation approaches. Meanwhile, we further conducted human evaluations, and the results indicate that subjects prefer CueMuse for its high musical fidelity, musicality and visual-music relation. The main contributions of this work can be concluded as follows:

- We propose CueMuse, a flexible and highly-controllable versatile video-to-music generation model capable of producing high-quality music highly correlated with multi-style videos under various conditional constraints.
- We develop V-Beater to detect potential frames in videos that can be aligned with musical beats, so as to provide high-level requirements for the V2M task.
- We collect a large-scale video-music dataset Universe-MV and conduct extensive experiments on out-of-domain Universe-Bench to demonstrate the superiority of our model.

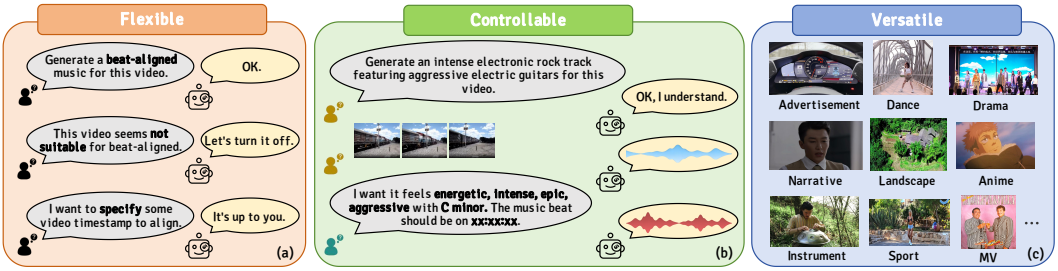

Figure 1: CueMuse shows advantages in flexibility, controllability and versatility compared with previous models.

2 Dataset

Previous video-to-music generation methods mostly adopt relatively small training datasets Zhuo et al. (2023); Li et al. (2024a); Zuo et al. (2025); Ji et al. (2026); Lokegaonkar et al. (2026); Tong et al. (2026) or single-category data Qi et al. (2024); Tian et al. (2025); Xie et al. (2025), which limits their generalization. Nevertheless, large-scale training data is the primary requirement for building high-performance video-to-music generation frameworks Lin et al. (2025). Some studies have constructed large-scale datasets yet failed to provide access to them Li et al. (2024a); Qi et al. (2024); Lin et al. (2025); Tong et al. (2026). Therefore, we decide to obtain the training data from scratch. To gather music-video data, VMAS filters data from Disco-10M Lanzendörfer et al. (2023) and Youtube-8M Abu-El-Haija et al. (2016). It ultimately produces roughly 2.2 million training pairs. However, music datasets with video labels such as Disco-10M Lanzendörfer et al. (2023) and LAION-DISCO-12M can acquire a substantial volume of data via music meta-information, yet they

Table 1: Statistics of common music-videos datasets, Class denotes the categories of major number of videos.

Music Class	Number	Duration (h)	Class
SymMv Zhuo et al. (2023)	1K	76.5	Universe
MUVideo Liu et al. (2023b)	13K	37	Universe
FilmScoreDB Qi et al. (2024)	32K	90	Film
MUVI Li et al. (2024a)	1.6K	280	Anime, Drama
MusicPro-7k Xie et al. (2025)	7k	-	Film
GVMGen Zuo et al. (2025)	-	147	MV, Documentary, Vlog
DISCO-MV Lin et al. (2025)	2.2M	47K	Universe
V2M Tian et al. (2025)	600K	16K	Film, Trailer, Documentary
harmonySet Zhou et al. (2025)	39K	340	Universe
TB-Match Tong et al. (2026)	18K	240	Advertisement
Universe-MV (Ours)	10K	7K	Universe

cannot guarantee the quality and categorization of data content. Furthermore, the music matching for these videos is not necessarily intended for creation. Given the resource overhead required for content filtering and categorization of such massive video data, we decided to directly crawl data from online video platforms. Our crawling approach is similar to Zuo et al. (2025). We retrieve data from free public platforms (Bilibili and Youtube) using keywords related to music creation, and assign higher weights to videos with high view counts and numerous comments. We then filter the rough data through a sophisticated pipeline to obtain suitable pairs for training V-Beater and CueMuse. In Table 1, we report the number, duration and category of the existing datasets related to V2M tasks. Our Universe-MV maintains a balanced and comprehensive range of music-video categories while boasting sufficient duration. For further details about our dataset and construction process, please refer to the Section D and Section E.

3 METHOD

In this section, we first define the video-to-music generation problem, then present CueMuse model in details, together with its main components and multi-stage training objectives.

3.1 PROBLEM DEFINITION

Given an input video $V = \{v_t\}_{t=1}^{N_f}$, where each frame $v_t \in \mathbb{R}^{C\times H\times W}$, the total number of video frames is

$$N_f = T \times f,$$

where T denotes the video duration and f denotes its original frame rate. Let the corresponding music be a waveform

$$Y = \{y_n\}_{n=1}^{N_a}, \qquad N_a = T \times r,$$

where r denotes the audio sampling rate. For example, a T-second audio segment sampled at $32\,\mathrm{kHz}$ contains $T \times r$ waveform samples, resulting in a substantially longer sequence than the video frame sequence. To obtain a compact representation, we employ an audio codec Défossez et al. (2022) to transform the raw waveform into a sequence of discrete acoustic tokens:

$$Y' = \mathrm{Codec}(Y) \in \mathcal{V}^{N_c\times K}, \qquad N_c = T \times r_c,$$

where r_c denotes the temporal token rate of the codec, K denotes the number of codebooks, and $\mathcal{V}$ is the discrete codebook vocabulary. Since $r_c \ll r$, Y' is substantially more compact than the raw waveform. The video-to-music (V2M) generation task is formulated as learning a conditional generative model

$$p_\theta(Y' \mid V, c),$$

where c denotes optional control signals. In our setting,

$$c \in \{\varnothing, I, B, (I, B)\},$$

where I denotes a natural-language instruction and

$$B = \{b_t\}_{t=1}^{N_f}, \qquad b_t \in \{0, 1\},$$

denotes the frame-level visual rhythm condition aligned with the video at f fps. When $c = \varnothing$, the model performs video-conditioned music generation without additional control signals. Otherwise, the model generates music under the corresponding instruction and/or rhythm constraints. Therefore, the overall training objective is

$$\boxed{\max_{\theta} \log p_\theta(Y' \mid V, c)}.$$

3.2 Visual Rhythm Extraction

3.2.1 Visual Encoder

Before utilizing the module we propose, we first need to encode the visual information from raw video inputs V. Specifically, we employ three visual encoders: DINOv3 Siméoni et al. (2025), Siglip2 Tschannen et al. (2025) and InternVideo2-S14 Wang et al. (2024) to extract diverse visual features. DINOv3 is trained via self-supervised learning and exhibits strong perception of visual edges and object categories. In contrast, Siglip2 is trained using contrastive learning and excels at capturing high-level visual semantics of images. InternVideo2-S14 is trained on extensive video data and is used to extract semantic information of global videos. To obtain visual rhythm and motion features, we employ DINOv3 with a uniform high frame rate (12-fps) to extract dense visual features F_{den} at a resolution of 32*32 and semantic feature F_{sem} extraction with Siglip2.

3.2.2 Visual Rhythm Acquirement

Well recognized visual rhythm points consist of two parts: cut-transition moments and timestamps of drastic intra-shot motion changes. In Section B.3, we summarize the efforts of V2M works toward achieving music-visual rhythm alignment. Cut timestamps can be annotated by models such as Soucek & Lokoc (2024), yet measuring motion variations within a video clip and deriving peak moments therefrom is far more challenging. Optical flow Beauchemin & Barron (1995); Shi et al. (2023) stands out among other features and RGB pixels for its capability to decouple motion independently. Prior work Lin et al. (2025) assumes that obtaining peak moments of optical flow is sufficient to represent visual rhythm, but this conclusion is one-sided. Davis & Agrawala (2018); Sun et al. (2023) suggests that visual rhythm should be defined based on local peaks of optical-flow acceleration (i.e., the second-order derivative of optical flow). We argue that humans perceive the sense of impact from the rate of change of motion (acceleration), rather than the magnitude of motion itself (velocity). Acceleration is arguably more consistent with the creative requirements of domains such as dance and montage. In the early stage of the research, we verified this viewpoint by quantitatively analyzed a small collection of dance videos as preliminary evidence.

To obtain visual rhythm, we first filter and then annotate approximately 2.3K hours of valid music-video data in 4 representative classes in Universe-MV using TransnetV2 Soucek & Lokoc (2024) and Videoflow Shi et al. (2023). See Table 7 for detail. Section F details the formula for extracting visual rhythm from visuals. To ensure there is no conflict between the two, we annotate optical flow within the scene annotations (i.e., video clips) and locate the peaks of the visual envelope. After obtaining the scene transitions and visual rhythms, we use their timestamps as supervision to train an end-to-end visual rhythm detection model, V-Beater.

3.2.3 Visual Beat Aligner

To train a visual prosody detector that predicts salient moments in visual content, we first need to set the frame rate for video input. A frame rate that is too low introduces inherent errors and misaligns predictions with actual video footage. An overly high frame rate wastes resources and creates GPU-memory bottlenecks. Finally, we decided to adopt an input frame rate of 12-fps to achieve a balance. As Figure 2 shows, given a batch of 12-fps dense visual features $F_{den} \in \mathbb{R}^{B\times T_h\times 32\times 32\times D_{dino}}$ and semantic feature $F_{sem} \in \mathbb{R}^{B\times T_h\times D_{siglip2}}$. To eliminate redundant visual information, V-Beater first uses 3×3 convolutions and bilinear uniformly to downsample the resolution of F_{den} to 24×24, and maps the dimension to $D_{hidden} = 512$ via a projector. Then

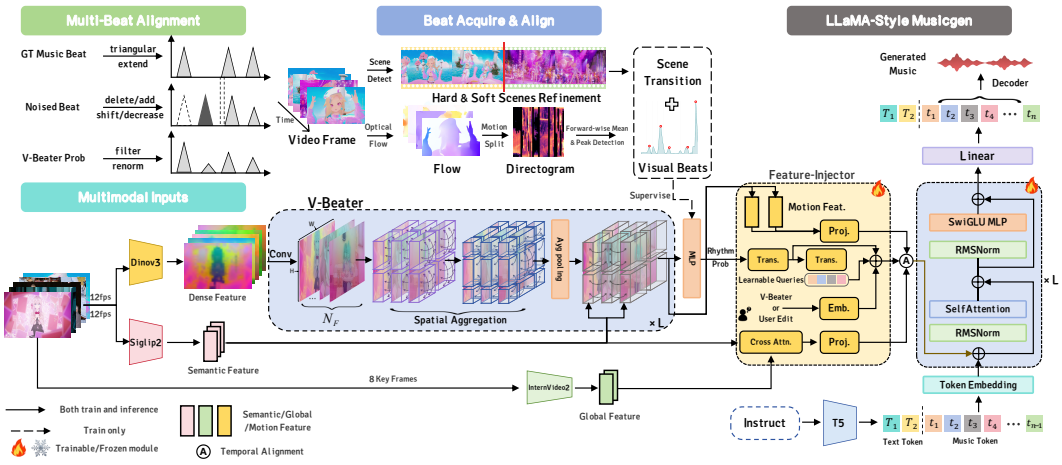

Figure 2: The overview of our model CueMuse, which is consist of end-to-end 12fps visual rhythm detecter V-Beater, multi-condition injector and conditional music generation backbone.

V-Beater performs progressive feature extraction and dimension reduction through multiple layers of window-spatial aggregation and local-temporal aggregation as well as Average Pooling.

Specifically, spatial aggregation performs shifted-window multi-head self-attention Liu et al. (2021) independently for each frame. The window size is fixed at 1/4 of the spatial resolution, which is used to model local spatial relationships within a single frame and explore intra-frame objects and motion textures. Since visual rhythms are all local, temporal aggregation performs local joint spatio-temporal attention under the time window $T_w = 13$ (roughly 1 sec). Meanwhile, the semantic features F_{sem} of the corresponding time also participate in the calculation of attention. To eliminate the inherent bias of window boundary frames caused by access to only one-sided context, we offset the time window by half and perform the same attention calculation. Average Pooling operation $Avgpool(\cdot)$ is applied to compress the spatial dimensions of latent feature. The final layer is directly compressed through an average pooling layer and fed into two separate MLPs to predict video beats and scene transitions respectively. With multi-layer temporal aggregation, the receptive field of each frame eventually expands to approximately 60 frames (5 secs).

Given visual latent $\hat{\mathbf{z}}^{l-1}$ and MLP module $\mathbf{z}^{l-1}$ for layer l, the spatial and temporal aggregation process can be denoted as

$$\hat{\mathbf{z}}^{l} = \mathrm{Attn}_{\mathcal{W}_l}\Big(\mathrm{LN}\big(\mathbf{z}^{l-1}\big);\ Guide\Big) + \mathbf{z}^{l-1}, \tag{1}$$

$$\mathbf{z}^{l} = \mathrm{MLP}\Big(\mathrm{LN}\big(\hat{\mathbf{z}}^{l}\big)\Big) + \hat{\mathbf{z}}^{l}. \tag{2}$$

Where $\mathrm{Attn}_{\mathcal{W}_l}(\cdot)$ is the generic window-attention operator. When $\mathcal{W}_l = \mathcal{W}$, it corresponds to regular-window attention (W-Attn for spatial branch, ST-WAttn for spatio-temporal branch). When $\mathcal{W}_l = \mathcal{SW}$, it denotes shifted-window attention (SW-Attn or ST-SWAttn) with disjoint weights from the regular-window counterpart. $\mathrm{LN}(\cdot)$ denotes Layer Normalization. Frame-level guidance $\mathrm{Guide} = \mathbf{F}_{\mathrm{sem}}$ modulates attention for temporal aggregation, while $\mathrm{Guide} = \emptyset$ for spatial aggregation.

3.3 Multi-conditional Music Generation

3.3.1 LLaMA-style Text-to-Music Backbone

Before fine-tuning the model under multi-constraint conditions, we first modify and retrain Musicgen Copet et al. (2023) based on our internal dataset Table 5 to improve the performance of the base model. Please refer to the Section C for specific model training details and architecture adjustments.

3.3.2 MULTI-CONDITION INJECTION

As mentioned above in Section 1, to complete the controllable V2M task, our music backbone receives four types of features for fine-tuning: **Text Feature** is extracted from music instructions by the T5 text encoder. To enhance the controllability and flexibility of instructions, music instructions consist of random combinations of elements including long/short music descriptions, BPM, chords, instruments, styles, keywords of music, and so on. Please refer to the Section D for the detailed music annotation prompt and instruction synthesis process. During training and inference, text features use the same insertion pattern as the original backbone. We place these features before the sequence of music tokens as a prefix. They guide the generation of music tokens via self-attention.

Video Semantic Feature is the 12-fps semantic feature F_{sem} extracted by SigLIP2 Tschannen et al. (2025). We also aggregate global video features using InternVideo2. This produces a low-dimensional global video embedding vector. The vector extracts the overall video style and emotion, and performs cross-attention with F_{sem}. This lets the final feature F'_{sem} focus on global information. To ensure the model can understand coarse-grained dynamics of videos (rather rhythm points precise to the millisecond level), we also introduce **Video Motion Feature** F_{motion}, which is obtained from the last hidden feature of V-Beater. Previous models have attempted to represent video motion using pixel RGB differencesZhuo et al. (2023) or feature differencesLi et al. (2024b); Lin et al. (2026) of adjacent frames. Since we have trained a visual rhythm detector V-Beater based on optical flow annotations, we can directly use its features which better extracts the motion features of videos. Video Motion Feature F_{motion} is then reshaped to $F'_{motion} \in \mathbb{R}^{T\times D_{backbone}}$ through a Linear projector. **Video Beat Timestamp** is represented as a one-hot sequence $B_v = \{b_t\}_{t=1}^{N_f}$. Its length matches the total number of video frames, and $b_t \in \{0, 1\}$. At the setting of 12 fps, if $b_t = 1$ for a frame image, it means the music beat should fall within $\pm 1/24s \approx \pm 42ms$ of the current moment. However, directly inputting a short single-frame activation with a duration of 20 ms (corresponding to 50 Hz) may leave no preview time for the model to generate an onset. Thus we extend each beat into a triangular window with a total width of 200 ms. In edit mode, the given set of rhythm time points $\{t_b\}$ will be extent to

$$B'_v = \max_i \left[1 - \frac{|t - t_b|}{0.1}\right]_+ . \tag{3}$$

In V-Beater detect mode, Visual rhythm B'_v is directly from the prob of V-Beater's last layer via local peak filtering. B'_v is then expanded to the model hidden layer dimension through a Linear layer $F_{beat} = linear(B'_v)$. An embedding layer is also applied to indicate the source of Beat Timestamp. F_{beat} first passes through a bidirectional temporal Transformer to integrate rhythm information. Then a learned query is used to obtain the global rhythm feature $F_{beat-global}$. The two features are added with the rhythm pattern embedding that identifies the rhythm source (0 for edit mode, 1 for V-Beater detect mode) to get F'_{beat}. Projected feature F'_{sem}, F'_{beat} and F_{motion} is first aligned with 50Hz music tokens by interpolation along the time dimension, then inject into model by added together with music hidden latent input $z \in \mathbb{R}^{T\times K\times D_{backbone}}$. Thus the final latent $z' = z + F'_{sem} + F'_{beat} + F'_{motion}$.

3.4 MULTI-STAGE VIDEO-MUSIC ALIGNMENT

As Section 3.1 defines, given a video clip V with its corresponding ground-truth music Y, we finetune CueMuse on a known text-to-music backbone using next-token prediction to predict the compressed token Y'. To insert Video Beat Timestamp into the model, we adopt a two-stage training strategy. In the first stage of training, we do not use music beat timestamps as input to train the model. This stage forces our model to transfer the capability of music generation to the V2M task. In the second stage, we train two rhythm control capabilities simultaneously. In edit mode, the ground-truth music beat timestamps B_a is added to guide the model to learn the perfect musical beat control signal. In addition, we first use gradually intensified noise data for warmup before inputting the predicted prob of V-Beater. The model is now required to predict compressed music tokens Y' conditioned on visual beat cues and other conditional features. Since not all candidate points predicted by V-Beater correspond to visual beats, we assign high loss weights to the overlapping regions of the two and reduce loss weights for parts where they diverge. This method bridges the domain gap between video signals and music rhythm, enabling CueMuse to actively attempt to align

potential visual rhythm points. For edit mode, we uniformly increase the loss weight for all rhythm timestamps.

During fine-tuning, we keep the next-token prediction objective unchanged to maintain the capacity of the text-to-music backbone. We unfroze parameters of the backbone to fine-tune the text-to-music model. To enhance the model's conditional control capability, we introduce the Classifier-Free Guidance (CFG) technique. Semantic, motion, and rhythm features each have a 0.1 probability of being dropped. The text feature has a 0.3 probability of being dropped. There is also a 0.1 probability of dropping all conditions.

In training, we adopt a masked weighted cross-entropy loss. Given the total codebook size K and audio token time steps T, the loss $\mathcal{L}$ is defined as

$$\mathcal{L} = \frac{\sum_{k,t}^{K,T} M_{kt} w_t, \mathrm{CE}\left(z_{kt}, y_{kt}\right)}{\sum_{k,t} M_{kt} w_t}. \tag{4}$$

When the rhythm condition B_v is satisfied, we apply a triangular pulse function to boost the weights around rhythm instants b_t within a window of $R = 80$ ms:

$$w_t = \begin{cases} 1 + \lambda e_t, & \text{if } B_v \text{ holds,} \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

where

$$e_t = \max_j \left[1 - \frac{|t - j|}{R}\right]_+, b_j \in B_v. \tag{6}$$

4 Experiments

In this section, we elaborate on the implementation details of our experiments and conduct comparative experiments under various conditions to thoroughly evaluate the effectiveness of our proposed method from both subjective and objective perspectives.

4.1 Implementation details

We use Encodec Défossez et al. (2022) as our default compression model. It runs at 32 kHz for mono audio by default. Before training the video background music model, we first partially adjusted the model architecture based on MusicGen. We then trained a text-to-music base model on our internal dataset. See the Section C for further details. Next, we fine-tuned this base model with Universe-MV.

During training, we consistently use the AdamW optimizer. Each GPU handles a batch size of 4 samples. We apply gradient accumulation to scale the effective batch size up to 32. We randomly extract audio segments of 5–30 seconds from Universe-MV for training. Every training sample comes from one single music track. The hyperparameters are β_1 = 0.9, β_2 = 0.95, weight decay of 1e-2, and gradient clipping of 1.0. We adopt a cosine learning rate schedule with a 500-step warm-up and an exponential moving average decay of 0.99. We use 16 H100 GPUs for pre-training over roughly 650 epochs, which takes about 150 hours. For fine-tuning, we use 8 H100 GPUs to train the model for 15 epochs, 10 for stage 1 and 5 for stage 2 with 1000 steps warmup, completing the process in approximately 30 hours. We following original inference strategy from Copet et al. (2023) which uses the top-k sampling strategy and retains the top 250 tokens with temperature set to 1.0. At inference time, we set the sliding window size to 30 seconds and the window overlap to 5 seconds for video duration over 30 seconds. In stage 2, we increase the loss weight of rhythm coincidence points to 2.5 times and reduce the loss weight of rhythm divergence points to 0.5 times.

4.1.1 DATASETS

To objectively evaluate the music-matching capability of current models on general videos, we constructed a general test set named Universe-Bench by combining two widely used test sets Tian et al. (2025); Zhou et al. (2025) in the V2M field and self-craft data. Specifically, since V2M Bench and HarmonySet-eval lack manual filtering mechanisms, we found that some videos or music pieces within them were of poor quality during usage. Therefore, we manually filtered both datasets and obtained 121 and 179 test samples respectively. Meanwhile, to balance the categories, we crawled and screened another 96 high-quality music-video creations online, all of which were produced after 2026 to ensure there is no overlap with the training sets of other models. For more information about the production process and details of our dataset, please refer to Section I.

4.2 EVALUATION METRICS

To quantitatively evaluate the effectiveness of our model, we use a set of metrics to assess the generated music from different models in terms of quality, fidelity, and diversity. These metrics include the Frechet Audio Distance (FAD), Frechet Distance (FD), Kullback-Leibler Divergence (KL) Liu et al. (2023a) as well as density and coverage Naeem et al. (2020). In addition, to characterize the cross-modal alignment degree, we adopted CLAMP3 Score, ImageBind (IB) Score and Sync Score. See Section H for details of these metrics. Finally, following vidmuse Tian et al. (2025), we conduct subjective evaluations of the music from four aspects in Table 9. MOS-Q represents the subjects' comprehensive evaluation of the music quality and fidelity. MOS-M to indicate the aesthetic quality of the generated music. MOS-R is used to evaluate the overall matching degree between music and visual content, including style, musical instruments, rhythm, etc. MOS-A is used to comprehensively assess the overall performance of the model.

4.3 BASELINES

In Table 2, we demonstrate the performance of CueMuse on Video-to-Music (V2M) generation task and Text-and-Video-to-Music (TV2M) generation under Universe-Bench. We also selected several accessible representative methods for benchmarking, including CMT Di et al. (2021), M2UGen Liu et al. (2023b), Video2Music Kang et al. (2024), GVMGen Zuo et al. (2025), Vidmuse Tian et al. (2025), Diff-V2M Ji et al. (2026) for V2M, and CoDi Tang et al. (2023), VIBE Bhosale et al. (2026) AudioX Tian et al. (2026b) for TV2M. We strictly implement these baselines according to their official codes to ensure performance. It is noted that although some methods Tong et al. (2026); Lin et al. (2025); Lokegaonkar et al. (2026) have achieved excellent results, they do not provide open-source access to the models, so further comparisons cannot be conducted.

4.4 SUBJECTIVE USER STUDY

Due to the subjectivity of the V2M task, it is crucial to evaluate the music itself and its alignment with the video through subjective evaluation. To fully evaluate our model, we follow the four evaluation metrics from Tian et al. (2025), including 1) Audio Quality (MOS-Q): focusing on the listening experience of generated music; 2) Video-Music Alignment (MOS-R): evaluating how well the music matches visual content, covering emotion, rhythm and so on; 3) Musicality (MOS-M): assessing the aesthetic quality of music, which examines whether the melody, rhythm, harmony and arrangement are pleasant, complete and structured, and whether they sound like a genuine musical work rather than a random pile of notes. 4) Overall Assessment (MOS-A): comprehensively evaluating the model's performance. In Table 9, we selected four competitive models, namely M2UGen, Vidmuse, Diff-V2M and GVMGen, to compare with CueMuse on V2M task. Participants are asked to score each criterion on a scaled 5-point anchored numerical rating scale. Results are reported as mean-opinion-scores (MOS) for each metrics. See Section J for more analysis.

4.5 MAIN RESULTS

In Table 2, CueMuse demonstrates impressive performance across all metrics, achieving the best or second-best results for all metrics in both V2M and TV2M. Specifically, CueMuse achieves a good balance between audio quality and content alignment. Although Diff-V2M also delivers excellent

Table 2: Quantitative comparison of music quality and alignment on Universe-Bench. Lower is better for FAD, FD, KL; higher is better for Density, Coverage, IB, CLAMP3 and Sync. **Bold** denotes best, underline denotes second-best. Var. denotes that the model supports variable-length generation and is generated in full length.

Model	Var.	FAD (↓)	FD (↓)	KL (↓)	Density (↑)	Coverage (↑)	IB (↑)	CLAMP3 (↑)	Sync (↑)
Video-Only to Music									
CMT Di et al. (2021)	×	8.7584	45.474	1.2530	0.4196	0.0389	0.1230	0.0739	0.0839
M2UGen Liu et al. (2023b)	×	3.7730	27.6753	1.3388	0.5073	0.1370	0.1057	0.0835	0.1659
Video2Music Kang et al. (2024)	×	20.1142	94.8204	1.6576	0.4700	0.0078	0.0689	0.0917	0.1204
GVMGen Zuo et al. (2025)	✓	3.2091	19.8722	1.1889	0.5104	0.1982	0.1366	0.0881	0.1519
VidMuse Tian et al. (2025)	✓	2.5096	19.1386	**0.9277**	0.6069	0.2264	**0.1613**	0.1104	0.1668
Diff-V2M Ji et al. (2026)	✓	2.7026	**17.3108**	1.0259	**0.8327**	**0.2601**	0.1532	0.0980	0.1556
CueMuse (Ours)	✓	**2.2010**	18.5054	0.9906	0.6855	0.2282	0.1577	**0.1113**	**0.1711**
Text + Video to Music									
CoDi Tang et al. (2023)	×	8.0605	40.7679	1.8947	-	-	0.0911	0.0893	-
VIBE Bhosale et al. (2026)	×	4.6415	26.2569	1.3719	0.2689	0.1095	0.0863	0.0967	0.0320
AudioX Tian et al. (2026b)	×	2.5018	17.9954	0.9778	0.5217	0.2227	0.1884	0.1492	0.0390
CueMuse (Ours)	✓	**1.8376**	**16.2303**	**0.8048**	**0.6523**	**0.2333**	0.1724	0.1382	**0.1824**

Table 3: Ablation study on different elements of CueMuse under on Universe-Bench. Lower is better for FAD, FD, KL; higher is better for Density, Coverage, IB, CLAMP3 and Sync. **Bold** denotes best, underline denotes second-best. Var. denotes that the model supports variable-length generation and is generated in full length.

Model	Var.	FAD (↓)	FD (↓)	KL (↓)	Density (↑)	Coverage (↑)	IB (↑)	CLAMP3 (↑)
CueMuse	✓	**2.2010**	18.5054	**0.9906**	0.6855	0.2282	**0.1577**	**0.1113**
CueMuse w/o motion	✓	2.3737	20.2242	1.0388	**0.8760**	**0.7146**	0.1462	0.1031
CueMuse w rhythm	✓	2.3166	18.2158	1.0135	0.6071	0.2104	0.1467	0.1091

audio quality, it underperforms our model in modality relevance. Notably, in terms of content alignment, CueMuse is only outperformed by Vidmuse on the ImageBind score, yet Vidmuse is trained on data filtered by ImageBind. For the TV2M task, CueMuse performs even better, outperforming the baseline models on nearly all metrics. Since some samples generated by CoDi are closer to sounds, we omit some metrics.Subjective metrics also show that CueMuse attains the highest scores on all metrics and leads the second-place model by a substantial margin in the overall score. This fully demonstrates the comprehensive and flexible advantages of CueMuse.

4.6 Ablation Studies

In Table 3, we present ablation experiments for two components. First, we remove the branch related to motion features and retrained CueMuse. Second, we report the results of the model control with rhythm. The results demonstrate that adding motion features helps the model generation. However, most metrics partially decline after rhythm control is introduced. This may be because the forced control of rhythm features breaks the original model constraints. Therefore, rhythm control is better regarded as a professional requirement rather than the default setting for general video dubbing. In the appendix, we also provide an analysis of the impact of data volume on model performance and the stability of rhythm control.

5 Conclusion

In this paper, we propose CueMuse, a general video-to-music model supporting multi-condition control. To this end, we collect a large-scale dataset from the web and build a rigorous pipeline to filter and obtain the valid dataset Universe-MV. We then perform multi-condition fine-tuning based on the text-to-music model, and employ the rhythm extraction module V-Beater to obtain fine-grained cues and motion features from videos for alignment. To validate our approach, we further construct the Universe-Bench test set, and conduct benchmark tests with a series of state-of-the-art methods as baselines. Through comprehensive quantitative studies and qualitative analysis, our method has demonstrated its superiority over existing approaches.

AI USE STATEMENT

In this work, we used generative AI tools for help develop theoretical models or conceptual frameworks, assist in the writing of proofs, assist with translation. We have not used generative AI tools for design or provide feedback on research methodology or experiments, propose or refine hypotheses, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. And Generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims are not applicable to this work. Additionally, we used generative AI tools for suggest experimental parameters, brainstorming, propose a title or keywords for a research paper. We have reviewed all AI-assisted work. We carefully examined the translation text generated by the large language model and the details of the paper. One researcher adjusted the idea based on the large model and achieved better results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, 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.

Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995.

Aryan Vijay Bhosale, Vaibhavi Lokegaonkar, Vishnu Raj, Gouthaman KV, Sreyan Ghosh, Ramani Duraiswami, Lie Lu, and Dinesh Manocha. Vibe: Video instruction-aligned background music generation. *arXiv preprint arXiv:2608.30125*, 2026.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. ICML, 2019.

Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer. *arXiv preprint arXiv:1809.07600*, 2018.

Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 1206–1210. IEEE, 2024.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

Abe Davis and Maneesh Agrawala. Visual rhythm and beat. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2037–2045, 2021.

Ragnhild Eg, Carsten Griwodz, Pål Halvorsen, and Dawn Behne. Audiovisual robustness: exploring perceptual tolerance to asynchrony and quality distortion. *Multimedia Tools and Applications*, 74 (2):345–365, 2015.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Zach Evans, Julian D Parker, Matthew Rice, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio 3. *arXiv preprint arXiv:2605.17991*, 2026.

Seth Forsgren and Hayk Martiros. Riffusion-stable diffusion for real-time music generation. *URL https://riffusion. com*, pp. 27, 2022.

Dongxu Ge, Shansong Liu, Cheng Gong, Xiao-Lei Zhang, Chi Zhang, and Xuelong Li. Towards expressive and faithful audio-to-image generation: A unified multimodal dataset and synthesis framework. *arXiv preprint arXiv:2608.09529*, 2026.

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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind one embedding space to bind them all. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15180–15190. IEEE, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pp. 131–135. IEEE, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Shulei Ji, Zihao Wang, Jiaxing Yu, Xiangyuan Yang, Shuyu Li, Songruoyao Wu, and Kejun Zhang. Diff-v2m: A hierarchical conditional diffusion model with explicit rhythmic modeling for video-to-music generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22219–22227, 2026.

Jaeyong Kang, Soujanya Poria, and Dorien Herremans. Video2music: Suitable music generation from videos using an affective multimodal transformer model. *Expert Systems with Applications*, 249:123640, 2024.

Haven Kim, Zachary Novack, Weihan Xu, Julian McAuley, and Hao-Wen Dong. Video-guided text-to-music generation using public domain movie collections. *arXiv preprint arXiv:2506.12573*, 2025.

Taejun Kim and Juhan Nam. All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio. In *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5. IEEE, 2023.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

Luca Lanzendörfer, Florian Grötschla, Emil Funke, and Roger Wattenhofer. Disco-10m: A large-scale music dataset. *Advances in Neural Information Processing Systems*, 36:54451–54471, 2023.

Daniel J Levitin, Karon MacLean, Max Mathews, Lonny Chu, and Eric Jensen. The perception of cross-modal simultaneity (or "the greenwich observatory problem" revisited). In *AIP Conference Proceedings*, volume 517, pp. 323–329. American Institute of Physics, 2000.

Ruiqi Li, Siqi Zheng, Xize Cheng, Ziang Zhang, Shengpeng Ji, and Zhou Zhao. Muvi: Video-to-music generation with semantic alignment and rhythmic synchronization. *arXiv preprint arXiv:2410.12957*, 2024a.

Sifei Li, Binxin Yang, Chunji Yin, Chong Sun, Yuxin Zhang, Weiming Dong, and Chen Li. Vidmusician: Video-to-music generation with semantic-rhythmic alignment via hierarchical visual features. *arXiv preprint arXiv:2412.06296*, 2024b.

Yan-Bo Lin, Yu Tian, Linjie Yang, Gedas Bertasius, and Heng Wang. Vmas: Video-to-music generation via semantic alignment in web music videos. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1155–1165. IEEE, 2025.

Yan-Bo Lin, Jonah Casebeer, Long Mai, Aniruddha Mahapatra, Gedas Bertasius, and Nicholas J Bryan. V2m-zero: Zero-pair time-aligned video-to-music generation. *arXiv preprint arXiv:2603.11042*, 2026.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023a.

Shansong Liu, Atin Sakkeer Hussain, Qilong Wu, Chenshuo Sun, and Ying Shan. M^2ugen: Multi-modal music understanding and generation with the power of large language models. *arXiv preprint arXiv:2311.11255*, 2023b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021.

Vaibhavi Lokegaonkar, Aryan Vijay Bhosale, Vishnu Raj, Gouthaman KV, Ramani Duraiswami, Lie Lu, Sreyan Ghosh, and Dinesh Manocha. Video-robin: Autoregressive diffusion planning for intent-grounded video-to-music generation. *arXiv preprint arXiv:2604.17656*, 2026.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning*, pp. 7176–7185. PMLR, 2020.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Fan Qi, L Ni, and C Xu. Harmonizing pixels and melodies: Maestro-guided film score generation and composition style transfer. *arXiv preprint arXiv:2411.07539*, 2024.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning*, pp. 4364–4373. PMLR, 2018.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Jiawei Shao and Xuelong Li. Ai flow at the network edge. *IEEE Network*, 40(1):330–336, 2025.

Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12435–12446. IEEE, 2023.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 11218–11221, 2024.

Kun Su, Judith Yue Li, Qingqing Huang, Dima Kuzmin, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, et al. V2meow: Meowing to the visual beat via video-to-music generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4952–4960, 2024.

Jiatian Sun, Longxiulin Deng, Triantafyllos Afouras, Andrew Owens, and Abe Davis. Eventfulness for interactive video alignment. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems*, 36:16083–16099, 2023.

Jinchuan Tian, Sang-gil Lee, Zhifeng Kong, Sreyan Ghosh, Arushi Goel, Chao-Han Huck Yang, Wenliang Dai, Zihan Liu, Hanrong Ye, Shinji Watanabe, et al. Ualm: Unified audio language model for understanding, generation and reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 24563–24585, 2026a.

Zeyue Tian, Zhaoyang Liu, Ruibin Yuan, Jiahao Pan, Qifeng Liu, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. Vidmuse: A simple video-to-music generation framework with long-short-term modeling. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18782–18793. IEEE, 2025.

Zeyue Tian, Zhaoyang Liu, Yizhu Jin, Ruibin Yuan, Liumeng Xue, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. Audiox: a unified framework for anything-to-audio generation. In *International Conference on Learning Representations*, volume 2026, pp. 110625–110651, 2026b.

Xinyi Tong, Yiran Zhu, Jishang Chen, Chunru Zhan, Tianle Wang, Sirui Zhang, Nian Liu, Tiezheng Ge, Duo Xu, Xin Jin, et al. Video echoed in music: Semantic, temporal, and rhythmic alignment for video-to-music generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25983–25991, 2026.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European conference on computer vision*, pp. 396–416. Springer, 2024.

Shangda Wu, Guo Zhancheng, Ruibin Yuan, Junyan Jiang, Seungheon Doh, Gus Xia, Juhan Nam, Xiaobing Li, Feng Yu, and Maosong Sun. Clamp 3: Universal music information retrieval across unaligned modalities and unseen languages. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2605–2625, 2025.

Zhifeng Xie, Qile He, Youjia Zhu, Qiwei He, and Mengtian Li. Filmcomposer: Llm-driven music production for silent film clips. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13519–13528. IEEE, 2025.

Kaituo Xu, Yan Jia, Kai Huang, Junjie Chen, Wenpeng Li, Kun Liu, Feng-Long Xie, Xu Tang, and Yao Hu. Fireredasr2s: A state-of-the-art industrial-grade all-in-one automatic speech recognition system. *arXiv preprint arXiv:2603.10420*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiashuo Yu, Yaohui Wang, Xinyuan Chen, Xiao Sun, and Yu Qiao. Long-term rhythmic video soundtracker. In *International Conference on Machine Learning*, pp. 40339–40353. PMLR, 2023.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

Lei Zhao, Linfeng Feng, Dongxu Ge, Rujin Chen, Fangqiu Yi, Chi Zhang, Xiao-Lei Zhang, and Xuelong Li. Uniform: A unified multi-task diffusion transformer for audio-video generation. *arXiv preprint arXiv:2502.03897*, 2025.

Zitang Zhou, Ke Mei, Yu Lu, Tianyi Wang, and Fengyun Rao. Harmonyset: A comprehensive dataset for understanding video-music semantic alignment and temporal synchronization. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3152–3162. IEEE, 2025.

Le Zhuo, Zhaokai Wang, Baisen Wang, Yue Liao, Chenxi Bao, Stanley Peng, Songhao Han, Aixi Zhang, Fei Fang, and Si Liu. Video background music generation: Dataset, method and evaluation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15591–15601. IEEE, 2023.

Heda Zuo, Weitao You, Junxian Wu, Shihong Ren, Pei Chen, Mingxu Zhou, Yujia Lu, and Lingyun Sun. Gvmgen: A general video-to-music generation model with hierarchical attentions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 23099–23107, 2025.

A Appendix Overview

Our appendix consists of:
1. Related Works.
2. LLaMA-Style Musicgen Reproduce.
3. Music-Video Data Pipeline.
4. Music-Video Data Analysis.
5. Visual Rhythm Extraction.
6. Additional Experiments.
7. Details of Evaluation Metrics.
8. Details of Evaluation Dataset.
9. User Study Interface.

B Related Works

B.1 Conditional Music Generation

Conditional music generation represented by text has undergone long-term exploration. Typically music generation can be divided into symbolic music generation as well as waveform music generation. The former uses structured representations such as MIDI as the intermediate link to render music, while the latter attempts to generate audio waveforms directly. Early researches mainly focus on symbolic music generation, such as MIDI-VAE Brunner et al. (2018) and MusicVAE Roberts et al. (2018). In recent years, waveform generation has attracted considerable research attention. As it directly models audio signals and can simulate better natural musical details. Works like Riffusion Forsgren & Martiros (2022), MusicLM Agostinelli et al. (2023), and MusicGen Copet et al. (2023) first use neural audio codec such as soundstream Zeghidour et al. (2021) and encodec Défossez et al. (2022) to convert continuous music signals into discrete tokens, and then train their text-conditional music generation model to predict their codebooks. Most recently, music generation methods based on diffusion Chen et al. (2024); Evans et al. (2025), flow-matching Evans et al. (2026), and the Unified Modeling method Tang et al. (2023); Tian et al. (2026a;b) have flourished.

B.2 Video-to-Music Generation

CMT Di et al. (2021) first proposed the video-to-music generation task by regarding disentangled visual features as musical attributes during inference to predict music. V-MusProd Zhuo et al. (2023) and Video2Music Kang et al. (2024) further perform end-to-end video-music alignment learning through deep neural networks, yet they ignore the effect of visual semantics. V2Meow Su et al. (2024) employs multiple encoders to extract conditional information and progressively predict the final musical waveform. VMAS Lin et al. (2025) employs contrastive learning to align high-level video features with the generated music features on a total of 2.2M training pairs. VidMuse Tian et al. (2025) proposes a large-scale dataset and achieves the capture of global and local video information through long and short-term modeling. V2M-Zero Lin et al. (2026) achieves generation by using visual curves to replace music event curves during the inference stage without video-music training pairs. Although achieving decent performance, these works fail to consider the constraining capability of text for video soundtracks and are somewhat inferior in practicality. Recently Tong et al. (2026) and Video-Robin Lokegaonkar et al. (2026) consider text-constrained video music matching models, yet they are limited by the small dataset size to achieve universal generation. Therefore, it remains challenging to achieve general cross-modal semantic constraints, motion constraints and temporal alignment in video-to-music generation.

B.3 Visual Beats & Alignment

The alignment of vision and audio has always been a key focus in cross-modal research Zhao et al. (2025); Ge et al. (2026). From its early days, V2M task has always tried to align video motion with musical rhythms. V-MusProd Zhuo et al. (2023) and VidMusicianLi et al. (2024b) uses differences of adjacent-frame RGB or CLIP features as motion to predict the tempo of music. V2M-Zero Lin et al. (2026) employs DINOv2 to extract and compute the cosine similarity of frames as video

rhythm. Diff-V2M Ji et al. (2026) points out that prior works lack explicit visual rhythm modeling, and explores the alignment effect between RGB differences of adjacent frames and multiple rhythm representations. Nevertheless, these works naively adopt differences between adjacent frames or features as video rhythm, and fail to effectively disentangle intrinsic motion information within visuals. VMAS Lin et al. (2025) takes a further step by leveraging optical-flow information Shi et al. (2023) to align visual rhythm with musical beats. However, it does not explicitly extract visual rhythm. Meanwhile, it performs alignment using optical-flow peaks instead of peaks of optical-flow acceleration Davis & Agrawala (2018), which sacrifices alignment precision and editing capability. Meanwhile, Tong et al. (2026) explicitly employs a shot detection model to obtain transition information as the condition for V2M, yet it fails to consider the more crucial motion information within video clips. Existing methods lack full and precise alignment as well as explicit modeling of visual rhythms, making it difficult to generate music flexibly according to requirements. To address this issue, we propose CueMuse, a general-purpose video-to-music model that can predict visual rhythm within videos at 12 fps while driven by integrating multiple conditions to achieve controllable generation.

C LLaMA-Style Musicgen Reproduce

Table 4: Experimental model architecture of music backbone models.

Module	Musicgen-LLaMA	Musicgen-Large	Musicgen-Medium
Transformer Layers	48	48	48
Hidden Dimension	2048	8192	1536
Attention heads	32	32	24
Codebook	4	4	4
Codebook cardinality	2048	2048	2048
Audio frame rate	50 Hz	50 Hz	50 Hz
EnCodec	32 kHz, mono	32 kHz, mono	32 kHz, mono
Pattern	Delayed [0, 1, 2, 3]	Delayed [0, 1, 2, 3]	Delayed [0, 1, 2, 3]
Text encoder	Frozen T5-base	Frozen T5-base	Frozen T5-base
Transformer norm	RMSNorm	LayerNorm	LayerNorm
FFN	SwiGLU, w1/w3/w2	FFN	FFN
Learnable Parameters	1.6B	3.3B	1.5B

Table 5: Statistics of instrumental music used for training Musicgen-LLaMA.

Music Dataset	Number	Duration (h)
Free Music Archive (FMA) Defferrard et al. (2016)	106,574	8.2k
MTG-Jamendo Bogdanov et al. (2019)	55,701	3.8k
Internal Music Data	1.05M	64.5k
Total	1.2M	76.5k

To generate higher-quality music, we restructured Musicgen-Large and trained it from scratch. Specifically, we adopted the LLaMA-style transformer, replaced LayerNorm with RMSNorm, and swapped sinusoidal positional encoding for RoPE encoding. We retained Pre-Norm, or normalization before the sub-layer from Musicgen-Large. For the FFN layer, we replaced ReLU with SwiGLU. Meanwhile, to cut the model's parameter count and reduce training and inference costs, we set the Hidden Dimension to 2048, matching the configuration of Musicgen-Medium. We also adjusted how we inject text instructions. In the original Musicgen, text information interacts with music features through cross-attention. In our approach, we place text information directly before music tokens as a prefix and feed both into the transformer at once. Since Musicgen does not release its training dataset, we trained our model on internal music data. In Table 4, we show the differences between our Musicgen-LLaMA and the standard model. We also present the statistics of this training music dataset in Table 5.

D MUSIC-VIDEO DATA PIPELINE

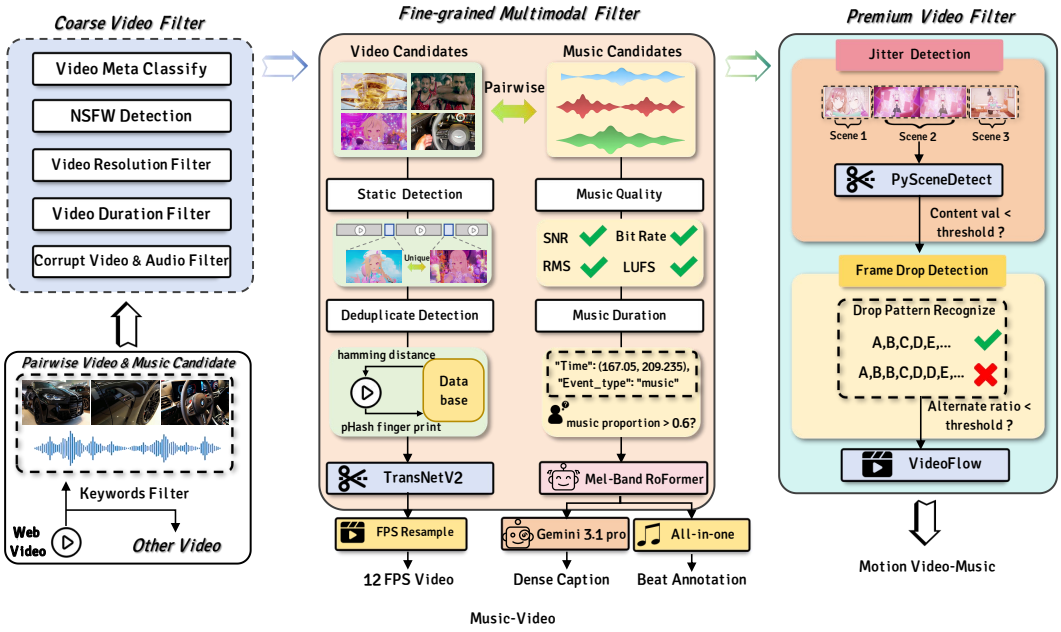

Figure 3: The overview of our data construct pipeline, we adopt premium video filter to further filter video suitable for optical label.

In Figure 3, we present the entire pipeline through which we obtain high-quality music-video data from noisy web video data and ultimately build Universe-MV. These data are mainly sourced from Bilibili and YouTube, covering a wide range of genres such as landscapes, dances, advertisements, narrative videos, hybrid cuttings and so on, as shown in Figure 1 (c). We hope the model can learn a wide variety of music videos to master the general visual-music correspondence. During the crawling process, we retained video data from recent years, as the quality of early music-video data was relatively lower. In the end, we collected more than 30,000 hours of source data. This data underwent simple filtering of meta information via automated scripts to ensure it centered on the music-video theme.

Although we have approximately 30,000 hours of data, we only selected a 10,000 hour subset for filtering. This choice accounts for our training resources and balanced data categories. We prioritized extracting data clips between 30 seconds and 3 minutes long. Raw videos often contain many low-quality samples. To solve this problem, we created strict procedures. These procedures remove unqualified data and generate a clean dataset. The overall pipeline of data filtering and labeling is shown in Figure 3.

Our data process ranges from Coarse Filtering to Fine-grained Filtering and then to the final Premium Filtering. In **Coarse Filtering**, we designed a rule-based filtering strategy for candidate data. Specifically, we removed data with missing or corrupted audio or video. We further filtered videos with too short duration (less than 5 seconds). Since we segment videos by music later, there is no restriction on the maximum video duration. We also checked video resolution and filtered out videos below 480p. An NSFW detector is used to filter inappropriate content. Finally, we categorized the videos into 9 classes Table 6 according to their meta information. Further data analysis please refer to Section E.

To further ensure the quality of our data, we performed more **Fine-grained Filtering** on audio and visual modalities. For the audio paired with videos, using low-quality music for training will cause the model to lose good generation quality in the pre-training stage. We adopt multiple approaches to filter and obtain high-quality music. Signal-to-Noise Ratio (SNR) measures the background noise level of audio using the ratio of the useful audio signal to background noise. Bit Rate describes the number of bits transmitted per unit time; The higher the bit rate, the more information is retained

in audio and so as better corresponding audio quality. Root Mean Square (RMS) describes the average energy of audio over a period of time and is used to filter excessive noise. Loudness Units Full Scale (LUFS), as a modern standardized loudness metric, further characterizes the human-ear-weighted actual auditory perception and is also used for noise filtering. Besides, the original videos may contain excessive audio segments without music. Such data can impair the final training performance of the model. To guarantee that the resulting dataset consists of high-quality video-music pairs, we retain only videos with a larger proportion of musical content. We leverage the sound event detection model FireRedVAD Xu et al. (2026), which can provide accurate timestamps for music segments. We require that the duration of music in the entire video should be greater than 60%.

For the visual modality, we first apply Static Detection to remove some videos that only consist of static images. Specifically, we extract video frames at a rate of 2 frames per second and calculate the proportion of changed frames among all sampled frames. Based on experience, we discard videos with a change rate below the threshold of 0.1. In addition, we also filtered duplicate videos. We record video fingerprints for each video based on pHash (perceptual hashing) to build a database. By comparing the Hamming distance of the hash value of each frame between the video and historical videos in the database, two videos are judged as duplicates when more than 80% of their frames are similar. This helps us rule out copies that have undergone video compression or minor cropping. After filtering both video and audio, we use TransNetV2 to annotate transition information in the video and downsample the content to a unified frame rate of 12 FPS. And Mel-Band RoFormer is used for music source separation of the audio. In particular, we use All-in-one to annotate music beats. To ensure the accuracy of annotations, we further perform secondary beat annotation within the music segments annotated by All-in-one Kim & Nam (2023) and compare the similarity between the two sets of results. Next, we use Gemini 3.1 Pro to annotate the separated music. Since some video creations involve splicing different pieces of music, we first segment the music based on the BPM of music segments and feed the segments into Gemini 3.1 Pro. Further segmentation is then carried out according to the annotations from Gemini 3.1 Pro. We annotated various musical elements in detail using Gemini 3.1 Pro, and the prompt can be found in Figure 7.

In **Premium Video Filter**, we further filtered out videos with severe jitter and frame drops. This is because although these videos meet the soundtrack matching criteria, the optical flow model performs poorly on them. To detect jitter, we refer to the detection method of MovieGen Polyak et al. (2024), which uses the frame content difference value of PySceneDetect to characterize the jitter level of scenes. We perform detection within each scene clip to avoid interference from original scene cuts. And evaluate jitter by counting the number of frame content difference values exceeding the threshold per second. For frame drop detection, videos of this type feature intermittent duplicate frames that are difficult to capture with the human eye, which causes abnormal detection by optical flow models. To address this, we use frame template matching to filter videos with regular frame drops. Considering false positives, filter out the model when its repetition rate exceeds the threshold of 0.1.

E Music-Video Data Analysis

In Table 6, we categorically record all data contained in Universe-MV. We classify whole music videos into 9 sub-categories: **TV** mainly includes original clips of series, TV dramas and movies, as well as anime clips including Anime Opening and Ending. **Advertisement** includes product promotion, promotional videos and other videos. **Dance** videos cover a wide range of solo and group dance scenes with multiple shooting perspectives. **HumanMV** includes official music video of world wide music track. **Hybrid Cutting** includes artistic creation based on a variety of video materials. **Landscape** includes landscape videos and some music videos for atmosphere rendering. **Musical** mainly includes musicals, dramas, etc. **Musical Instrument** includes creative videos of various musical instruments. **Sport** refers to sports-themed videos, including sports highlights, clips and fitness videos.

In Table 7, we further report the data used to train V-Beater, which was additionally filtered by the Premium Video Filter mentioned in Section D. Since the optical flow extraction model consumes substantial resources, and the visual rhythm in videos of some categories is not prominent, we

Prompt for Music Caption

Core Task
Analyze this piece of music and give followings in **JSON format**:
short captions (around 20 words) & long captions (40-80 words),
musical key: The musical key of the music (e.g., "C Major", "G Minor"),
keywords: 3-5 comma-separated strings that capture the music's key attributes (e.g., "epic, orchestral, adventure, heroic"),
instruments: All identifiable instruments used in the music as comma-separated strings (e.g., "cello, double bass, flute, harp, horn"),
genre: Fill in 1-5 musical genres (e.g., "classical", "pop", "jazz", "electronic") incluing the major genre of the music in the first place and following other genres if detected,
moods: 1-4 strings representing the music's dominant moods (e.g., "happy, sad, energetic, calm"),
time period: Corresponding time interval in the input audio.
All time periods must be relative to the current input audio, beginning at 0:00.
If the music is obviously composed of multiple edited clips, analyze each clip separately; Otherwise, answer by merging information as much as possible. If no music is detected, return an empty list [].
Example Output: ["time period": "0:00-1:30", "key": "A minor", "keywords": "romantic, piano solo, expressive, melancholic, rubato", "genre": "classical", "instrument": "piano", "moods": ["emotional", "intimate", "reflective"], "short_caption": "Romantic-style piano solo with a delicate and smooth melody, carrying a slight sense of sadness.", "long_caption": "A romantic-style piano solo with a free rhythm and delicate emotions. The music is based on soft left-hand arpeggios, with the right-hand melody flowing lightly in the treble range, carrying a Chopin-like lyrical quality. In the middle section, the dynamics and use of the pedal gradually intensify, showing intense emotional fluctuations. The coda returns to calmness with a warm tone." "time period": "1:31-3:00", "key": "C Major", "keywords": "orchestral, uplifting, cinematic, triumphant, epic", "genre": "orchestral", "instrument": "strings, brass, percussion", "moods": ["uplifting", "inspiring", "heroic"], "short_caption": "An uplifting orchestral piece with a triumphant and epic feel, featuring soaring strings and bold brass.", "long_caption": "This orchestral piece opens with a bright and uplifting theme, characterized by soaring string melodies and bold brass fanfares. The music builds to a triumphant climax, evoking a sense of heroism and grandeur. The percussion section adds rhythmic drive, while the harmonic progressions create a sense of resolution and triumph. The overall mood is one of inspiration and cinematic grandeur, making it suitable for epic film scores or motivational contexts."]

Figure 4: Prompt for Music Caption.

selected four music categories where visual rhythm exerts a strong influence on music to train V-Beater.

Table 6: Statistics of total music videos used for training.

Music Class	Number	Duration (h)	Avg Duration
TV	18,043	987.3	197.0 s
Advertisement	8,693	682.0	282.4 s
Dance	11,642	887.8	274.5 s
HumanMV	15,397	977.3	228.5 s
Hybrid Cutting	16,837	990.3	211.7 s
Landscape	17,533	967.2	198.6 s
Musical	3,323	343.4	372.0 s
Musical Instrument	6,943	453.9	235.3 s
Sport	4,365	647.7	534.2 s
Total	102,776	6936.9	243.0 s

Table 7: Statistics of music videos used for training V-Beater.

Music Class	Number	Duration (h)	Avg Duration
TV	9,467	545.9	197.0 s
Advertisement	5,805	520.4	282.4 s
Dance	9,664	663.9	274.5 s
Hybrid Cutting	9,987	585.3	211.7 s
Total	34,923	2315.5	238.7 s

F VISUAL RHYTHM EXTRACTION

Given a dense optical-flow field at frame t,

$$\mathbf{F}_t(x, y) = \big(f_x(x, y, t), f_y(x, y, t)\big), \tag{7}$$

its flow magnitude and orientation are defined as

$$A_t(x, y) = \sqrt{f_x^2(x, y, t) + f_y^2(x, y, t)}, \tag{8}$$

$$\Theta_t(x, y) = \operatorname{atan2}\big(f_y(x, y, t), f_x(x, y, t)\big) + \pi, \tag{9}$$

where $A_t(x, y) \in [0, \infty)$ denotes the local motion magnitude and $\Theta_t(x, y) \in [0, 2\pi)$ denotes the corresponding flow orientation.

To characterize the spatial concentration of motion magnitudes, we compute the Gini coefficient G_t over the sorted non-zero amplitudes

$$a_1 \le a_2 \le \cdots \le a_n, \tag{10}$$

where

$$G_t = \frac{2\sum_{i=1}^{n} i\, a_i}{n\sum_{i=1}^{n} a_i} - \frac{n+1}{n}. \tag{11}$$

A modulation factor is then defined as

$$w_G = 0.01 + G_t, \tag{12}$$

which adaptively emphasizes concentrated motion while retaining sensitivity to distributed motion.

We discretize the flow orientations into N_{bins} angular bins. For an angular bin centered at ϕ, the corresponding indicator function is defined as

$$\mathbb{1}_{\theta}(\phi) = \begin{cases} 1, & \text{if } d_{\mathbb{S}^1}(\theta, \phi) \le \dfrac{\pi}{N_{\text{bins}}}, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where $d_{\mathbb{S}^1}(\cdot, \cdot)$ denotes the circular angular distance.

Let Ω_t denote the valid image region after removing the boundary dead zone, and let $m_0(x, y, t) \in \{0, 1\}$ be a noise mask that retains only pixels whose flow magnitude exceeds the noise floor n_f. We construct three directional motion representations with different magnitude weightings:

$$H_0(t, \phi) = w_G \sum_{(x,y)\in\Omega_t} m_0(x, y, t)\, \mathbb{1}_{\theta}\big(\Theta_t(x, y)\big), \tag{14}$$

$$H_1(t, \phi) = w_G \sum_{(x,y)\in\Omega_t} m_0(x, y, t) \sqrt{A_t(x, y)}\, \mathbb{1}_{\theta}\big(\Theta_t(x, y)\big), \tag{15}$$

$$H_2(t, \phi) = w_G \sum_{(x,y)\in\Omega_t} m_0(x, y, t) A_t(x, y)\, \mathbb{1}_{\theta}\big(\Theta_t(x, y)\big). \tag{16}$$

These three representations progressively increase the sensitivity to motion magnitude, with H_0, H_1, and H_2 corresponding to uniform, sub-linear, and linear amplitude weighting, respectively.

For simplicity, let

$$\angle \mathbf{F}_t(x, y) = \Theta_t(x, y), (x, y) \in \Omega_t, \qquad |\mathbf{F}_t(x, y)| = m_0(x, y, t) A_t(x, y). \tag{17}$$

The second-order directional motion histogram (Directogram) is then written as

$$\mathcal{D}(t, \phi) = \sum_{(x,y)} |\mathbf{F}_t(x, y)|\, \mathbb{1}_{\phi}\big(\angle \mathbf{F}_t(x, y)\big). \tag{18}$$

To suppress isolated fluctuations and obtain a temporally stable representation, we smooth $\mathcal{D}(t, \phi)$ using Gaussian and median filtering, yielding

$$\widetilde{\mathcal{D}}(t, \phi). \tag{19}$$

We then capture abrupt decreases in directional motion energy through a temporal finite difference:

$$\Delta\widetilde{\mathcal{D}}(t, \phi) = \widetilde{\mathcal{D}}(t, \phi) - \widetilde{\mathcal{D}}(t - 1, \phi). \tag{20}$$

An impact response is obtained by retaining only negative temporal changes:

$$I(t, \phi) = \max\Big(-\Delta\widetilde{\mathcal{D}}(t, \phi),\, 0\Big). \tag{21}$$

This operation highlights moments at which directional motion undergoes a sudden decrease, which often corresponds to visually salient impacts or abrupt motion stops.

The directional responses are aggregated across all angular bins to form a one-dimensional visual impact envelope:

Table 8: Quantitative comparison of music quality on text-to-music generation. Lower is better for FAD, FD, KL; Higher is better for Density, Coverage, IB, CLMAP3. **Bold** denotes best, underline denotes second-best. Var. denotes that the model supports variable-length generation and is generated in full length.

Model	Var.	FAD (↓)	FD (↓)	KL (↓)	Density (↑)	Coverage (↑)	IB (↑)	CLAMP3 (↑)
Text-Only to Music								
Musicgen-Medium Copet et al. (2023)	✓	2.4341	16.3883	0.8240	**0.8205**	0.2643	0.1484	0.1321
SAO Evans et al. (2025)	×	2.4537	17.2102	0.8102	0.7482	0.2364	0.1449	0.1301
SA3-Medium Evans et al. (2026)	✓	**1.8756**	**13.3954**	**0.7344**	0.7995	**0.3010**	**0.1726**	**0.1496**
Musicgen-LLaMA (Ours)	✓	2.2303	15.7598	0.7621	0.6918	0.2378	0.1420	0.1419
CueMuse (Ours)	✓	1.9768	18.0393	0.8436	0.5843	0.2169	0.1401	0.1403

Table 9: Subjective comparison of multiple metrics on Universe-Bench. All indicators are better when they are higher. **Bold** denotes best.

Model	MOS-Q (↑)	MOS-M (↑)	MOS-R (↑)	MOS-A (↑)
	Quality	Musicality	Relation	Overall
M2UGen Liu et al. (2023b)	65.90 ± 20.94	66.60 ± 20.82	55.05 ± 23.70	59.65 ± 20.59
GVMGen Zuo et al. (2025)	59.15 ± 21.99	56.15 ± 22.12	53.70 ± 23.78	53.60 ± 21.41
VidMuse Tian et al. (2025)	60.00 ± 22.03	62.30 ± 21.68	66.35 ± 22.76	62.15 ± 21.58
Diff-V2M Ji et al. (2026)	58.80 ± 21.72	52.75 ± 20.92	51.15 ± 23.15	50.65 ± 20.75
CueMuse (Ours)	**70.95 ± 19.13**	**69.65 ± 21.14**	**71.90 ± 21.20**	**67.05 ± 20.70**

$$u_v(t) = \frac{1}{N_{\text{bins}}} \sum_{k=0}^{N_{\text{bins}}-1} I(t, \phi_k). \tag{22}$$

Finally, we detect salient peaks in $u_v(t)$ using a Median Absolute Deviation (MAD)-based adaptive threshold δ. Given the discrete-time envelope signal $\{u_v(t)\}$, the threshold is computed as

$$\text{MED} = \text{median}\left(\{u_v(t)\}\right), \tag{23}$$

$$\text{MAD} = \text{median}\left(\left|u_v(t) - \text{MED}\right|\right), \tag{24}$$

$$\delta = \max\left(1.4826 \cdot \text{MAD},\ \delta_f\right), \tag{25}$$

where the factor 1.4826 asymptotically scales MAD to one standard deviation under Gaussian noise assumptions, and the constant δ_f acts as a hard lower bound to avoid near-zero thresholds on quiet, near-flat signal segments. A frame τ_0 is selected as a peak location t_k if it is a local maximum within its temporal neighborhood and exceeds the adaptive threshold:

$$\tau_0 = \underset{\tau \in [\tau_0 - \tau_{\text{pre}},\, \tau_0 + \tau_{\text{post}}]}{\arg\max} u_v(\tau), \qquad u_v(\tau_0) > \delta, \tag{26}$$

subject to the minimum inter-peak distance constraint τ_{wait}

$$t_k - t_{k-1} \geq \tau_{\text{wait}}. \tag{27}$$

In practice, we set $\tau_{\text{wait}} = 0.5$ to ensure the significance of local peaks. The resulting visual rhythm sequence is represented as

$$\mathcal{P}_v = \left\{\left(t_k, u_v(t_k)\right)\right\}_{k=1}^{N_p}, \tag{28}$$

where t_k denotes the temporal location of a salient visual rhythm event and $u_v(t_k)$ its corresponding response strength.

G Additional Experiments

In Table 8, we further demonstrate the text-to-music generation performance of CueMuse and the base model Musicgen-LLaMA. We select 3 representative works for compareing, including

Musicgen-MediumCopet et al. (2023), Stable Audio Open (SAO) Evans et al. (2025), and Stable Audio3 Medium Evans et al. (2026).

It can be seen from the results that Musicgen-LLaMA improves the sound quality of Musicgen-Medium while maintaining a similar modality matching score. Although CueMuse is fine-tuned on video-music pairs, it still retains strong text-to-music generation capability. This demonstrates the flexibility of CueMuse: it supports both video-to-music generation and text-to-music generation, and remains competitive when compared with state-of-the-art music generation models.

In Table 10, we show the performance of CueMuse obtained by fine-tuning under different data ratios. It can be seen that the performance of our model improves steadily with the increase of data volume and has not yet reached its limit.

In Table 12, we further present the performance comparison between CueMuse and baseline models on OES-Com Kim et al. (2025). OES-Com contains 100 annotated clips from 37 independent films. We use it as a supplement to the evaluation results to better solidify our conclusions. The results show that CueMuse maintains the same outstanding performance as on Universe-Bench, which further demonstrates the generalizability of our proposed model.

In Table 11, we demonstrate our model's stability in edit mode. We apply delay perturbations to the input rhythm using a small test set of 50 samples. We perform a comprehensive evaluation with recall, precision, and F1 scores, using a tolerance of ±80 ms relative to the provided timestamps. Notably, beats can render correctly in unannotated regions, which may reduce precision. For this task, recall takes priority. Recall measures how many editing requirements the model actually satisfies. As shown in the table, our model maintains stable performance across different delays. The alignment ratio between audio quality and rhythm stays unchanged.

Table 10: Ablation studies on the ratio of training data in V2M task.

Training Data	Metrics						
	FAD ↓	FD ↓	KL ↓	Density ↑	Coverage ↑	IB ↑	CLAMP3 ↑
20%	2.5142	21.4268	1.1218	0.4758	0.1474	0.1452	0.1013
50%	2.4754	19.8647	1.0651	0.5170	0.2180	0.1521	0.1084
80%	2.2869	**18.3330**	1.0084	0.5941	0.2121	0.1544	0.1079
100%	**2.2010**	18.5054	**0.9906**	**0.6855**	**0.2282**	**0.1577**	**0.1113**

Table 11: Analysis of rhythm control stability on edit mode.

Time Disturb	Metrics				
	FAD ↓	KL ↓	Recall(@±80ms) ↑	Precision(@±80ms) ↑	F1 ↑
Origin	5.348	0.9896	0.767	0.352	0.468
+100ms	5.020	0.9848	0.697	0.357	0.452
+200ms	5.681	1.0442	0.718	0.342	0.454
+300ms	5.800	1.0185	0.724	0.332	0.448

H Details of Evaluation Metrics

Fréchet Audio Distance (FAD) is a metric similar to Fréchet Inception Distance (FID) Heusel et al. (2017), which is often used to evaluate the distribution gap between generated audio and real audio, and is widely applied in speech, music and sound effect generation models. A lower FAD typically indicates better human-perceived acoustic-level audio quality. We report the FAD based on the most commonly used pre-trained audio feature extraction model VGGishHershey et al. (2017).

Fréchet Distance (FD) also measures similarity between generated audio samples and target original samples. It resembles FAD but uses the feature extractor of PANNs Kong et al. (2020) pre-trained on AudioSet Gemmeke et al. (2017). Likewise, a smaller FD value means the distribution of generated audio is closer to that of the original audio.

Table 12: Quantitative comparison of music quality and alignment on OES-Com. Lower is better for FAD, FD, KL; higher is better for Density, Coverage, IB, CLAMP3. **Bold** denotes best, underline denotes second-best. Var. denotes that the model supports variable-length generation and is generated in full length.

Model	Var.	FAD (↓)	FD (↓)	KL (↓)	Density (↑)	Coverage (↑)	CLAMP3 (↑)
Video-Only to Music							
CMT Di et al. (2021)	×	15.3526	71.7782	1.6930	0.2422	0.0181	0.0980
M2UGen Liu et al. (2023b)	×	13.2826	59.7584	1.7454	0.3465	0.1129	**0.1034**
Video2Music Kang et al. (2024)	×	30.5021	137.2014	2.3472	0.2000	0.0028	0.0981
GVMGen Zuo et al. (2025)	✓	9.4098	**44.4089**	1.4955	0.4584	0.1584	0.0937
VidMuse Tian et al. (2025)	✓	8.1275	47.7070	**1.4557**	0.5011	0.1590	0.0964
Diff-V2M Ji et al. (2026)	✓	9.6908	45.2500	1.5180	**0.6751**	**0.1612**	0.0967
CueMuse (Ours)	✓	**8.0921**	45.1118	1.4930	0.4670	0.1511	0.1026

Kullback-Leibler Divergence (KL) is used to measure the difference between two probability distributions, and it also reflects the similarity between generated samples and reference samples. It is also calculated on the multi-label class predictor PANNs.

Density and Coverage measure the fidelity and diversity of generated audio samples, respectively. Fidelity measures how closely the generated audio samples fit the real data distribution, while diversity evaluates whether the generated samples can cover the full variation range of real sample audio. We use default setting of PANNs embeddings for manifold estimation.

ImageBind (IB) Score is used to evaluate the matching degree between generated music and videos. We only use the audio and visual branches to compute the modal similarity. Notably, as stated in Tian et al. (2025), ImageBind Girdhar et al. (2023) is not trained specifically on music data, which may limit its effectiveness in capturing the full complexity of video-music alignment. Nevertheless, it remains one of the most suitable available options for evaluating cross-modal alignment between video and audio.

CLAMP3 Score indirectly evaluates the alignment between video and music by measuring the Clamp3 similarity between the GT text and the generated music. Clamp3 Wu et al. (2025) is trained on 2.31M music-text pairs using the contrastive learning approach. A higher Clamp3 score of the model indicates that the generated music is closer to the GT music in terms of emotion, instruments, beats and other aspects.

Sync Score measures the degree of overlap between the beat timestamps of music generated by the model and the ground-truth (GT) beats. Let N_{GT} denote the total number of GT beats and N_{Gen} denote the total number of beats in the generated music. We first calculate M, the number of matched beat pairs with an 80 ms tolerance, where the matching is mutually exclusive. The Sync Score is defined as the IoU of matched beats: $Sync = \frac{M}{N_{GT}+N_{Gen}-M}$. Finally, we compute the weighted average of Sync Score weighted by the number of beats, meaning test samples with longer durations and more beats carry greater weights. This yields the overall alignment score, Sync Score. A higher Sync Score generally indicates that the model can generate music that aligns better with the GT. In practice, we use All-in-one Kim & Nam (2023) uniformly for annotation.

I Details of Evaluation Dataset

V2M Bench Tian et al. (2025) and HarmonySet-test Zhou et al. (2025) are two common datasets for evaluating the video-to-music generation task. However, we notice that V2M Bench adopts demucs for music separation, while some audio in HarmonySet-test still retains the original human vocals. The construction of V2M Bench lacks manual filtering, and the manual work for HarmonySet-test mainly focuses on video annotation rather than verifying data alignment. In addition, both datasets suffer severe category imbalance: as a dataset originating from V2M, V2M Bench mainly contains categories such as films, promotional videos, trailers and music videos, whereas HarmonySet-test places more emphasis on instrumental performances, concerts, dance clips and music video snippets. Ultimately, we performed manual filtering on both datasets. We selected 121 high-quality samples from the 300 test samples of V2M Bench. HarmonySet-test has as many as 3858 test samples. We first used Qwen-3.8-max Yang et al. (2025) to score the music-video matching degree of the original

annotations, extracted 489 samples with scores no less than 4 points out of 5, and further obtained 179 samples via manual filtering. The prompt can be found in Figure 7. Finally, to guarantee the overall category balance of the test set and ensure separation from all training data, we collected 96 newly created videos from 2026 on BiliBili. The HarmonySet-test subset and BiliBili subset uniformly use Mel-Band RoFormer for music separation. These three parts together form our test set, Universe-Bench. In Figure 5 and Figure 6, we present the detailed content of Universe-Bench, including video types, subcategories and their corresponding quantities. Meanwhile, we also show the sample music tags, music duration distribution and tempo distribution of Universe-Bench in Figure 5(b)(c)(d), respectively.

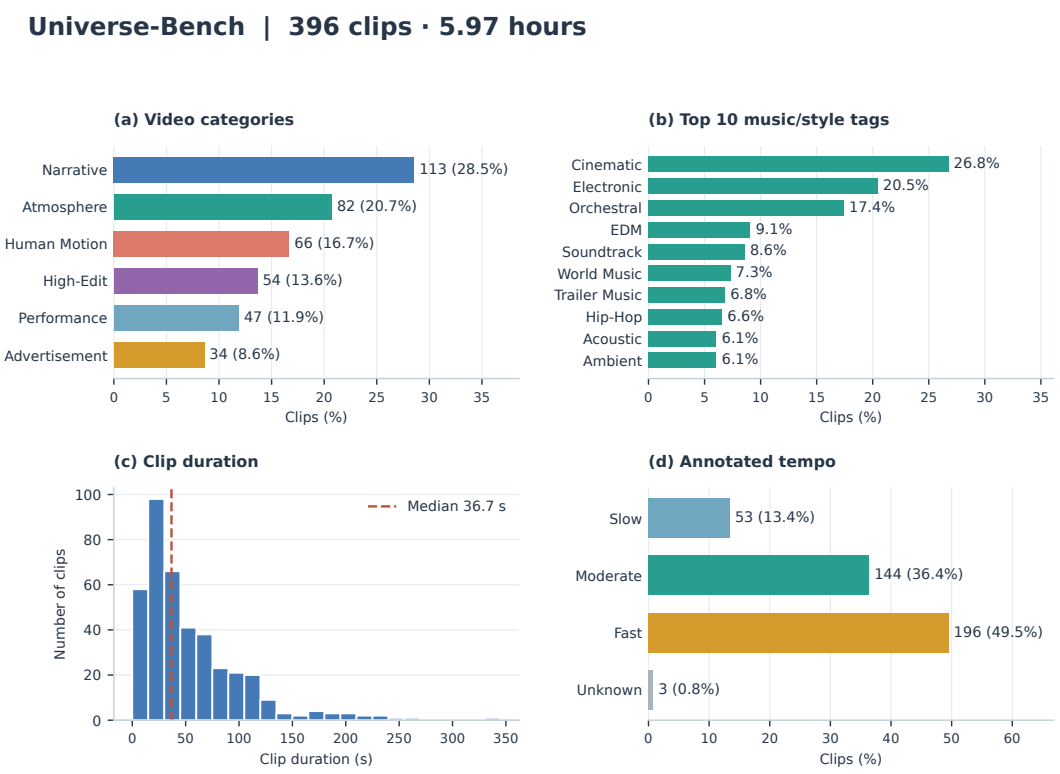

Figure 5: The overview of Universe-Bench, which is consist of 6 main classes with multiple music style, duration and tempo.

J User Study Interface

For subjective evaluation, we adopt the Mean Opinion Score (MOS) metric, scoring based on a scaled 5-point discrete Likert scale where 100 corresponds to excellent and 20 corresponds to very poor. In Figure 8, we present our subjective evaluation interface. Given five pairs of identical videos with soundtracks generated by different models, users are required to score them across four dimensions, including MOS-Q, MOS-R, MOS-M and MOS-A. Scores ranging from 20 to 100 indicate increasing satisfaction with the corresponding metrics. On the left, we provide explanations for the four evaluation dimensions and the expectations associated with each score. We sampled 100 videos from Universe-Bench with balanced main categories. All videos are shuffled, and 40 clips are randomly selected for distribution to users. To reduce evaluation complexity, videos longer than 30 seconds are truncated.

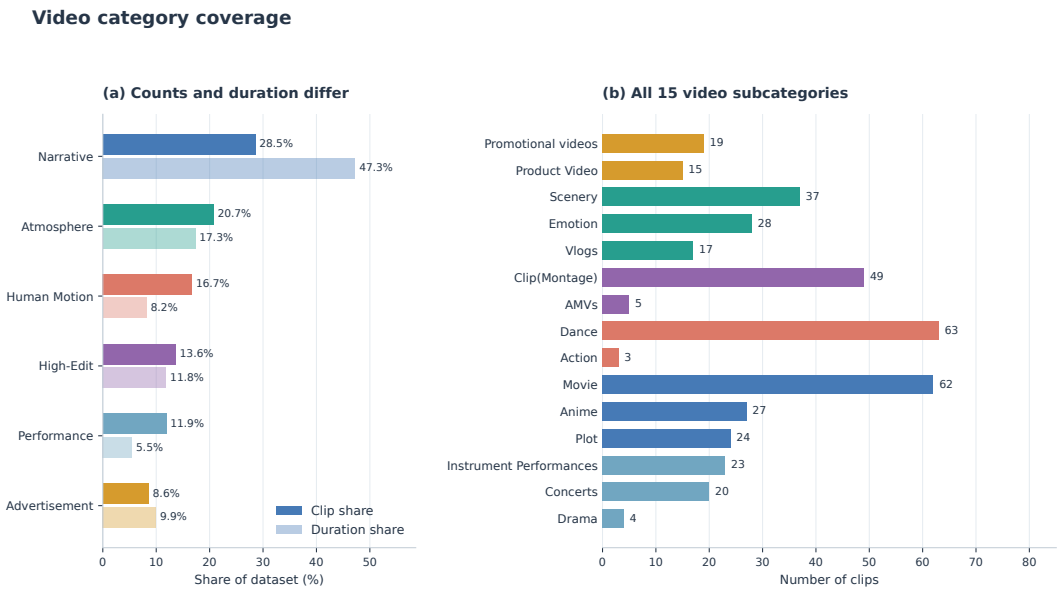

Figure 6: Universe-Bench contains 15 video subcategories among 6 main classes. We also display the duration differ between them.

Prompt for Music-Video Filter

Core Task
You are an evaluator for music video dataset. Read the following four-dimension assessment of a music video (rhythm, theme, emotion, culture). Rate the audio-visual matching quality from 1(worst) to 5(best). Only output the numeric score.
Assessment:

Figure 7: Prompt for Music-Video Filter.

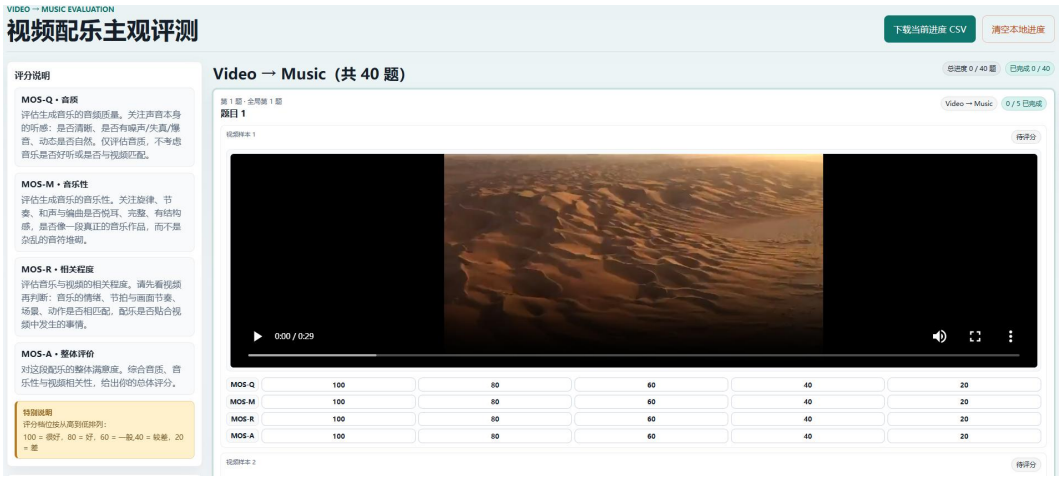

Figure 8: Screenshot of subject evaluation tool.