V2M-ZERO: Zero-Pair Time-Aligned Video-to-Music Generation

Yan-Bo Lin* UNC Chapel Hill
Jonah Casebeer Adobe Research
Long Mai Adobe Research
Aniruddha Mahapatra Adobe Research
Gedas Bertasius UNC Chapel Hill
Nicholas J. Bryan Adobe Research

Abstract

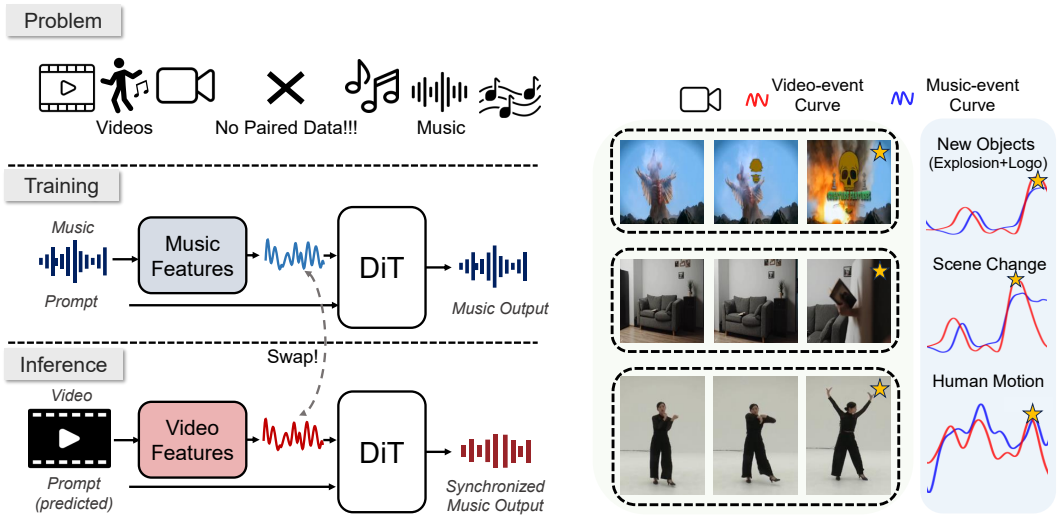
Generating music that temporally aligns with video events is challenging for existing text-to-music models, which lack fine-grained temporal control. We introduce V2M-ZERO, a video-to-music generation approach that generates time-aligned music with disentangled time synchronization and semantic control (e.g. genre, mood) from video while requiring zero video-music pairs at training time. Our method is motivated by a key observation: temporal synchronization requires matching when and how much change occurs, not what changes. While musical and visual events differ semantically, they exhibit shared temporal structure that can be captured independently within each modality. We capture this structure through event curves computed from intra-modal similarity using pretrained music and video encoders. By measuring temporal change within each modality independently, these curves provide comparable representations across modalities. This enables a simple training strategy: fine-tune a text-to-music model on music-event curves, then substitute video-event curves at inference without cross-modal training or paired data. Across OES-Pub, MovieGenBench-Music, and AIST++, V2M-ZERO achieves state-of-the-art performance *without any paired music-video data*, surpassing the strongest prior baselines per metric with 5–9% higher audio quality, 13–15% better semantic alignment, 21–52% improved temporal synchronization, and 28% higher beat alignment on dance videos. We find similar results via a large crowd-source subjective listening test. Our results validate that temporal alignment through within-modality features is not only effective for video-to-music generation but leads to *better* performance over paired cross-modal supervision. Furthermore, our approach enables independent controls for timing and music style (e.g., genre, mood) for more controllable generation. Results are available at https://genjib.github.io/project_page/v2m_zero/

1 Introduction

Generative music is growing in popularity among creators from online influencers on social media platforms (e.g., YouTube, Instagram, TikTok) to professionals in film, gaming, and advertising. Such content creators seek music that both complements their video content and supports separate control over style and time synchronization. While recent text-to-music (T2M) methods [109, 1, 16, 43, 77, 118, 85, 25, 60, 22, 66] enable automatic music generation from textual prompts, they operate solely on text inputs and typically are not designed to generate time-synchronized music that matches a target video. As a result, creators have to manually and tediously edit videos to fit

*Work done during an internship at Adobe Research.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) **Zero-pair training & inference. Top**: Paired methods require large paired video-music datasets. **Middle**: V2M-ZERO trains with music-event curve conditioning and does not use paired music and video data. **Bottom**: At inference, we *swap* music-event curves with video-event curves to generate time-synchronized music.

(b) **Shared temporal structure across modalities.** Real event curves from video and music share similar temporal patterns. Ground-truth pairs have correlation $\approx$ **0.6**; random temporal offsets degrade this to $\approx$ **0.2**.[1]

Figure 1: **Why zero-pair video-to-music generation is possible.** V2M-ZERO is trained with music-event curves and, at inference, replaces them with video-event curves. This transfer is enabled by the shared temporal structure of event curves across music and video.

the generated music for synchronization, a laborious and time-consuming process. For instance, in a short product promotion video, musical cues must align with each reveal or motion highlight to create a strong and memorable impression within seconds.

The lack of time-synchronized text-to-music generation motivates video-to-music (V2M) models [52, 91, 104, 92, 59, 38, 119, 101, 50, 47, 13], or the task of creating background music that is both temporally and semantically aligned with a given video. Existing V2M methods typically rely on paired video–music datasets curated from online videos, entangling these two aspects of control. Recent work [111, 28] explores an alternative direction by using pretrained multimodal large language models (MLLMs) to infer music prompts for video, which are then used as input to a pretrained T2M model. While promising for semantic alignment, such methods do not explicitly model temporal correspondence between visual and musical events, limiting user control. Furthermore, we note that it is now common to require licensed data for training music models [107] and licensing *paired* music and video is significantly more complicated than licensing music and silent video separately [15, 75]. This motivates our goal: time-synchronized video-to-music generation with independent timing and semantic controls, trained without paired video-music data.

We realize this with V2M-ZERO– a video-to-music generation method that generates ***time-synchronized*** music from videos with separate timing and semantic controls – all ***without*** paired video–music data (zero-pair). Our key observation is that synchronization primarily depends on ***when*** change occurs, rather than ***what*** changes. In practice, video and music synchronization often corresponds to (sparse) moments of interest or events over time (e.g., video events of dancing and scene transitions match music events like beats, instrumental and/or dynamic changes). Concretely, we represent events over time as event ***curves*** computed from intra-modal similarity in encoder feature spaces. The result is a representation that captures perceptually relevant change across video and music (Figure 1b). Crucially, our design decouples temporal structure from semantic and emotional grounding: event curves specify ***when*** musical change should occur, while textual conditioning determines ***how*** it should sound. The two are highly expressive in conjunction, though neither is sufficient alone. To do so, we lightly fine-tune a pretrained T2M model with added music event

[1] Correlation w/1-sec window around video scene cuts. Correlation degrades as music is time-shifted relative to video.

curves, enabling test-time transfer by simply substituting video event curves at inference. For text conditioning, we extract visual captions from video and summarize them with an LLM to capture the musical style and mood [28], achieving robust adaptation to diverse and complex scenes.[2]

Our contributions are threefold. First, we propose V2M-ZERO, the first zero-pair framework for time-synchronized video-to-music generation. Our key insight is that event curves from intra-modal similarity yield structurally comparable representations, enabling test-time transfer from music to video conditioning with only lightweight fine-tuning (192-768 GPU hours) of a pretrained model. Second, through extensive ablations, we show that V2M-ZERO is robust across video encoders rather than tied to a specific backbone, and we analyze design choices that mitigate the modality gap between music and video representations. Finally, we demonstrate state-of-the-art results across three diverse datasets. Our zero-pair approach surpasses the strongest prior baselines per metric on all metrics across all datasets: **5–9%** better in music quality, **13–15%** in semantic alignment, **21–52%** in temporal synchronization, and **28%** in beat alignment on dance videos. A large-scale crowdsourced listening test corroborates these gains.

2 Related work

2.1 Text-to-music generation

General text-to-audio generation has advanced significantly in the past few years [55, 96, 53, 54, 4, 32, 106, 20, 88, 87, 21, 39, 65, 76]. Text-to-music generation has followed a similar trend [109, 93, 1, 16, 43, 77, 118, 85, 25, 60, 22, 66]. Both music and audio generation methods typically fall into two modeling paradigms. The first, autoregressive (AR) modeling [16, 1, 39, 118, 43], operates on discrete audio tokens produced by a neural audio codec [110, 17, 40]. A causal transformer predicts each token conditioned on the previously generated tokens and the input text prompt, and the resulting token sequence is decoded back to a waveform by the codec. The second, latent diffusion models (LDMs) [22, 53, 54, 24, 64, 66], learns a denoising process over continuous latents conditioned on text. These employ a similar kind of neural audio codec with continuous latents [21]. While recent text-to-music models [1, 16, 43, 77, 118, 85, 25, 60, 22] can effectively capture high-level semantics, such as genre, mood, and instrumentation, it remains challenging to align music with fine-grained visual events and/or separate timing and semantic control.

2.2 Video-to-music generation

Building on video-to-audio research [19, 11, 41, 10, 63, 61, 69, 8, 112, 9, 98, 58, 83, 12], recent video-to-music methods generate soundtracks aligned with visual rhythm and motions (i.e., dance videos) [44, 116, 81, 114, 115, 108, 48, 84]. Early approaches relied on symbolic data such as MIDI or ABC notation [117, 35, 18, 90, 49], but these datasets were limited in scale and expressivity. Recent works train on paired internet videos accompanied by music tracks [52, 34, 91, 92, 59, 38, 119, 101, 50, 47, 13, 56, 113, 82, 94, 73, 27]. Such internet data is often noisy, containing imperfect mixing, and/or potential copyright issues. These limitations motivate the exploration of unpaired or zero-shot paradigms that leverage independent, unpaired music and video sources.

2.3 Video-to-music generation via prompting

Without paired video-music training data, recent methods [97, 45, 79, 33, 111, 104] bridge the video and music domains through prompts. Specifically, they translate visual content into text prompts (via LLM reasoning or learned prediction) and then generate music using off-the-shelf text-to-music models. While this strategy effectively captures high-level semantics (e.g., genre, mood), it struggles with fine-grained temporal alignment because such prompts lack the expressiveness to specify timing and dynamics. In contrast, V2M-ZERO directly conditions on music or video event curves, enabling temporal synchronization with scene cuts, motion patterns, and dance rhythm.

[2] Demos with fixed event-curves but varying text inputs are in the supp. material.

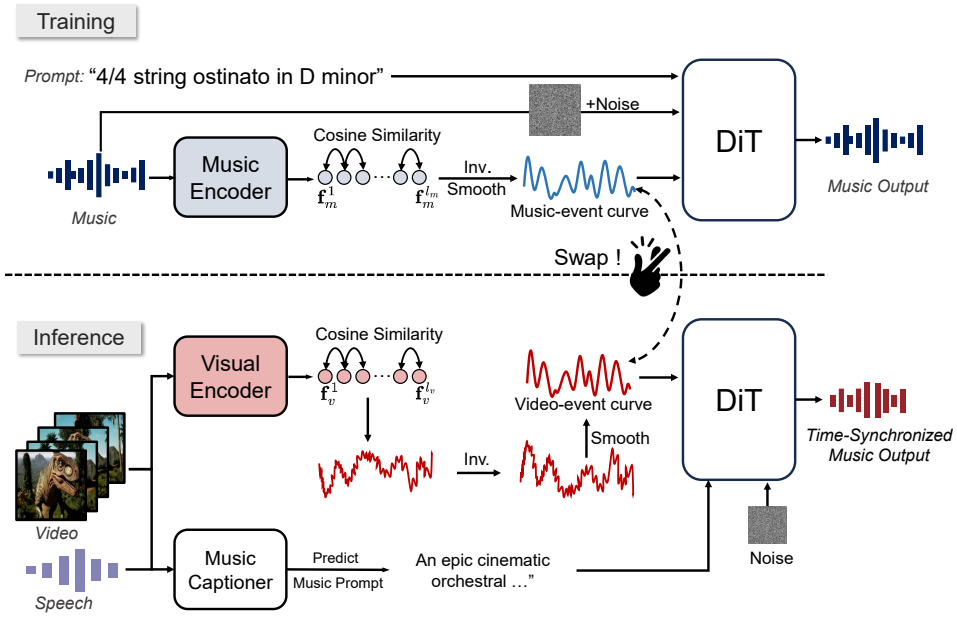


Figure 2: **Method overview. Top**: During training, V2M-ZERO learns a rectified-flow diffusion process conditioned on text prompts and a *music-event curve* derived from intra-music similarity. **Bottom**: At inference, music conditioning is *swapped* with a *video-event curve* based on framewise similarity, enabling zero-pair, time-synchronized video-to-music generation. For semantic alignment, a text prompt is predicted from video and speech via Vibe [28], which first summarizes video content and recommends a matching music text prompt.

3 Technical approach

We fine-tune a pretrained T2M latent rectified flow model by augmenting it with temporal event curve conditioning, enabling test-time transfer to video at inference. The curves contribute timing, while semantic content (energy, genre) comes from text prompts. Our insight is that temporal change, measured via intra-modal similarity in the feature space of strong encoders, captures perceptually meaningful domain-agnostic event structure. Events such as beat onsets in music or scene cuts in video both manifest as local dissimilarity between consecutive temporal segments. By standardizing these dissimilarity signals per-sample, we obtain structurally comparable representations across modalities. During training, the model is conditioned on text prompts and music-event curves. At inference, we replace music-event with video-event curves for zero-paired, time-synchronized generation without any model retraining. This alleviates the need for paired video-music data while generating temporally aligned outputs. An overview is shown in Figure 2.

3.1 Preliminaries

Music autoencoder. We use a pretrained music autoencoder [7] to encode stereo music waveforms into continuous latents $\mathbf{x}_0 \in \mathbb{R}^{d \times l}$, where d is the latent dimension and l is the temporal length. These latents represent the low-level representation used by our rectified flow model. Separately, we extract high-level semantic features from other pretrained encoders to compute event curves.

Rectified flows. Given a text condition $\mathbf{c}$ and a music latent $\mathbf{x}_0$ sampled from the empirical data distribution $\mathcal{P}_D$, a rectified flow model learns a velocity field $f_\theta(\cdots)$ that transports samples from a Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to $\mathcal{P}_D$. The timestep variable $t \in [0, 1]$ interpolates between noise ($t = 1$) and data ($t = 0$) via the linear path $\mathbf{x}_t = t\boldsymbol{\epsilon} + (1 - t)\mathbf{x}_0$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We train $f_\theta(\cdots)$ to predict the velocity $\boldsymbol{\epsilon} - \mathbf{x}_0$ via:

$$\min_\theta \ \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t, \mathbf{c}} \, \| \, (\boldsymbol{\epsilon} - \mathbf{x}_0) - f_\theta(\mathbf{x}_t, \mathbf{c}, t) \, \|_2^2 \, . \tag{1}$$

We generate samples by solving the ODE $d\mathbf{x}_t = f_\theta(\mathbf{x}_t, \mathbf{c}, t)\, dt$ from $t = 1$ to $t = 0$ with 96 sampling steps and classifier-free guidance [30].

Model architecture. We use a pretrained Diffusion Transformer (DiT) architecture [70] $f_\theta(\cdots)$ with cross-attention text conditioning $\mathbf{c}$ following standard T2M models [22, 53, 54, 24, 64, 21].

3.2 Temporal event curves

We construct event curves that capture when and how much meaningful change occurs (sometimes denoted as novelty curves). Our procedure is identical for music and video, enabling direct substitution at inference.

Perspectives on event curves. Our event curve formulation connects to established uses of self-similarity across modalities. In music, self-similarity has been widely used to analyze musical structure and rhythm [23, 71] and is also considered a generalization of the concept of autocorrelation. In video, similar approaches segment footage and detect natural boundaries such as shot transitions [80, 42] Mathematically, our approach is equivalent to extracting one or more off-diagonal bands of the self-similarity matrix. In that sense, our formulation leverages the natural structure of self-similarity to enable zero-shot transfer.

Features for event curves. To compute event curves, we begin by extracting temporal feature sequences using pretrained encoders. For music (training only), we apply a music encoder to obtain $\mathbf{f}_m \in \mathbb{R}^{d_m \times l_m}$, where l_m is the temporal length and d_m is the feature dimension. For video (inference only), we encode each frame with a visual encoder and spatially pool to obtain $\mathbf{f}_v \in \mathbb{R}^{d_v \times l_v}$, where l_v is the number of frames and d_v is the feature dimension. These high-level semantic features are used solely to compute event curves.

Computing event curves. Given a feature sequence $\mathbf{f} \in \mathbb{R}^{d_f \times l_f}$ (either $\mathbf{f}_m$ or $\mathbf{f}_v$), where $\mathbf{f}^k$ denotes the $k-th$ temporal feature-vector, we measure temporal change via the cosine similarity between consecutive vectors:

$$s^k = \frac{\mathbf{f}^k \cdot \mathbf{f}^{k+1}}{\|\mathbf{f}^k\| \, \|\mathbf{f}^{k+1}\|}, \quad k = 1, \ldots, l_f - 1. \tag{2}$$

We obtain a dissimilarity sequence via $a^k = 1 - s^k$ and set $\mathbf{A} = \{a^k\}$. Higher values indicate stronger temporal change in high-level encoder spaces, including beat onsets, scene cuts, motion, and musical-structure changes. We provide preliminary analysis of more elaborate self-similarity schemes in Appendix D, Tab. 8.

Mitigating the modality gap. To enable zero-shot transfer from music to video, we apply three operations that align event curves across modalities. First, we standardize each datum $\mathbf{A}$ to have zero mean and unit variance:

$$\bar{a}^k = \frac{a^k - \mu(\mathbf{A})}{\sigma(\mathbf{A})}. \tag{3}$$

Without standardization, music and video event curves have different scales and offsets, creating a distribution shift when swapping at inference. This intentionally makes the curve a relative temporal-change signal. Absolute energy, mood, and instrumentation are instead controlled by the text prompt. Second, we resample to length l (matching the temporal dimension of $\mathbf{x}_0$). Third, we apply temporal smoothing (e.g. via a Hann window) to suppress modality-specific details while preserving the larger structure, yielding the final event curve:

$$\mathbf{e} = \text{Smooth}\big(\text{Resample}\big(\bar{\mathbf{A}}, l\big)\big) \in \mathbb{R}^l. \tag{4}$$

During training, we compute $\mathbf{e}_m$ from music features. At inference, we compute $\mathbf{e}_v$ from video features. Our design is agnostic to the choice of feature encoders. We use MusicFM [100] and DINOv2 [68] by default but explore alternative audio and video encoders in Section 5.3.

3.3 Rectified flow model fine-tuning

We incorporate event curves into a pretrained text-conditioned rectified flow model via fine-tuning.

Event curve conditioning via concatenation. We inject the event curve $\mathbf{e}$ by concatenating it as an additional channel to the rectified flow latent:

$$\widetilde{\mathbf{x}}_t = \big[\, \mathbf{x}_t \,,\, \mathbf{e} \,\big] \in \mathbb{R}^{(d+1)\times l}, \tag{5}$$

where $[\cdot, \cdot]$ denotes channel-wise concatenation. This approach is simple and parameter-efficient, only requiring additional parameters in the input projection layer of the DiT. The same concatenation approach also supports event curves computed from different temporal scales, as explored in Appendix D.

Fine-tuning objective. We initialize from a pretrained T2M model, add the conditioning signal, and fine-tune it with music-event curve conditioning via:

$$\min_{\theta} \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t, \mathbf{e}_m, \mathbf{c}} \left\| (\boldsymbol{\epsilon} - \mathbf{x}_0) - f_\theta(\widetilde{\mathbf{x}}_t, \mathbf{c}, t) \right\|_2^2, \tag{6}$$

where we explicitly denote $\mathbf{e}_m$ to emphasize that during training, event curves are computed from music. Given a music clip during training, we extract music encoder features $\mathbf{f}_m$ and compute the music event curve $\mathbf{e}_m$ via the procedure in Section 3.2. The text condition $\mathbf{c}$ is obtained from ground-truth music descriptions in the pre-training data. Fine-tuning allows our model to learn to generate music conditioned on both text and temporal events.

3.4 Zero-pair inference via event curve swapping

At inference, we perform test-time transfer to video-to-music generation by swapping in video event curve conditioning signals, while keeping all model weights fixed. Given an input video, we extract visual encoder features $\mathbf{f}_v$ and compute the video event curve $\mathbf{e}_v$ using the same procedure as $\mathbf{e}_m$. For text conditioning, we use an LLM [28] to generate and summarize video captions into a music-appropriate text prompt $\mathbf{c}$. We then generate music using standard flow-model inference and add the generated audio to the input video. Note, our music and video event curves are designed to mitigate the domain gap, so this swapping does not require retraining.

3.5 Implementation details

Architecture. We adopt an internal, pretrained T2M model similar to [67, 3] for fine-tuning. The audio autoencoder is similar to [21, 5, 3, 6] and compresses stereo 44.1 kHz waveforms into continuous latents (d=64) at 12.3 Hz, yielding 394 frames for 32-second clips. The rectified flow model uses the DiT architecture with $\approx$ 1 billion parameters: 16 layers, hidden dimension 2048, and feed-forward dimension 8192. Text conditioning is implemented via cross-attention layers using Gemma-4B [89] embeddings.

Feature encoder. We use MusicFM [100] and DINOv2-L [68] to extract music features $\mathbf{f}_m$ and video features $\mathbf{f}_v$ by default. For dance videos, we report results with CoTracker [36], while the default DINOv2 setup already exceeds prior AIST++ baselines. We explore a comprehensive ablation of encoder choices, including an audio autoencoder, AVSiam [51], and V-JEPA [2] in Section 5.3.

Training. We initialize from a pretrained text-to-music model, adding 2048 parameters to input and project our event curve conditioning, and fine-tune on $\approx$ 25k hours of licensed instrumental music-text pairs. Fine-tuning is lightweight, requiring only 2–4 days on 4–8 A100 GPUs using 32-second clips (192-768 GPU hours). We use the AdamW optimizer with a learning rate 10^{-4} and apply classifier-free guidance with 10% condition dropout. Cumulatively, this fine-tuning scheme enables video-to-music generation without paired video-music data or training from scratch, given a pretrained model.

4 Experimental setup

We use three diverse benchmark datasets, multiple baselines, and a standard protocol for evaluation.

4.1 Evaluation data

We test on datasets spanning general, cinematic, and dance videos:

- **OES-Pub** [38] is the public evaluation split of the Open Screen Soundtrack Library (OSSL), containing 115 public-domain movie clips paired with royalty-free music. Each clip is $\approx$ 30 seconds and includes human-annotated music prompts.

Table 1: **Comparison with state of the art.** We evaluate all methods [57, 119, 92, 91, 101, 111] on OES-Pub [38] and MovieGenBench-Music [72] in audio quality (FAD), high-level semantic alignment (CLAP), and our proposed temporal metric (SCH). V2M-ZERO outperforms all competing methods across all datasets and metrics. Note, OES-Pub reference data includes non-musical sounds, so we use SongDescriber [62] for the reference distribution for FAD results.

			OES-Pub.			MovieGenBench-Music		
			Audio Quality	High-Level Alignment	Low-Level Matching	Audio Quality	High-Level Alignment	Low-Level Matching
Method	**V–M Pairs**		**FAD*↓**	**CLAP↑**	**SCH↑**	**FAD↓**	**CLAP↑**	**SCH↑**
M^2UGen [57]	✔	36.7h	6.67	0.07	0.35	5.84	0.02	0.24
GVMGen [119]	✔	147h	6.25	0.18	0.35	3.96	0.06	0.48
MTCV2M [101]	✔	147h	5.44	0.20	0.37	4.02	0.16	0.29
VidMuse [92]	✔	18000h	10.4	0.16	0.40	2.98	0.04	0.47
AudioX [91]	✔	15793h	7.46	0.19	0.33	2.82	0.08	0.30
SONIQUE [111]	✘	0h	6.80	0.09	0.27	6.47	0.16	0.21
V2M-ZERO (Ours)	✘	0h	**4.95**	**0.23**	**0.61**	**2.68**	**0.18**	**0.58**

- **MovieGenBench-Music** [72] is the music subset of MovieGenBench, consisting of 527 video-music pairs with sound effects across diverse video types. Each clip is $\approx$ 10 seconds and includes a music prompt.
- **AIST++** [46, 95] is a street dance dataset with copyright-cleared dance music, consisting of 20 video–music pairs across 10 dance genres. Each clip is $\approx$ 7 seconds and includes BPM.

Collectively, these benchmarks cover the principal video categories evaluated in prior video-to-music work: natural, cinematic, and dance.

4.2 Evaluation metrics

Following [92, 91, 52, 119], we measure: audio fidelity, semantic alignment, and temporal synchronization. For each dataset under test, we customize the metrics used to highlight the critical aspects of the specific video-to-music generation task.

OES-Pub & MovieGenBench-Music:

- **Fréchet audio distance (FAD)** [37]: audio fidelity via distributional distance between reference and generated music in VGGish [29] space. We denote the use of a separate reference set [62] via *. Lower is better.
- **CLAP score** [103]: semantic alignment via cosine similarity between generated music and text prompts in CLAP space. Higher is better.
- **Scene cut hit (SCH)**[3]: aperiodic temporal alignment. A *hit* occurs when a beat falls within ± 100 ms of a scene cut, computed as hits/total cuts. Inspired by [52]. Higher is better.
- **Human evaluation**: measures subjective preference for music quality and video synchronization. Given two music tracks for the same video, raters answer: 1) *Which has better quality?* and 2) *Which better synchronizes with visuals?* We collected 1403 valid votes from the crowdsource platform Appen, evenly sampled across datasets and randomly sampled across baselines We report win-rates with pairwise t-tests vs. [57, 119, 92, 91, 111, 101]. Higher is better.

AIST++:

- **Beat coverage (BCS), beat hit score (BHS), and F1**: periodic rhythm alignment [48]. BCS and BHS are recall and precision of music beats relative to motion beats, respectively, and F1 is their harmonic mean. We use a temporal tolerance of 0.2 seconds for beat matching. Higher is better.
- **Tempo difference (TD)**: mean absolute difference in beats per minute (BPM) between generated and ground-truth music [48]. Lower is better.

[3] Pseudocode in appendix.

Table 2: **Human evaluation.** Pairwise win rates of our model against each baseline (1403 ratings, Bonferroni-corrected t-tests, $p < 0.0083$). † above chance but not significant after correction. Scene cut subset comprises 67% of all votes.

(a) All Videos

	Win Rate of Ours ↑	
Baseline	**Music Quality**	**Temp. Align.**
M^{2}UGen	67.45%	66.51%
GVMGen	67.00%	67.00%
MTCV2M	59.72%	59.72%
VidMuse	68.85%	53.77%†
AudioX	72.36%	60.36%
SONIQUE	77.16%	73.60%
Average	**68.76%**	**63.49%**

(b) Scene Cut Videos

	Win Rate of Ours ↑	
Baseline	**Music Quality**	**Temp. Align.**
M^{2}UGen	67.88%	66.42%
GVMGen	70.29%	67.39%
MTCV2M	65.35%	66.14%
VidMuse	74.63%	59.51%
AudioX	73.17%	61.46%
SONIQUE	81.54%	78.46%
Average	**71.14%**	**66.56%**

5 Results and analysis

5.1 Comparison with state of the art

V2M-ZERO demonstrates strong performance across all benchmarks, outperforming both paired/unpaired baselines despite using no paired video-music training data. We analyze results across audio fidelity, semantic alignment, and temporal synchronization, highlighting key insights on cross-domain generalization and the utility of event-curves.

General & cinematic video to music generation. Table 1 shows results on OES-Pub and MovieGenBench-Music with DINOv2 [68] as the visual encoder. V2M-ZERO achieves the best audio quality on both benchmarks (i.e., FAD*: 4.95 on OES-Pub; FAD: 2.68 on MovieGenBench). Notably, our method exhibits the most consistent performance across datasets, while paired baselines show substantial variation (e.g., VidMuse: 10.4 $\rightarrow$ 2.98). Interestingly, VidMuse, trained on 18k hours of paired data, achieves the worst FAD on OES-Pub but the best among paired methods on MovieGenBench, suggesting dataset-specific overfitting rather than robust generalization. We compute FAD* on OES-Pub using SongDescriber [62] as the reference distribution, since the original OES-Pub audio contains speech, sound effects, and background noise. MovieGenBench uses generated music references, which generally yield lower absolute FAD scores than real recordings. For semantic alignment (CLAP), paired-data baselines perform reasonably on OES-Pub but exhibit noticeable degradation on MovieGenBench-Music. For instance, AudioX achieves 0.08 on MovieGenBench-Music, compared to 0.19 on OES-Pub. This suggests limited generalization to MovieGenBench's generated videos, which differ from the real paired data used during training. Interestingly, SONIQUE is the only baseline that improves on MovieGenBench (i.e., 0.09 to 0.16), likely because its pure-text approach is less sensitive to domain shift in the visual content. In contrast, V2M-ZERO demonstrates robust performance across all datasets (0.23 on OES-Pub, 0.18 on MovieGenBench-Music), showing cross-domain generalization with better CLAP scores.

For temporal alignment (SCH), methods trained on paired video-music data achieve strong performance, ranging from 0.33–0.40 on OES-Pub and 0.24–0.48 on MovieGenBench-Music. In comparison, SONIQUE, which uses a pure LLM-based prompting approach, achieves lower temporal alignment (i.e., 0.27 on OES-Pub, 0.21 on MovieGenBench-Music). This demonstrates the benefit of learning from paired data for capturing temporal structure. However, V2M-ZERO achieves strong temporal alignment (i.e., 0.61 on OES-Pub, 0.58 on MovieGenBench-Music), showing that explicit event-curve conditioning can surpass paired supervision for fine-grained synchronization.

Human evaluation. We analyze our 1403 pairwise human ratings across six baselines and two questions via multiple pairwise t-tests with Bonferroni correction [31]. All win-rates reported in Table 2 show our win-rate (higher is better). We show results across all videos in Table 2a and videos with a scene cut in Table 2b. Our method is consistently preferred when compared head-to-head in win-rate against each alternative. On general videos, our approach has a statistically significant win-rate over all baselines in music quality. We see a similar trend in temporal alignment, where our method is best in 6/6 cases and the lead is significant in 5/6. To understand when and why V2M-

Table 3: **Generalization across video types and model implementations.** (**a**) Eval. on [95, 46] using beat consistency (BCS, BHS, F1) and tempo difference (TD). (**b**) V2M-ZERO on a public text-to-music model evaluated on OES-Pub.

(a) Dance Video Results on AIST++.

Method	BCS↑	BHS↑	F1↑	TD↓
CMT [18]	0.3368	0.1515	0.2090	21.74
CDCD [115]	0.4233	0.2151	0.2852	19.25
LORIS [108]	0.3721	0.3371	0.3537	17.80
MDM [86]	0.3798	0.4185	0.3982	22.96
Text Inv. [48]	0.4761	0.4398	0.4572	20.34
V2M-ZERO (Ours)	**0.5818**	**0.6274**	**0.5856**	**12.24**

(b) Cross-architecture generalization (OES-Pub).

Method	FAD*↓	CLAP↑	SCH↑
SD-Audio-ctrl [14]	4.86	**0.18**	0.28
+V2M-ZERO(Ours)	**4.13**	0.17	**0.38**

ZERO is preferred, we test scenes with a scene-cut in Table 2b, a clear event for raters to anchor their evaluation on. Average win-rate increases and is significant in all 12/12 cases. V2M-Zero is preferred over paired-data baselines and this preference increases with an anchoring visual event.

5.2 Generalization across video types and models

Dance video to music generation. To evaluate on content with tightly coupled temporal dynamics, we test on AIST++ [95, 46], a dance dataset where video and music are highly related. Table 3a shows results comparing against methods explicitly designed or trained for dance-to-music generation. V2M-ZERO achieves strong performance across all rhythm metrics (0.5818/0.6274/0.5856/12.24), outperforming specialized methods. Improvements are most pronounced for BHS and TD, suggesting that event curve conditioning achieves precise and accurate motion-rhythm correspondence. By decoupling event-curve extraction from music generation, the same model can be used on diverse video domains without retraining.

Generalization to public text-to-music models. To confirm the generality of V2M-ZERO , we evaluate with a pre-trained open text-to-music model, Stable-Audio-ControlNet [14]. This model is trained with audio RMS curves, which we swap for video-event curves at inference time. Table 3b has comparable FAD* and CLAP, but improved SCH (0.28 to 0.38). The improvement in temporal alignment, via zero-shot integration, shows that event-curve conditioning is model-agnostic.

5.3 Ablation studies

We systematically ablate three design axes: (i) kernel size for modality gap mitigation, (ii) encoders for event-curve extraction, and (iii) LLM selection for prompt generation.

Mitigating modality gap. Music-event curves (training) and video-event curves (inference) differ in temporal granularity, creating a modality gap. We mitigate this via Hann-window smoothing applied to both modalities. We sweep kernel size on OES-Pub with the MusicFM and DINOv2 encoders, then fix this value for all other datasets and encoders without re-tuning. In Figure 3 we find that increasing the kernel from 9 (.7 seconds) to 63 (5 seconds) improves FAD: 8.17 to 3.12. However, excessive smoothing blurs fine-grained events, causing SCH to degrade: 0.61 to 0.27. This reveals a trade-off. Stronger smoothing improves audio quality but weakens temporal alignment. This matches intuition as music is not frame-by-frame synchronized to video unless for comedic effect, known as Mickey Mousing [99]. We pick kernel size 31 to balance the two.

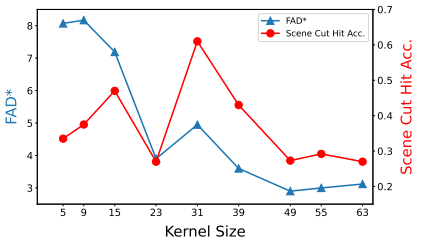


Figure 3: **Smoothing kernel size sweep on OES-Pub with MusicFM and DINOv2:** audio quality (FAD*) improves with larger kernels; synchronization (SCH) peaks at intermediate sizes. We hyperparameter tune the kernel size and apply a standard (31) across all other datasets and encoders.

Encoder architecture for event curves. Table 4a compares three encoder design paradigms: shared, reconstruction, and self-supervised encoders.

Table 4: **(a) Encoder selection for event curves.** Impact of music (training) and visual (inference) encoders on OES-Pub. MusicFM + DINOv2 achieves the best balance. **(b) LLM selection for music prompts.** Impact of LLM choice for music prompt generation on OES-Pub. V2M-ZERO uses Vibe [28] based on Gemma-4B [89].

(a)

Music Enc. (Training)	Visual Enc. (Inference)	FAD*↓	CLAP↑	SCH↑
AVSiam [51]		**4.52**	0.19	0.35
VAE [7]	V-JEPA 2 [2]	5.13	0.18	0.41
VAE [7]	DINOv2 [68]	4.77	0.16	0.31
MusicFM [100]	V-JEPA 2 [2]	5.02	0.18	0.48
MusicFM [100]	DINOv2 [68]	4.95	**0.23**	**0.61**

(b)

Music Captioner	FAD*↓	CLAP↑	SCH↑
Qwen3-4B [105]	4.98	**0.23**	0.58
Llama-3.2-3B [26]	5.02	0.21	0.60
Gemma-4B [89]	**4.95**	**0.23**	**0.61**

Shared encoders. We test whether shared-weight audio-visual encoders, like AVSiam [51], can reduce the modality gap by embedding both modalities in a common space. AVSiam achieves the best FAD (4.52) among all configurations, suggesting improved distribution matching. However, temporal alignment degrades significantly (SCH: 0.61 → 0.35), likely due to AVSiam's lower specialized capacities in both the audio and visual domains compared to foundation models like DINOv2 [68] and a fundamental difference on how music and audio are aesthetically matched to music. We hypothesize that large-scale shared encoders could close this gap as a promising future direction.

Music encoder. The music encoder has the strongest impact on performance. We initially use the same encoder as the latent diffusion VAE, for simplicity. Using a self-supervised model like MusicFM [100] improves temporal synchronization (SCH: 0.31 → 0.61, with DINOv2), while slightly reducing audio quality (FAD: 4.77 → 4.95). MusicFM's more semantic representations better correlate with visual event patterns, enabling more reliable curve alignment during inference.

Video encoder. We compare self-supervised visual encoders: V-JEPA [2] and DINOv2 [68]. DINOv2 yields slightly better audio quality across music encoders (FAD: 4.77 vs. 5.13 with VAE; 4.95 vs. 5.02) with MusicFM [100]). However, alignment results are mixed: V-JEPA outperforms with VAE (SCH: 0.41 vs. 0.31), while DINOv2 pairs best with MusicFM (SCH: 0.61 vs. 0.48). Overall, video encoder choice has minimal impact; alignment quality depends more critically on the music encoder. We use MusicFM + DINOv2 as our default configuration.

LLM selection for music prompt generation. Table 4b compares three LLMs for generating music prompts from video: Gemma-4B [89], Qwen3-4B [105], and Llama-3.2-3B [26]. Event curves are fixed, and we ablate the prompts. Differences across LLMs are minimal: FAD, CLAP, and SCH vary by less than 5%. Gemma-4B is marginally better (FAD: 4.95, CLAP: 0.23, SCH: 0.61). We conclude that LLM choice has a negligible impact and any modern LLM suffices for semantic guidance.

6 Conclusions

We introduce V2M-ZERO, a video-to-music generation approach that produces time-aligned music with disentangled time synchronization and semantic control from video while requiring zero video-music pairs at training time. Our observation is that while musical and visual events differ semantically, they exhibit similar temporal patterns when embedded by pretrained music and video encoders. We exploit this property via event curves, signals computed from consecutive frame/segment dissimilarity within each modality. We then fine-tune text-to-music models on music-event curves and swap in video-event curves at inference time. Thus, we bridge modalities through temporal structure rather than paired supervision while enabling independent timing and semantic control. On OES-Pub, MovieGenBench-Music, and AIST++, V2M-ZERO consistently achieves state-of-the-art results without paired music and video data, outperforming paired-data methods in audio quality, semantic alignment, and temporal synchronization. Direct human evaluation reinforces this.

Acknowledgments and Disclosure of Funding

This work was supported by Laboratory for Analytic Sciences via NC State University, ONR Award N00014-23-1-2356, NIH Award R01HD11107402, and Sony Focused Research award.

A Appendix overview

Our appendix consists of:

1. Limitations and broader impact.
2. Additional Implementation details.
3. Additional quantitative results.
4. Qualitative results on event curves.

B Limitations and broader impact

Limitations. Existing automatic metrics capture complementary but incomplete aspects of video-to-music generation. They measure audio quality, coarse semantic alignment, and local synchronization between scene and music changes, but do not fully capture complex temporal structures, macro-level phrasing, or professional scoring quality. As a result, some extensions of V2M-ZERO that may improve longer-term musical structure cannot yet be fairly evaluated, and we therefore keep the main system focused on the aspects that can be measured reliably.

Broader impact. This work can reduce the effort needed to create time-aligned background music and make video-to-music tools more accessible. It may be especially useful for commercial systems, since licensed video-music pairs are difficult and expensive to obtain, while licensing text-music data and silent videos is more feasible. Potential negative impacts include encouraging greater media consumption and screen time, which may raise concerns around digital addiction and social isolation.

Algorithm 1 Implementation details of scene cut hit.

```python
import torchaudio
import librosa
from scenedetect import detect, AdaptiveDetector

def process_video(video_path, tolerance=0.1):
    # Detect scenes
    scene_list = detect(video_path, AdaptiveDetector())
    scene_cut_times = []
    for i, (start, end) in enumerate(scene_list):
        if i < len(scene_list) - 1:
            scene_cut_times.append(end.get_seconds())

    if len(scene_cut_times) == 0:
        return None

    # Load audio
    x, sr = torchaudio.load(video_path)
    x = x.mean(0).numpy()

    # Beat/onset extraction
    tempo, beats = librosa.beat.beat_track(y=x, sr=sr)
    onsets = librosa.frames_to_time(beats, sr=sr)

    # Hit-rate computation
    hits = 0
    for c in scene_cut_times:
        for o in onsets:
            if abs(o - c) <= tolerance:
                hits += 1
                break

    return hits / len(scene_cut_times)
```

Algorithm 2 Music prompt generation from video [28]

Require: Video V, Automatic Speech Recognition $\mathcal{A}$, Vision-Language Model $\mathcal{L}_v$, Large Language Model $\mathcal{L}$

Extract transcript: $T \leftarrow \mathcal{A}(V)$
Initialize caption set: $C \leftarrow \varnothing$
for each sampled frame f_i from V **do**
    Obtain visual description: $C_i \leftarrow \mathcal{L}_v(f_i)$
    $C \leftarrow C \cup \{C_i\}$
end for
Generate visual summary: $S \leftarrow \mathcal{L}(C)$
Generate music prompt: $P \leftarrow \mathcal{L}(T, S)$
return P

C Additional Implementation details

Scene cut hit (SCH). To evaluate the temporal consistency between generated music and the underlying video dynamics, we design and use the SCH metric. The intuition is as follows: When a scene change occurs in the video, well-aligned background music should exhibit a corresponding onset or noticeable rhythmic change. This principle reflects common filmmaking conventions, where directors often synchronize cuts with musical beats to enhance pacing and emotional impact. The SCH metric is conceptually similar to beat-matching metrics [48] but describes aperiodic visual events. Given a video, we first detect its scene cuts using the `PySceneDetect`, obtaining a set of scene-cut timestamps that serve as temporal anchors. We then extract the audio track and compute musical onsets using the beat–tracking algorithm provided in `Librosa`. While more sophisticated onset detectors are available, beat tracking offers a reproducible, robust, and musically meaningful proxy for rhythmic emphasis and percussion peaks. For each detected scene cut, we count a *hit* if at least one musical onset falls within a small temporal tolerance (we use ± 0.1 seconds). The final SCH score is the ratio of matched cuts to all scene cuts. The complete procedure is summarized in Algorithm 1, which outlines the pipeline from scene detection to music onset extraction and the computation of the overall hit rate.

VLLM for music prompts. To obtain high-level semantic music prompts that reflect the mood/energy/etc. of a video, we adopt the Vibe framework [28] with modern multimodal components. Given an input video, we first obtain its transcript using the Whisper [74] ASR model, which provides a robust textual representation of the spoken content. Dialogue and narration often contain emotionally or contextually important cues, making the transcript an essential input to the music–prompt generation process.

In parallel, we sample video frames and extract frame-level visual descriptions using Gemma-4B [89]. These descriptions capture high-level semantic attributes such as environment, actions, character states, and affective cues. Since individual captions may be redundant, we aggregate them using Gemma-4B, which summarizes the set into a compact, coherent representation of the entire video.

Finally, the LLM conditions jointly on the transcript and the visual summary to produce the final music prompt, which describes mood, instrumentation, intensity, and emotional character. This process encourages prompts that capture multimodal semantics and better align with creators' intuitive descriptions. The full procedure is provided in Algorithm 2.

D Additional quantitative results.

Domain-specific visual encoders. Our event-curve framework enables performance gains via inference-time encoder selection. We validate this on dance videos (AIST++), where motion is tightly synchronized to a single object. Our default configuration with DINOv2 outperforms all the competing methods (BCS: 0.5522, BHS: 0.5748, F1: 0.5750, TD: 17.23). We hypothesize that CoTracker [36], a point-level motion tracker, is more suitable for this task. This is true and improves all metrics (BCS: 0.5818, BHS: 0.6274, F1: 0.5856, TD: 12.24) in Table 3a. This shows that V2M-ZERO can be customized at inference by picking a domain-specific encoder.

Table 5: **Comparison with a text-only baseline.** We compare V2M-ZERO against the text-to-music model used for our finetuning, without event-curve conditioning, on OES-Pub.

Method	FAD↓	CLAP↑	SCH↑
Text-only	**3.63**	0.23	0.35
V2M-ZERO	4.95	0.23	**0.61**

Table 6: **Comparison with a large-scale open-source model.** We compare V2M-ZERO with HunyuanVideo-Foley [78] on OES-Pub.

Method	FAD*↓	CLAP↑	SCH↑
HunyuanVideo-Foley [78]	15.02	0.12	0.36
V2M-ZERO	**4.95**	**0.23**	**0.61**

Table 7: **Event-curve Fréchet distance comparison.** M evaluates generated vs. ground-truth music event curves. M+V compares concatenated music–video event curves. M-V compares the distribution of generated music and ground-truth video. M|V evaluates conditional music event distributions given video. We additionally report human preference for temporal alignment. * denotes statistical significance.

	OES-Pub				MovieGenBench-Music				Human Eval
Method	**M**	**M+V**	**M-V**	**M | V**	**M**	**M+V**	**M-V**	**M | V**	**Temporal Align.**
M^2UGen [57]	6.29	6.96	18.26	1.33	1.96	1.99	24.76	1.52	Ours wins*
GVMGen [119]	3.45	4.04	10.10	0.48	1.31	1.34	20.09	1.02	Ours wins*
MTCV2M [101]	4.33	4.53	10.90	0.92	1.53	1.75	21.03	0.98	Ours wins*
VidMuse [92]	3.25	3.86	12.40	0.79	4.27	4.29	40.89	4.02	Ours wins
AudioX [91]	4.13	4.63	1.75	1.93	1.52	1.54	19.70	1.18	Ours wins*
SONIQUE [111]	3.13	3.75	8.50	0.15	11.10	11.12	6.33	9.75	Ours wins*
V2M-ZERO (Ours)	5.34	5.87	1.20	3.09	3.27	3.29	27.90	1.75	N/A

Comparison with the text-only baseline. In Table 5, we compare V2M-ZERO with a text-only baseline, i.e., the same text-to-music backbone, checkpoint, and predicted music prompts without event-curve conditioning, on OES-Pub. The results show that event-curve conditioning substantially improves temporal alignment, increasing SCH from 0.35 to **0.61**, while preserving semantic consistency, as both models achieve the same CLAP score of 0.23. Although the text-only baseline achieves similar music quality, its much weaker SCH suggests that text guidance alone is insufficient for accurate synchronization. Overall, these results suggest that event curves improve synchronization by injecting explicit temporal structure, while leaving semantic alignment largely unchanged.

Comparison with a large-scale open-source model. In Table 6, we compare V2M-ZERO with HunyuanVideo-Foley, a large-scale open-source model designed for general audio generation rather than music generation, which has not been evaluated on video-to-music benchmarks. Since HunyuanVideo-Foley generates music but also speech and environmental sounds, it performs substantially worse across all three metrics: FAD* (15.02 vs. 4.95), CLAP (0.124 vs. 0.23), and SCH (0.36 vs. 0.61). However, this comparison should be interpreted with caution, as HunyuanVideo-Foley is not designed for music generation and is therefore not well-suited to this setting or our music-focused evaluation protocol.

Temporal alignment analysis. Since the core idea of V2M-ZERO is based on event curves, we further investigate how event-curve distances relate to low-level temporal alignment quality. Specifically, since our method is designed to match these event curves, we attempt to quantify and understand the distributional behavior of this tight synchronization. In Table 7, we report four variants of Fréchet Distance computed between different event-curve distributions: (1) **M**, generated to ground-truth music event curves, (2) **M+V**, concatenated generated music-video event curves and ground-truth music-video event curves. (3) **M-V**, generated music event curves to video event curves, and (4) **M|V**, generated music event curves and ground-truth music event curves, both conditioned on video event curves. **M|V** uses a block partitioned Gaussian to represent the music given video con-

Table 8: **Video-event curves across different temporal scales.** We study video-event curves computed from different frame offsets in the similarity calculation using Stable-Audio-ControlNet [14] on OES-Pub.

Video-event curve design	FAD*↓	CLAP↑	SCH↑
No video-event conditioning	4.86	**0.18**	0.28
60 frames (2 s)	3.76	0.15	0.29
15 frames (500 ms)	**3.65**	0.15	0.35
Multiple Offsets (1, 15, 60)	3.84	0.16	0.33
1 frame (33 ms, default)	4.13	0.17	**0.38**

ditional distribution. The music event curves are extracted by musicfm [100], and the video event curves are extracted by DINOv2 [68]

We compare these additional quantitative results with human judgments of temporal alignment following the protocol used in the main paper. Interestingly, the human preference results (rightmost column of Table 7) do not correlate strongly with any of the event-curve distances. Models that achieve low Fréchet distances are not necessarily preferred by human raters, and vice versa.

This mismatch suggests an important insight: event curves may be highly suitable as a generative intermediate representation, but they may not be appropriate as an evaluation metric. Event curves are good at capturing dense, continuous temporal dynamics that help guide generation, yet humans appear to assess alignment based on sparse, salient temporal moments (e.g., impactful transitions or rhythm–scene coincidences), not global curve similarity. As a result, metrics that rely purely on curve distributions reward continuous structural alignment, while human perception is more selective and non-uniform over time.

A second contributing factor is that current evaluation datasets (OES-Pub and MovieGenBench-Music) are not specifically curated for temporal synchronization assessment. Many clips lack strong or consistent rhythmic structure, limiting the signal available to curve-based metrics.

In sum, these observations indicate that while **event curves provide a powerful foundation for generation, but the distributional distances do not necessarily reflect human-perceived temporal alignment quality**. Future benchmarks tailored to temporal synchronization and metrics that account for sparse perceptual saliency may bridge this gap.

Event curve robustness to non-semantic changes. To test the robustness of DINOv2 to non-semantic visual changes that may affect event curves, we perform random non-semantic augmentations to 3,450 OES-Pub frames (fps=1), where each frame is randomly perturbed: subpixel translation ± 4 px, rotation $\pm 4^\circ$, brightness step $[0.4, 1.6]\times$, or gamma change $[0.4, 1.8]$. The cosine similarity between DINOv2 features before and after augmentation is $\mu = 0.983$ with $\sigma = 0.025$, indicating that the video event curves remain robust under these perturbations.

Representation power of event curves. Event curves give relative timing information, while text conveys complex semantic guidance. To further measure the representation power of event curves, we set up a 3-way classification task among cinematic/natural/dance videos using samples from OES-Pub/MovieGenBench/AIST++. Using the video-event curves as input, a 1-layer MLP with a 90%/10% train–test split achieved 68.2% test accuracy, showing that the curves contain meaningful information based on video dynamics. This shows that video-event curves already capture non-trivial video characteristics, which are further complemented by text prompts in the full model.

Different temporal dependencies for video-event curves. In Table 8, we study the effect of different similarity offsets (i.e., temporal dependencies) in video-event curve extraction on OES-Pub. We use Stable-Audio-ControlNet [14], which is trained with RMS control rather than music event curves, as a controlled testbed. Since this setup does not involve music-event-curve offsets during training, we vary only the offset used to compute video-event curves at inference, allowing us to study the effect of video-event temporal scale in isolation.

At 30 fps, offsets 1, 15, and 60 correspond to about 33 ms, 500 ms, and 2 s, respectively. The multiple-offset variant uses video-event curves computed with offsets 1, 15, and 60 as joint conditioning signals through compositional guidance. Longer or multiple offsets are feasible and preserve

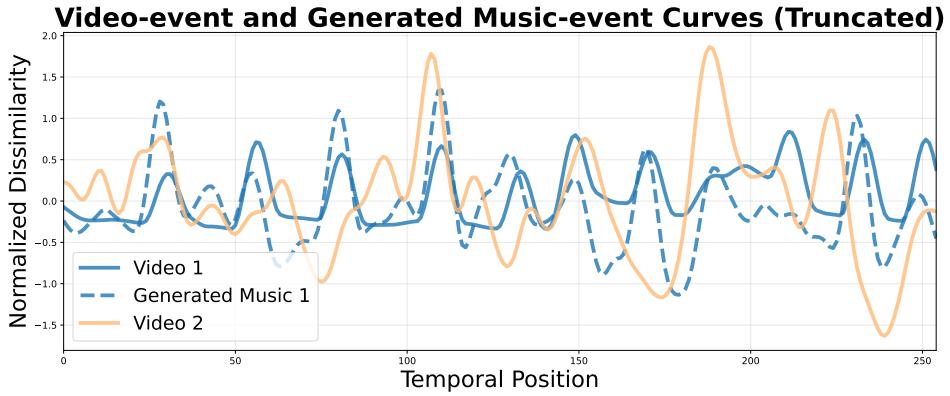


Figure 4: **Example event curves with different temporal dynamics.** The blue solid curve corresponds to a video with frequent scene cuts, while the orange curve corresponds to a video with slower visual motion, showing distinct temporal structures. This supports our design choice of using event curves to represent relative timing, while text provides complementary semantic guidance.

overall audio quality (i.e., FAD) and semantic alignment (i.e., CLAP). However, we do not have a metric that properly evaluates longer temporal dependencies across video and music. Since the default offset-1 design yields the strongest measured synchronization under the available metric (i.e., SCH), we keep offset to 1 in the main model and leave longer-term dependencies and multiple-offset conditioning for future work.

Potential extension to macro-level phrasing. Our music captioning pipeline currently produces a single prompt for the entire video, which may limit the ability to capture macro-level phrasing changes in music that correspond to different video segments. However, the same captioning framework could be applied to shorter video chunks, allowing the prompt to change over time and potentially capture more fine-grained music semantic changes.

To test this possibility, we need a shared model that supports both text-to-music (T2M) and video-to-music (V2M) generation, so that chunk-wise T2M outputs can be compared with V2M outputs under the same backbone. We choose AudioX [91] for this purpose, as it supports V2M and T2M in a single model, which supports previous music chunks as a condition. In the chunked T2M setting, this allows each new music chunk to be generated from the current chunk-level prompt while conditioning on the previously generated music chunk, making time-varying prompting practical.

We compare whole-video T2M prompting, chunked T2M prompting from 5 s video segments, and V2M. We then compute cosine similarity with MusicFM [100] features between T2M output and the V2M output on 5 s music chunks, then average across chunks within a video. Chunked prompting is closer to V2M than whole-video prompting (i.e., **0.86** versus **0.63** for the whole-video prompt), suggesting that time-varying captions can make T2M generation more like V2M and may help capture macro-level phrasing. Since there is no direct metric that properly evaluates macro-level phrasing, we treat this as preliminary evidence rather than a main design choice. We therefore keep full-video prompts in the main model for simplicity, and leave systematic chunk-wise prompting for future work.

E Qualitative results on event curves

Event curves over diverse dynamics. V2M-ZERO is conditioned on both event curves and text. Event curves with normalization provide relative temporal structure, aiming to represent *when* changes happen, while text conveys complementary information such as semantic content, energy, and mood.

To validate this behavior, Figure 4 shows two videos with distinct visual dynamics together with an example generated music event curve. The blue solid curve corresponds to a video with frequent scene cuts (i.e., Good Sample, (Sora2) 20251120_0147_01kaengrpqetrrwv7jyycmnkzn.mp4), while the orange curve corresponds to a video dominated by slower camera motion, such as pans (i.e., Ran-

dom Sample, (MovieGen) 148.mp4, in demo). Although both visual signals are normalized, their temporal patterns remain clearly different: the scene-cut video exhibits sharp, sparse peaks, whereas the slow-pan video varies more smoothly over time. This example shows that normalization does not erase temporal structure; instead, it preserves relative changes that are useful for synchronization.

Fine-grained temporal precision. Although event curves are extracted using temporal smoothing, the resulting peaks (i.e., video 1 in Figure 4) still retain precise temporal localization. We use a Hann window, whose central lobe preserves peak locations while suppressing high-frequency noise. In equivalent bandwidth, our smoothing corresponds to roughly 21 samples at 30 fps, or about a 700 ms window. In practice, this smoothing improves robustness without preventing accurate localization of salient events, which is consistent with our use of a ± 100 ms tolerance in Scene Cut Hit (SCH).

References

[1] 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*, 2023. 1, 3

[2] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv Preprint*, 2025. 6, 10

[3] Yatong Bai, Jonah Casebeer, Somayeh Sojoudi, and Nicholas J. Bryan. DRAGON: Distributional rewards optimize diffusion generative models. *TMLR*, 2025. 6

[4] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. AudioLM: a language modeling approach to audio generation. *TASLP*, 2023. 3

[5] Dimitrios Bralios, Jonah Casebeer, and Paris Smaragdis. Re-bottleneck: Latent re-structuring for neural audio autoencoders. In *MLSP*, 2025. 6

[6] Dimitrios Bralios, Paris Smaragdis, and Jonah Casebeer. Learning to upsample and upmix audio in the latent domain. *arXiv Preprint*, 2025. 6

[7] Jonah Casebeer, Ge Zhu, Zhepei Wang, and Nicholas J Bryan. A generative-first neural audio autoencoder. In *ICASSP*, 2026. 4, 10

[8] Kan Chen, Chuanxi Zhang, Chen Fang, Zhaowen Wang, Trung Bui, and Ram Nevatia. Visually indicated sound generation by perceptually optimized classification. In *ECCVW*, 2018. 3

[9] Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. Generating visually aligned sound from videos. *TIP*, 2020. 3

[10] Ziyang Chen, Daniel Geng, and Andrew Owens. Images that sound: Composing images and sounds on a single canvas. In *NeurIPS*, 2024. 3

[11] Ziyang Chen, Prem Seetharaman, Bryan Russell, Oriol Nieto, David Bourgin, Andrew Owens, and Justin Salamon. Video-guided foley sound generation with multimodal controls. In *CVPR*, 2025. 3

[12] Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *CVPR*, 2025. 3

[13] Sanjoy Chowdhury, Sayan Nag, KJ Joseph, Balaji Vasan Srinivasan, and Dinesh Manocha. MeLFusion: Synthesizing music from image and language cues using diffusion models. In *CVPR*, 2024. 2, 3

[14] Ruben Ciranni, Giorgio Mariani, Michele Mancusi, Emilian Postolache, Giorgio Fabbro, Emanuele Rodolà, and Luca Cosmo. Cocola: Coherence-oriented contrastive learning of musical audio representations. *arXiv Preprint*, 2024. 9, 14, 26

[15] Jorge L. Contreras. Music licensing. In *Intellectual Property Licensing and Transactions: Theory and Practice*, page 493–522. Cambridge University Press, 2022. 2

[16] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. In *NeurIPS*, 2023. 1, 3

[17] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv Preprint*, 2022. 3

[18] 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 *ACM MM*, 2021. 3, 9

[19] Yuexi Du, Ziyang Chen, Justin Salamon, Bryan Russell, and Andrew Owens. Conditional generation of audio from video via foley analogies. In *CVPR*, 2023. 3

[20] Zach Evans, CJ Carr, Josiah Taylor, Scott H Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. *arXiv Preprint*, 2024. 3

[21] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP*, 2025. 3, 5, 6

[22] Zhengcong Fei, Mingyuan Fan, Changqian Yu, and Junshi Huang. Flux that plays music. *arXiv Preprint*, 2024. 1, 3, 5

[23] Jonathan Foote and Matthew Cooper. Visualizing musical structure and rhythm via self-similarity. In *ICMC*, 2001. 5

[24] Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction guided latent diffusion model. In *ACM MM*, 2023. 3, 5

[25] Junmin Gong, Sean Zhao, Sen Wang, Shengyuan Xu, and Joe Guo. ACE-Step: A step towards music generation foundation model. *arXiv Preprint*, 2025. 1, 3

[26] 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*, 2024. 10

[27] Xin Gu, Wei Jiang, Yujian Jiang, Zhibin Su, and Ming Yan. Vmsp: Video-to-music generation with two-stage alignment and synthesis. In *ICASSP*, 2026. 3

[28] Noor Hammad, C Ailie Fraser, Erik Harpstead, Jessica Hammer, and Mira Dontcheva. "it's more of a vibe i'm going for": Designing text-to-music generation interfaces for video creators. In *DIS*, 2025. 2, 3, 4, 6, 10, 12

[29] 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 *ICASSP*, 2017. 7

[30] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv Preprint*, 2022. 4

[31] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 1979. 8

[32] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *ICML*, 2023. 3

[33] Fathinah Izzati, Xinyue Li, Yuxuan Wu, and Gus Xia. MusiScene: Leveraging mu-llama for scene imagination and enhanced video background music generation. *arXiv Preprint*, 2025. 3

[34] 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 *AAAI*, 2026. 3

[35] Jaeyong Kang, Soujanya Poria, and Dorien Herremans. Video2music: Suitable music generation from videos using an affective multimodal transformer model. *arXiv Preprint*, 2023. 3

[36] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024. 6, 12

[37] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv Preprint*, 2018. 7

[38] Haven Kim, Zachary Novack, Weihan Xu, Julian McAuley, and Hao-Wen Dong. Video-guided text-to-music generation using public domain movie collections. In *ISMIR*, 2025. 2, 3, 6, 7

[39] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. In *ICLR*, 2023. 3

[40] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *NeurIPS*, 2023. 3

[41] Saksham Singh Kushwaha and Yapeng Tian. VinTAGe: Joint video and text conditioning for holistic audio generation. In *CVPR*, 2025. 3

[42] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for video action recognition. In *ICCV*, 2021. 5

[43] Max WY Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, et al. Efficient neural music generation. In *NeurIPS*, 2023. 1, 3

[44] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *NeurIPS*, 2019. 3

[45] Jiajun Li, Tianze Xu, Xuesong Chen, Xinrui Yao, Jingchou Han, and Shuchang Liu. Mozart's touch: a lightweight multimodal music generation framework based on pre-trained large models. In *AIGC*, 2025. 3

[46] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. AI choreographer: Music conditioned 3d dance generation with AIST++. In *ICCV*, 2021. 7, 9

[47] 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*, 2024. 2, 3

[48] Sifei Li, Weiming Dong, Yuxin Zhang, Fan Tang, Chongyang Ma, Oliver Deussen, Tong-Yee Lee, and Changsheng Xu. Dance-to-music generation with encoder-based textual inversion. In *SIGGRAPH Asia*, 2024. 3, 7, 9, 12

[49] Sizhe Li, Yiming Qin, Minghang Zheng, Xin Jin, and Yang Liu. Diff-BGM: A diffusion model for video background music generation. In *CVPR*, 2024. 3

[50] 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*, 2024. 2, 3

[51] Yan-Bo Lin and Gedas Bertasius. Siamese vision transformers are scalable audio-visual learners. In *ECCV*, 2024. 6, 10

[52] 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 *WACV*, 2025. 2, 3, 7

[53] 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. In *ICML*, 2023. 3, 5

[54] Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. AudioLDM 2: Learning holistic audio generation with self-supervised pretraining. *arXiv Preprint*, 2023. 3, 5

[55] Huadai Liu, Jialei Wang, Kaicheng Luo, Wen Wang, Qian Chen, Zhou Zhao, and Wei Xue. ThinkSound: Chain-of-thought reasoning in multimodal large language models for audio generation and editing. In *NeurIPS*, 2025. 3

[56] Shansong Liu, Atin Sakkeer Hussain, Qilong Wu, Chenshuo Sun, and Ying Shan. M^2 UGen: Multi-modal music understanding and generation with the power of large language models. *arXiv Preprint*, 2023. 3

[57] Shansong Liu, Atin Sakkeer Hussain, Qilong Wu, Chenshuo Sun, and Ying Shan. MuMu-LLaMA: Multi-modal music understanding and generation via large language models. *arXiv Preprint*, 2024. 7, 13

[58] Xiulong Liu, Kun Su, and Eli Shlizerman. Tell what you hear from what you see-video to audio generation through text. In *NeurIPS*, 2024. 3

[59] Xiaohao Liu, Teng Tu, Yunshan Ma, and Tat-Seng Chua. Extending visual dynamics for video-to-music generation. *arXiv Preprint*, 2025. 2, 3

[60] Zihan Liu, Shuangrui Ding, Zhixiong Zhang, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. SongGen: A single stage auto-regressive transformer for text-to-song generation. In *ICML*, 2025. 1, 3

[61] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-Foley: Synchronized video-to-audio synthesis with latent diffusion models. In *NeurIPS*, 2023. 3

[62] Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, et al. The song describer dataset: a corpus of audio captions for music-and-language evaluation. *arXiv Preprint*, 2023. 7, 8

[63] Xinhao Mei, Varun Nagaraja, Gael Le Lan, Zhaoheng Ni, Ernie Chang, Yangyang Shi, and Vikas Chandra. FoleyGen: Visually-guided audio generation. *arXiv Preprint*, 2023. 3

[64] Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward controllable text-to-music generation. *arXiv Preprint*, 2023. 3, 5

[65] Zachary Novack, Zach Evans, Zack Zukowski, Josiah Taylor, CJ Carr, Julian Parker, Adnan Al-Sinan, Gian Marco Iodice, Julian McAuley, Taylor Berg-Kirkpatrick, et al. Fast text-to-audio generation with adversarial post-training. *arXiv Preprint*, 2025. 3

[66] Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan. DITTO: Diffusion inference-time t-optimization for music generation. In *ICML*, 2024. 1, 3

[67] Zachary Novack, Ge Zhu, Jonah Casebeer, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan. Presto! distilling steps and layers for accelerating music generation. In *ICLR*, 2025. 6

[68] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv Preprint*, 2023. 5, 6, 8, 10, 14

[69] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *CVPR*, 2016. 3

[70] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 5

[71] Geoffroy Peeters. Self-similarity-based and novelty-based loss for music structure analysis. In *International Society of Music Information Retreival*, 2023. 5

[72] 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*, 2024. 7

[73] Fan Qi, Kunsheng Ma, and Changsheng Xu. Customized condition controllable generation for video soundtrack. In *CVPR*, 2025. 3

[74] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *ICML*, 2023. 12

[75] Lita Rosario-Richardson and John Simson. Synchronization licensing in the new music digital ecosystem. In *The Oxford Handbook of Music Law and Policy*. Oxford University Press, 2024. 2

[76] Ciara Rowles, Varun Jampani, Simon Donné, Shimon Vainer, Julian Parker, and Zach Evans. Foley control: Aligning a frozen latent text-to-audio model to video. *arXiv Preprint*, 2025. 3

[77] Flavio Schneider, Ojasv Kamal, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion. *arXiv Preprint*, 2023. 1, 3

[78] Sizhe Shan, Qiulin Li, Yutao Cui, Miles Yang, Yuehai Wang, Qun Yang, Jin Zhou, and Zhao Zhong. Hunyuanvideo-foley: Multimodal diffusion with representation alignment for high-fidelity foley audio generation. *arXiv Preprint*, 2025. 13

[79] Megha Sharma, Muhammad Taimoor Haseeb, Gus Xia, and Yoshimasa Tsuruoka. M2M-Gen: A multimodal framework for automated background music generation in japanese manga using large language models. *arXiv Preprint*, 2024. 3

[80] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 5

[81] Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *CVPR*, 2018. 3

[82] 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 music generation. In *AAAI*, 2024. 3

[83] Kun Su, Xiulong Liu, and Eli Shlizerman. From vision to audio and beyond: A unified model for audio-visual representation and generation. In *ICML*, 2024. 3

[84] Changchang Sun, Gaowen Liu, Charles Fleming, and Yan Yan. Enhancing dance-to-music generation via negative conditioning latent diffusion model. In *CVPR*, 2025. 3

[85] Or Tal, Felix Kreuk, and Yossi Adi. Auto-regressive vs flow-matching: a comparative study of modeling paradigms for text-to-music generation. *TMLR*, 2025. 1, 3

[86] Vanessa Tan, Junghyun Nam, Juhan Nam, and Junyong Noh. Motion to dance music generation using latent diffusion model. In *SIGGRAPH Asia*, 2023. 9

[87] Zineng Tang, Ziyi Yang, Mahmoud Khademi, Yang Liu, Chenguang Zhu, and Mohit Bansal. Codi-2: In-context, interleaved, and interactive any-to-any generation. *arXiv Preprint*, 2023. 3

[88] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. In *NeurIPS*, 2023. 3

[89] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv Preprint*, 2025. 6, 10, 12

[90] Sida Tian, Can Zhang, Wei Yuan, Wei Tan, and Wenjie Zhu. XMusic: Towards a generalized and controllable symbolic music generation framework. *TMM*, 2025. 3

[91] Zeyue Tian, Yizhu Jin, Zhaoyang Liu, Ruibin Yuan, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. Audiox: Diffusion transformer for anything-to-audio generation. *arXiv Preprint*, 2025. 2, 3, 7, 13, 15

[92] 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 *CVPR*, 2025. 2, 3, 7, 13

[93] Zeyue Tian, Binxin Yang, Zhaoyang Liu, Jiexuan Zhang, Ruibin Yuan, Hubery Yin, Qifeng Chen, Chen Li, Jing Lv, Wei Xue, et al. Audio-omni: Extending multi-modal understanding to versatile audio generation and editing. In *SIGGRAPH*, 2026. 3

[94] Xinyi Tong, Yiran Zh, Jishang Chen, Chunru Zhan, Tianle Wang, Sirui Zhang, Nian Liu, Tiezheng Ge, Duo Xu, Xin Jin, Feng Yu, and Song-Chun Zhu. Video echoed in music: Semantic, temporal, and rhythmic alignment for video-to-music generation. *arXiv Preprint*, 2025. 3

[95] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. AIST dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *ISMIR*, 2019. 7, 9

[96] Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. Audiobox: Unified audio generation with natural language prompts. *arXiv Preprint*, 2023. 3

[97] Baisen Wang, Le Zhuo, Zhaokai Wang, Chenxi Bao, Wu Chengjing, Xuecheng Nie, Jiao Dai, Jizhong Han, Yue Liao, and Si Liu. Multimodal music generation with explicit bridges and retrieval augmentation. *arXiv Preprint*, 2024. 3

[98] Jun Wang, Xijuan Zeng, Chunyu Qiang, Ruilong Chen, Shiyao Wang, Le Wang, Wangjing Zhou, Pengfei Cai, Jiahui Zhao, Nan Li, et al. Kling-Foley: Multimodal diffusion transformer for high-quality video-to-audio generation. *arXiv Preprint*, 2025. 3

[99] Peter Wegele. *Max Steiner: Composing, Casablanca, and the Golden Age of Film Music*. Bloomsbury Publishing PLC, 2014. 9

[100] Minz Won, Yun-Ning Hung, and Duc Le. A foundation model for music informatics. In *ICASSP*, 2024. 5, 6, 10, 14, 15

[101] Junxian Wu, Weitao You, Heda Zuo, Dengming Zhang, Pei Chen, and Lingyun Sun. Controllable video-to-music generation with multiple time-varying conditions. In *ACM MM*, 2025. 2, 3, 7, 13

[102] Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J Bryan. Music controlnet: Multiple time-varying controls for music generation. *TASLP*, 2024. 26

[103] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*, 2023. 7

[104] Zhifeng Xie, Qile He, Youjia Zhu, Qiwei He, and Mengtian Li. FilmComposer: Llm-driven music production for silent film clips. In *CVPR*, 2025. 2, 3

[105] 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*, 2025. 10

[106] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *TASLP*, 2023. 3

[107] M Yinghao, Ø Anders, R Anton, S Bleiz MacSen Del, S Charalampos, and D Chris. Foundation models for music: A survey. *arXiv preprint arXiv:2408.14340*, 3, 2024. 2

[108] Jiashuo Yu, Yaohui Wang, Xinyuan Chen, Xiao Sun, and Yu Qiao. Long-term rhythmic video soundtracker. In *ICML*, 2023. 3, 9

[109] Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, et al. Yue: Scaling open foundation models for long-form music generation. *arXiv Preprint*, 2025. 1, 3

[110] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *TASLP*, 2021. 3

[111] Liqian Zhang and Magdalena Fuentes. SONIQUE: Video background music generation using unpaired audio-visual data. In *ICASSP*, 2025. 2, 3, 7, 13

[112] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Visual to sound: Generating natural sound for videos in the wild. In *CVPR*, 2018. 3

[113] 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 *CVPR*, 2025. 3

[114] Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. Quantized GAN for complex music generation from dance videos. In *ECCV*, 2022. 3

[115] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal music and image generation. In *ICLR*, 2023. 3, 9

[116] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. Music2dance: Dancenet for music-driven dance generation. *TOMM*, 2022. 3

[117] 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 *ICCV*, 2023. 3

[118] Alon Ziv, Itai Gat, Gael Le Lan, Tal Remez, Felix Kreuk, Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. Masked audio generation using a single non-autoregressive transformer. In *ICLR*, 2024. 1, 3

[119] 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 *AAAI*, 2025. 2, 3, 7, 13

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the zero-pair setting, event-curve conditioning, and evaluated datasets.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work at first section of the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

Justification: We do not provide any theoretical result.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include experimental setup in main draft and implementation details in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Evaluation datasets are publicly available. Code and training datasets won't be released due to copyright and licensing issues. However, we provide detailed instructions for reproducing and results based on public models. Also note, Stable Audio ControlNet [14] provides an RMS curve conditioned text-to-music generation model as introduced in Music ControlNet [102] that can be used as a starting point for reproduction of our work.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We include all details in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow standard practices similar to other methods, which do not provide error bars for these evaluations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include training resources in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We fully follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of our work at first section of the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not have such data or models to release.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use licensed data for training and follow the term of use. We also cite all evaluation datasets. Note, our work is precisely motivated by the complexity of licensing paired music and video data and enables training video-to-music generation models on separately licensed silent videos and music files, respectively.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We provide the details in the main draft.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: We describe the human evaluation setup in the paper. IRB approval was not required for listening generated music.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use publicly available LLMs only to generate music-related prompts from video inputs as an intermediate output in our video-to-music generation pipeline. The LLMs are not used to design experiments, analyze results, or write the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.