KeyVID: Keyframe-Aware Video Diffusion for Audio-Synchronized Visual Animation

Anonymous authors
Paper under double-blind review

Abstract

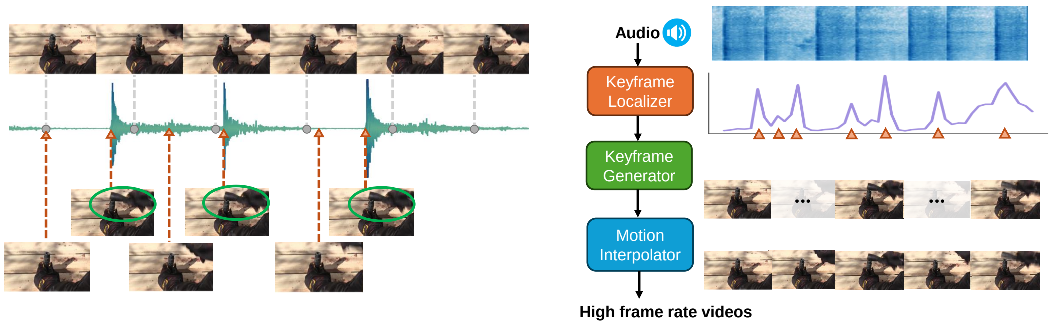
Generating video from various conditions, such as text, image, and audio, enables precise spatial and temporal control, leading to high-quality generation results. Most existing audio-to-visual animation models rely on uniformly sampled frames from video clips. Such a uniform sampling strategy often fails to capture key audio-visual moments in videos with dramatic motions, causing unsmooth motion transitions and audio-visual misalignment. To address these limitations, we introduce **KeyVID**, a keyframe-aware audio-to-visual animation framework that adaptively prioritizes the generation of keyframes in audio signals to improve the generation quality. Guided by the input audio signals, KeyVID first localizes and generates the corresponding visual keyframes that contain highly dynamic motions. The remaining frames are then synthesized using a motion interpolation module, effectively reconstructing the full video sequence. This design enables the generation of high frame-rate videos that faithfully align with audio dynamics, while avoiding the cost of directly training with all frames at a high frame rate. Through extensive experiments, we demonstrate that KeyVID significantly improves audio-video synchronization and video quality across multiple datasets, particularly for highly dynamic motions.

1 Introduction

Recent years have witnessed remarkable progress in video generation, driven by advancements in diffusion-based models (Xing et al., 2024; Chen et al., 2023a; 2024; He et al., 2022; Singer et al., 2023; Ho et al., 2022b; Guo et al., 2024; Hong et al., 2022; Yang et al., 2024; Fan et al., 2025; Blattmann et al., 2023a;b). These frameworks typically condition the generation process on *text* prompts and/or *image* inputs, where the text provides semantic guidance (*e.g*., actions, objects, or stylistic cues), while the image specifies spatial composition (*e.g*., object layout, scene structure or visual styles). Despite their success, these methods largely focus on aligning visual outputs with static text or images, leaving dynamic, time-sensitive modalities such as *audio* underexplored.

Audio-Synchronized Visual Animation (ASVA) (Zhang et al., 2024b) aims to animate a static image into a video with objects' motion dynamics that are semantically aligned and temporally synchronized with the input audio. It utilizes audio cues to provide more fine-grained semantic and temporal control for video generation, which requires deep understanding of audio semantics, audio-visual correlations, and object dynamics. To achieve precise audio-visual synchronization in ASVA, it is crucial to align key visual actions accurately with their corresponding audio signals. For example, given an audio clip of hammering sounds, the hammer in the video should strike the nail exactly when the impact sound occurs. However, this synchronization is constrained by the frame rates of the video generation models. For example, AVSyncD (Zhang et al., 2024b) is trained to generate videos at 6 FPS, posing a significant challenge for audio-synchronized video generation. Since audio carries fine-grained temporal information, the key moments in the audio can be lost in uniformly sampled low frame rate videos (see Fig. 1(a)), leading to compromised audio-video synchronization.

A straightforward solution is to train a video generation model on high frame rate data to match the fine-grained temporal information in audio. However, this brute-force approach treats all time steps equally and introduces redundant frames in low-motion regions. It also fails to leverage the structural information in the input audio to focus the model capacity on salient moments, which is crucial for audio-visual synchronization. In addition, this approach incurs substantial computational costs in terms of GPU memory and training time. To alleviate this, a two-stage strategy has been



(a) (b)

Figure 1: (a) **Uniform frames vs. keyframes.** *Top*: Uniformly sampled sparse frames, which fail to capture the key moments evident in the corresponding audio (*Middle*). *Bottom*: Keyframes precisely aligned with the hammer striking down, matching the critical moments in the audio waveform. (b) **KeyVID video generation pipeline.** KeyVID first detects keyframe time steps from the audio input with the *keyframe localizer* and then utilizes a *keyframe generator* to generate the corresponding visual keyframes. Intermediate frames are generated with the *motion interpolator*.

proposed that first generates low frame rate videos and then applies frame interpolation to obtain high frame rate videos (Blattmann et al., 2023a; Singer et al., 2023; Ho et al., 2022a). And a random frame rate strategy is proposed to use random frame sampling rates while maintaining a small, fixed number of frames during training (Singer et al., 2023; Zhou et al., 2022). However, the two-stage approach struggles in modeling highly dynamic sequences, where critical events may be lost due to the sparsity of the initial uniform frames, and the random frame rate strategy fails to model long-term temporal dependency at high frame rates due to the limited number of total frames.

In this work, instead of sampling uniform frames, we propose **KeyVID**, a **Key**frame-aware **VI**deo **D**iffusion framework that adaptively selects and generates sparse yet informative keyframes guided by audio cues to capture critical audio-visual events (Fig. 1(b)). We first develop a keyframe selection strategy that identifies critical moments in the video sequence based on an optical flow-based motion score. We train a *keyframe localizer* that predicts such keyframe positions directly from the input audio cue. Next, instead of applying uniform downsampling to video frames, we select the keyframes to train a *keyframe generator*. The keyframe generator explicitly captures crucial moments of dynamic motion that might otherwise be missed with uniform sampling without requiring an excessively high number of frames. Then, we train a specialized *motion interpolator* to synthesize intermediate frames between the keyframes to generate high frame rate videos. The motion interpolator ensures smooth motion transition and precise audio-visual synchronization throughout the sequence. This approach is similar to how the animation industry creates smooth and dynamic movements, where the *Key Animator* establishes key moments in a scene and the *Inbetweener* fills in the gaps to ensure that the movements appear seamless and fluid. This selective temporal focus enables smoother motion transitions and sharper audio-visual synchronization without the overhead of dense uniform sampling.

We conducted extensive experiments across diverse datasets featuring varying degrees of motion dynamics and audio-visual synchronization. We demonstrate that our keyframe-aware approach outperforms state-of-the-art methods in video generation quality and audio-video synchronization. In particular, on the AVSync15 dataset (Zhang et al., 2024b), we achieve an FVD score (Unterthiner et al., 2018) of 263.3, and a RelSync score (Zhang et al., 2024b) of 49.06, outperforming the state-of-the-art by absolute margins of **85.8**, and **3.54**, respectively. Our user study demonstrates a clear preference towards videos generated by KeyVID over those produced by baseline methods.

The main contributions of our work are as follows:

- We propose a novel keyframe-aware audio-to-visual animation framework that first localizes keyframe positions from the input audio and then generates the corresponding video keyframes using a diffusion model.
- We design a keyframe generator network that selectively produces sparse keyframes from the input image and audio, effectively capturing crucial motion dynamics.
- Comprehensive experiments demonstrate our superior performance in audio-synchronized video generation, particularly in highly dynamic scenes with distinct audio-visual events.

2 Related Work

Video Diffusion Models. Diffusion models (Xing et al., 2024; Chen et al., 2023a; 2024; He et al., 2022; Singer et al., 2023; Ho et al., 2022b; Guo et al., 2024; Hong et al., 2022; Yang et al., 2024; Fan et al., 2025; Blattmann et al., 2023a;b) emerge as powerful tools to generate high-quality videos. For the data sample $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$, Gaussian noise is added over T steps, creating a noisy version $\mathbf{x}_T$. A model ϵ_θ is trained to invert this process by predicting and subtracting the noise. For latent video generation (Xing et al., 2024; Zhang et al., 2023; He et al., 2022; Blattmann et al., 2023b), $\mathbf{x}$ is encoded into a latent vector $\mathbf{z}$ using an encoder $\mathcal{E}(\cdot)$ to reduce computation. The noise-adding diffusion process and the learned reverse process are conducted on $\mathbf{z}$ instead. Recent advancements in video diffusion models leverage pre-trained text encoders (Radford et al., 2021; Raffel et al., 2020) to inject text conditions into the denoising process for text-to-video generations (Blattmann et al., 2023b; Hong et al., 2022; Chen et al., 2023a; Luo et al., 2023). Moreover, image conditioning can also be introduced to enhance video generation by providing visual features that control the visual contents (Wu et al., 2024a; Yang et al., 2023; Li et al., 2023b; Chen et al., 2023b; Wei et al., 2023) or frame conditions (Xing et al., 2024; Chen et al., 2024; Guo et al., 2024; Zhang et al., 2020; Voleti et al., 2022; Franceschi et al., 2020; Babaeizadeh et al., 2018).

Audio-to-Video Generation. Compared to text and image, audio provides not only semantic cues but also fine-grained temporal signals for motion generation. Prior studies explored domain-specific audio-conditioned motion synthesis in 2D and 3D (Sun et al., 2023; Zhang et al., 2024a; Wu et al., 2024b; Sung-Bin et al., 2024; Richard et al., 2023; Liu et al., 2024), and more recent works leverage pretrained audio encoders (Girdhar et al., 2023; Elizalde et al., 2023) for general video generation. Existing methods either treat audio as a *global feature* for style/semantic control (Hertz et al., 2023; Kim et al., 2023; Wu et al., 2023) or enforce *uniform temporal alignment* with audio clips (Lee et al., 2022; Ruan et al., 2023; Zhang et al., 2024b). However, their motion quality is often limited by low frame rates or costly uniform sampling strategies, especially in highly dynamic scenes. In contrast, we introduces a *keyframe-aware framework* that localizes audio-critical moments, generates visual keyframes accordingly, and interpolates intermediate frames. This selective temporal focus enables smoother motion transitions and sharper audio-visual synchronization without the overhead of dense uniform sampling.

Keyframe-based Video Processing. In video processing, keyframes are pivotal in compressing video clips by retaining essential features, thereby facilitating efficient analysis of lengthy videos or high-dynamic motions (Kulhare et al., 2016; Shen et al., 2024; Lee et al., 2024; Xu et al., 2024; Ataallah et al., 2024). In the realm of video generation, keyframes serve as foundational references, enabling the synthesis of intermediate frames that ensure temporal coherence and visual consistency. For long video generation, current approaches employ keyframe-based generation pipelines to enhance long-term coherence in video synthesis (Zheng et al., 2024; Yin et al., 2023). Others focus on interpolation techniques from keyframes, which predict missing frames between keyframes input, ensuring motion realism and visual consistency in dynamical motions (Geng et al., 2024; Jain et al., 2024).

3 Methods

In this section, we present our keyframe-aware audio-conditioned video generation framework **KeyVID**. Given an input audio and the first frame of a video, we follow a three-stage generation process (Fig. 1(b)) and train three separate models: (1) **Keyframe Localizer** predicts a motion score curve from the input audio and detects the keyframe positions (Sec. 3.1); (2) **Keyframe Generator** generates keyframe images at detected keyframe positions conditioned on the input image and audio (Sec. 3.2); (3) **Motion Interpolator** synthesizes intermediate frames to reconstruct a smooth video with dense frames conditioned on the generated keyframe images and input audio (Sec. 3.3).

3.1 Keyframe Localization from Audio

We train a keyframe localizer to infer keyframe locations from input by exploiting the correlation between acoustic events and motion changes. For instance, a hammer striking a table generates a sharp sound that often aligns with a sudden visual transition. The network learns to predict motion scores from the input audido and then localizes keyframes from the motion score sequence.

Optical Flow based Motion Score. To train the keyframe localizer, we first generate keyframe labels by analyzing optical flow from training video sequences, as shown in Fig. 2(a). We first obtain a *motion score* for each frame by calculating the optical flow and averaging it across all pixels to represent the motion intensity of the frame. These scores collectively form a temporal motion curve across the frames.

Specifically, we employ a pre-trained RAFT model (Teed & Deng, 2020) as the optical flow estimator. Given a video clip consisting of frames $\{I_j\}_{j=1}^{T}$, RAFT computes the optical flow field $\mathbf{OF}_t$ between two frames I_j and I_{j+1}. The optical field consists of horizontal (u_t) and vertical (v_t) components at each pixel, and the motion score $M(t)$ of frame t is calculated as:

$$M(t) = \sum_{i,j} \left(|u_t(h,w)| + |v_t(h,w)|\right), \tag{1}$$

where $t = 1, \ldots, T-1$ denotes the time step of the video with T frames. (h, w) represents the pixel location.

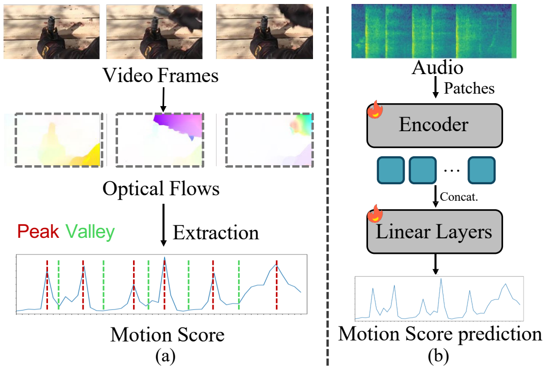


Figure 2: **Motion score computation and prediction.** (a) We compute motion scores as the average of the optical flow of each frame and localize keyframe from the peaks and valleys. (b) Keyframe localizer is trained to predict motion scores from audio to identify keyframe locations.

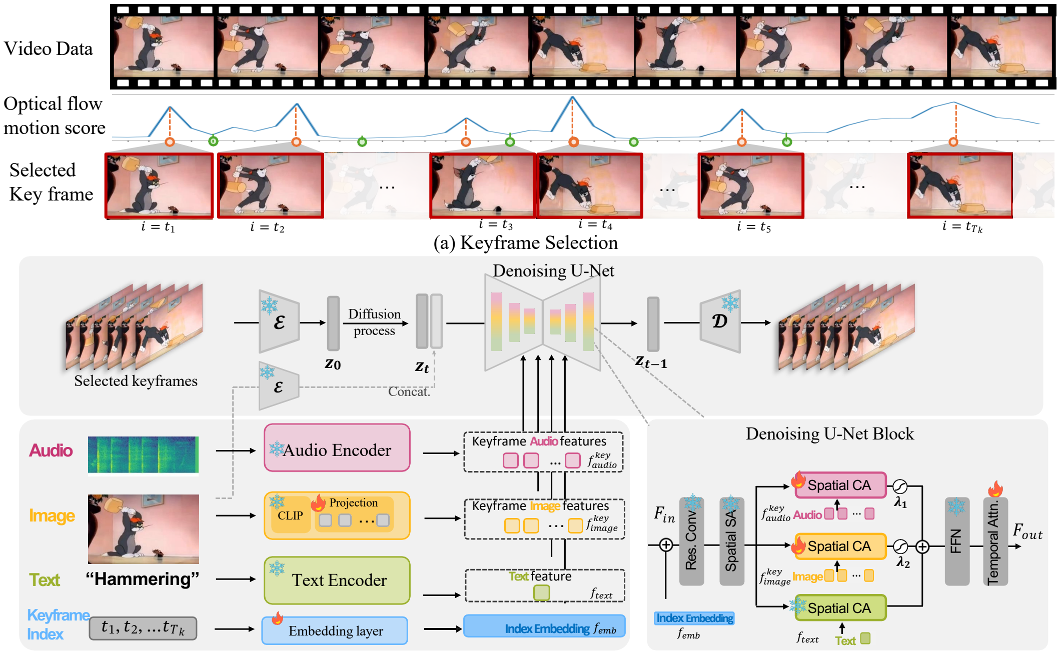
Motion Score Prediction. We train the keyframe localizer to predict motion scores from input audio, enabling it to learn the underlying relationship between motion dynamics and acoustic cues. As shown in Fig. 2(b), the keyframe localizer first converts the raw audio into a spectrogram and extract audio features using a pretrained Transformer-based encoder (Girdhar et al., 2023). To better align the audio features with the temporal resolution of motion cues, we modify the patchify stride to increase the number of patches and interpolate the positional embeddings of the encoder (see Appendix A). The audio features are then passed through fully connected layers to predict motion scores. We train the model with $\mathcal{L}_1$ loss between the prediction and the ground-truth motion score calculated by Eq. (1).

Keyframe Selection. Given motion scores $\{M(t)\}_{t=1}^{T}$ of the video frames, we select $T_K \ll T$ keyframes that capture salient motion dynamics with minimal redundancy. Keyframes are identified from local maxima ("peaks") and minima ("valleys"), which indicate dramatic motion changes (Wolf, 1996; Kulhare et al., 2016). We first include the initial frame and sample up to $\frac{T_K}{2} - 1$ peaks; if fewer peaks exist, all are used. For each pair of peaks, we select one valley to preserve motion completeness. The remaining keyframes are obtained by evenly sampling across frame bins. This design ensures robustness to sequences with smooth motion or weak audio cues. Further details and examples are provided in Appendix A and D. We use the selected T_K keyframes to train the keyframe generator and the keyframe indices $\{t_i\}_{i=1}^{T_K}$ serve as additional input conditions.

3.2 Audio-conditioned Keyframe Generation

We propose a novel keyframe generator network to generate T_K keyframes for a video sequence of length T from the input audio and first frame image. Unlike previous video generation models (Xing et al., 2024; Zhang et al., 2024b) that are trained on uniformly downsampled frames, the keyframe generator aims to generate sparse keyframes that captures crucial motions. To enable this, we propose two key designs: (1) *Frame Index Conditioning* - we introduce keyframe index embedding that encodes each frame's absolute position, which provides explicit temporal anchors and ensures coherence when generating non-uniformly distributed frames; (2) *Keyframe-aligned Feature Extraction* - we extract image and audio features that are aligned with the corresponding keyframe time steps to serve as accurate conditions for keyframe generation. In the following, we first provide an overview of the keyframe generator and explain the input conditioning in details.

Overview. We leverage the image dynamic prior of pretrained text-to-video latent diffusion models, and inject the input audio, first frame, and keyframe indices as additional input conditions. The model architecture is shown in Fig. 3(b). We encode the selected keyframes into a latent code $\mathbf{z_0} \in \mathbb{R}^{T_k \times C \times H \times W}$ with a pretrained encoder $\mathcal{E}$, where H and W denotes the spatial dimensions, and C denotes the feature channels. The denoising U-Net learns to iteratively denoise the noisy latent code $\mathbf{z_t}$, and the input conditions are encoded and injected into each denoising U-Net block. The final keyframes are generated from the denoised latent code using the pretrained decoder $\mathcal{D}$.



(a) Keyframe Selection

(b) Keyframe Generation

Figure 3: **Keyframe data selection and keyframe generator**. (a) We select keyframes based on the local maxima and minima of the motion score. (b) The keyframe generator is trained to generate these sparse keyframes conditioned on the audios, first frame image, text, and keyframe indices. These conditions are encoded and passed into the denoising U-Net. In each denoising U-Net block, the index embeddings are added with video features and passed into Residual convolutional block (**Res. Conv.**). The following layers contain a spatial self-attention (**SA**) and spatial cross attention (**CA**) on each three conditional features. The output of each CA is followed by a gating with learnable weights λ_1 and λ_2. Please see details in Sec. 3.2.

Frame Index Embedding. Off-the-shelf video diffusion models assume uniformly sampled frames and cannot directly handle sparsely distributed keyframes. To address this, we introduce a frame index embedding layer that encodes the absolute index of each keyframe $\{t_i\}_{i=1}^{T_K}$ within the original video sequence into frame index embedding $\mathbf{f}_{\text{emb}} \in \mathbb{R}^{T_K \times C}$. $\mathbf{f}_{\text{emb}}$ is added with the latent video features $\mathbf{z}$ before passing into the denoising U-Net blocks, ensuring explicit positional information is provided to the network for global temporal consistency and accurate cross-modal alignment.

Audio Feature Condition. We use a pretrained ImageBind audio encoder (Girdhar et al., 2023) to extract audio features for video synthesis. Given an input spectrogram $\mathbf{A} \in \mathbb{R}^{C_A \times T_A}$, the encoder splits it into overlapping patches of size (c_a, t_a) with a stride $\Delta t < t_a$ and encodes it into a sequence of feature embeddings $\{\mathbf{h}_i\}_{i=1}^{N}$ using Transformer layers. We decrease the patchify stride Δt of the pretrained encoder to obtain finer-grained temporal embeddings. We segment the extracted audio features into T time steps to match the full video length, resulting in $\mathbf{f}_{\text{audio}} \in \mathbb{R}^{T \times C \times M}$, where M is the number of audio features in each time step. Using the keyframe indices $\{i_t\}_{t=1}^{T_K}$, we extract the corresponding T_K audio features from the full T-length sequence and obtain the keyframe-aligned audio features $\mathbf{f}_{\text{audio}}^{\text{key}} = \{\mathbf{f}_{\text{audio}}^{(i_t)}\}_{t=1}^{T_K}$. These keyframe-aligned audio features are fused with text and image conditions via cross-attention layers in the U-Net, ensuring accurate synchronization between generated keyframes and their associated audio cues.

Image Feature Condition. The first frame image $\mathbf{I}$ is injected into the keyframe generation process via two pathways. First, we extract the image feature using a frozen CLIP image encoder (Radford et al., 2021). We project the image features into T frame-specific image conditions using a Q-Former Li et al. (2023a) projection layer, yielding $\mathbf{f}_{\text{img}} \in \mathbb{R}^{T \times C \times H \times W}$. We then select the corresponding T_K features using keyframe indices $\{i_t\}_{t=1}^{T_K}$ to obtain keyframe-aligned image feature $\mathbf{f}_{\text{img}}^{\text{key}} \in \mathbb{R}^{T_K \times C \times H \times W}$. Second, we encode the image with the encoder $\mathcal{E}$, concatenate it with noisy latent code z_t, and feed them to the denoising U-Net. This provides additional visual details from $\mathbf{I}$ to guide the keyframe generation (Xing et al., 2024).

Text Feature Condition. Following prior work, we encode the text prompt of the video using a frozen CLIP text encoder ((Radford et al., 2021). The extracted text embedding $\mathbf{f}_{\text{text}}$ is repeated for all T_K keyframe to provide consistent semantic guidance during the denoising process.

Feature Fusion. Each conditioning feature ($\mathbf{f}^{\text{key}}_{\text{audio}}$, $\mathbf{f}^{\text{key}}_{\text{img}}$, and $\mathbf{f}_{\text{text}}$) is processed separately through spatial cross-attention layers in the U-Net blocks. Given input latent features $\mathbf{F}_{\text{in}}$, we compute query projections $\mathbf{Q} = \mathbf{F}_{\text{in}}\mathbf{W}_Q$ and apply spatial attention to text, image, and audio features:

$$\mathbf{F}_{\text{out}} = \text{SA}(\mathbf{Q}, \mathbf{K}_{\text{text}}, \mathbf{V}_{\text{text}}) + \lambda_1 \cdot \text{SA}(\mathbf{Q}, \mathbf{K}_{\text{audio}}, \mathbf{V}_{\text{audio}}) + \lambda_2 \cdot \text{SA}(\mathbf{Q}, \mathbf{K}_{\text{img}}, \mathbf{V}_{\text{img}}). \quad (2)$$

where SA stands for spatial attention, $\mathbf{K}$ and $\mathbf{V}$ are the key and value projections for each modality, and λ_1, λ_2 are learnable fusion weights. The fused features are then processed through a feedforward network (FFN) and temporal self-attention to ensure spatial and temporal consistency.

3.3 Motion Interpolation

After generating T_K keyframes, we use a *motion interpolator* to generate the missing frames to obtain the a full video sequence of length T. Interpolation has been widely used in uniform frame generation (Blattmann et al., 2023a; Xing et al., 2024), where a model predicts a fixed number of intermediate frames given the first and last frame. However, for keyframe-based generation, the positions of missing and available frames vary, introducing additional challenges. To address this, we adapt our *keyframe generator* diffusion model into a *motion interpolator* model that generates T_K frames at once using masked frame conditioning. The overall architecture remains mostly the same, with the primary difference in how image conditions are incorporated. Rather than conditioning solely on the first frame, the model utilizes the features of generated keyframes as conditions, thereby learning to synthesize the missing frames in between. This approach facilitates interpolation between non-uniformly distributed keyframes while maintaining temporal consistency. Details can be found in Appendix C. To generate a full video with T frames in a single pass, we incorporate FreeNoise (Qiu et al., 2023) to increase the number of output frames during inference. This allows the interpolation model to take all generated keyframes as conditioning inputs and predict all missing frames in one single step. Further details on the training and inference time of this model are provided in the Appendix G.

4 Experiments

4.1 Implementation Details

Datasets. We train and evaluate our method on three datasets: *AVSync15* (Zhang et al., 2024b), *Greatest Hits* (Owens et al., 2016), and *Landscapes* (Lee et al., 2022). *AVSync15* is a subset of the VGG-Sound (Chen et al., 2020) dataset, consisting of fifteen classes of activities with highly synchronized audio and video captured in the wild. Some activities have more intense motions, such as hammer hitting and capgun shooting. *Greatest Hits* contains videos of humans hitting various objects with a drumstick, producing hitting sounds that are temporally aligned with the motions. *Landscapes* is a collection of natural environment videos with corresponding ambient sounds without synchronized video motion. We sample two-second audio-video pairs from these datasets for experiments. Videos were sampled at 24 fps with 48 frames, and resized to 320×512. Audios were sampled at 16kHz and converted into 128-d spectrograms. We set $T_K = 12$ as the temporal length of keyframe generation and interpolations.

Training. We adopted the pre-trained DynamiCrafter (Xing et al., 2024) as the backbone video diffusion model and pre-trained ImageBind (Girdhar et al., 2023) as the audio encoder. All models were trained using Adam optimizer with a batch size of 64 and a learning rate of 1×10^{-5}.

Baselines. We follow (Zhang et al., 2024b) to compare our method with the simple *static* baseline where the input frame is repeated to form a video, as well as state-of-the-art video generation models with different input modalities: **(1) T+A** is the video generation model conditioned only on text and audio, such as TPoS (Jeong et al., 2023) and TempoToken (Yariv et al., 2024). **(2) I+T** includes many state-of-the-art video generation models, which are conditioned on images and text prompts. We compare with I2VD (Zhang et al., 2024b), VideoCrafter (Chen et al., 2023a) and DynamiCrafter (Xing et al., 2024). **(3) I+T+A** takes image, text and audio inputs for video generation, which includes CoDi (Tang et al., 2023), TPoS (Jeong et al., 2023), AADiff (Lee et al., 2023) and AVSyncD (Zhang et al., 2024b).

Table 1: Performance on the *AVSync15* and the *Greatest Hits* datasets. Best is marked in **bold**.

Input	Model	AVSync15						Greatest Hits					
		FID↓	**IA↑**	**IT↑**	**FVD↓**	**AlignSync↑**	**RelSync↑**	**FID↓**	**IA↑**	**IT↑**	**FVD↓**	**AlignSync↑**	**RelSync↑**
T+A	TPoS	13.5	23.38	24.83	2671.0	19.52	42.50	33.85	11.50	**17.90**	3327.90	21.48	44.90
	TempoToken	12.2	18.84	17.45	4466.4	19.74	44.05	25.90	4.88	9.28	3300.53	21.56	45.38
I+T	I2VD	12.1	-	30.35	398.2	21.80	43.92	9.10	-	13.42	425.0	22.05	44.58
	DynamiCrafter	11.7	-	30.02	400.7	21.76	43.68	12.40	-	13.73	337.71	22.82	45.85
I+T+A	CoDi	14.5	28.15	23.42	1522.6	19.54	41.51	21.78	12.01	14.11	1336.00	22.30	45.35
	TPoS	11.9	38.36	**30.73**	1227.8	19.67	39.62	28.43	9.36	13.19	1370.57	22.04	45.55
	AADiff	18.8	34.23	28.97	978.0	22.11	45.48	-	-	-	-	-	-
	AVSyncD	11.7	38.53	30.45	349.1	22.62	45.52	**8.70**	12.07	13.31	249.30	22.83	45.95
	KeyVID (Ours)	**11.1**	**39.21**	30.12	**263.3**	**24.44**	**49.06**	12.10	**12.40**	**15.66**	**202.10**	**22.91**	**46.03**
	Static	-	39.76	30.39	1220.4	21.83	43.66	-	13.33	16.56	348.9	24.36	48.73
	Groundtruth	-	40.06	30.31	-	25.04	50.00	-	13.52	16.49	-	25.02	50.00

Metrics. We use the Frechet Image Distance (**FID**) (Heusel et al., 2017) and Frechet Video Distance (**FVD**) (Unterthiner et al., 2018) to evaluate the visual quality of the individual frames and videos. We also compare the average image-text (**IT**) and image-audio (**IA**) semantic alignment scores of video frames using CLIP (Radford et al., 2021) and ImageBind (Girdhar et al., 2023). To measure audio-video synchronization, we evaluate the generated videos with **RelSync** and **AlignSync** proposed by Zhang et al. (2024b).

4.2 Quantitative Results

Table 1 presents the quantitative evaluation results on the *AVSync15* and *Greatest Hits* datasets. Results on the *Landscape* dataset can be found in the Appendix **??**. On the *AVSync15* dataset, KeyVID demonstrates superior performance across both audio-visual synchronization and visual quality metrics. It achieves the highest synchronization scores with AlignSync of 24.44 and RelSync of 49.06, substantially outperforming the previous state-of-the-art AVSyncD (22.62 and 45.52, respectively). These improvements highlight the effectiveness of our keyframe-aware strategy in capturing critical dynamic moments that align with audio events. In terms of visual quality, KeyVID also excels with an FID score of 11.00 and FVD score of 263.3, representing the best performance among all compared methods. Additionally, our approach achieves the highest image-audio semantic alignment score (IA: 39.21), demonstrating strong correspondence between generated visual content and audio input. The *Greatest Hits* dataset presents a particularly challenging scenario with distinct percussive audio events that require precise temporal alignment with visual motions. KeyVID achieves competitive performance across all evaluation metrics. Notably, KeyVID attains the best FVD score of 202.10, indicating superior visual quality in the generated videos. For audio-visual synchronization, KeyVID achieves AlignSync and RelSync scores of 22.91 and 46.03, respectively, outperforming most baseline methods while maintaining strong visual quality with competitive FID performance.

4.3 Ablation Study

Keyframe vs. Uniform Sampling. To validate the effectiveness of keyframe-aware generation, we compare KeyVID with a uniform sampling baseline, **KeyVID-Uniform**, where KeyVID-Uniform generates 12 uniform frames instead of keyframes before motion interpolation.As shown in Table 2, KeyVID consistently outperforms KeyVID-Uniform across all metrics, with larger improvements in audio-visual synchronization scores AlignSync and RelSync, while maintaining competitive visual quality metrics. In addition, KeyVID achieves greater improvement in high-intensity motion scenarios as shown in Fig. 5. These results confirm our hypothesis that strategically selecting keyframes based on audio and motion cues leads to superior audio-visual synchronization.

Table 2: Ablation study results on *AVSync15*.

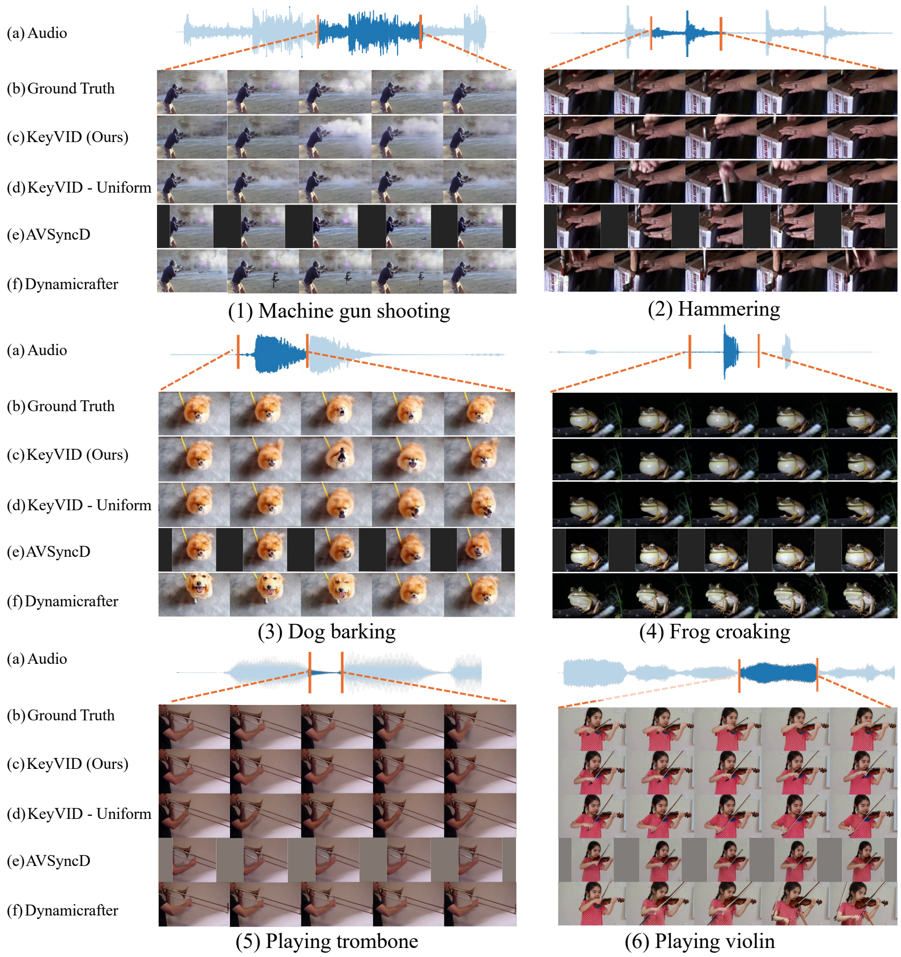
Setting	FID↓	FVD↓	AlignSync↑	RelSync↑
KeyVID	11.1	263.3	**24.44**	**49.06**
KeyVID-Uniform	**11.0** (-0.9%)	273.4 (+3.8%)	23.53 (-3.7%)	47.23 (-3.7%)
w/o *Frame Index*	**11.0** (-0.9%)	**258.9** (-1.7%)	23.93 (-2.1%)	47.90 (-2.4%)
w/o *First Frame*	11.7 (+5.4%)	265.5 (+0.8%)	24.02 (-1.7%)	48.49 (-1.2%)

Frame Conditioning. We further analyze the contribution of two components in our frame conditioning mechanism in Table 2. Removing the frame index embedding leads to degraded audio-visual synchronization, with AlignSync and RelSync scores decreasing by 2.1% and 2.4%, respectively. This demonstrates that frame index embedding provides crucial temporal information that helps the model understand the sequential ordering of keyframes during generation.

Removing the first-frame condition from the motion interpolator results in significant performance degradation, particularly in visual quality metrics. The FID increases by 5.4% and FVD increases by 0.80%, indicating that the first frame serves as an essential reference for maintaining visual consistency during interpolation. The combination of both components achieves optimal performance, confirming the importance of our complete frame conditioning design.

4.4 Visualization

Fig. 4 presents qualitative comparisons between KeyVID and baseline approaches on different type of motions. Our keyframe-aware approach more accurately captures motion peaks that align with audio events, such as the exact moment of impact in hammering or the smoke in gun shooting. Compared to the uniform frame sampling variant KeyVID-Uniform, KeyVID better preserves temporal coherence by focusing on key moments of motion. In sequences like dog barking and frog croaking, KeyVID ensures that mouth movements align precisely with sound peaks, whereas KeyVID-Uniform and AVSyncD introduce temporal misalignment or missing frames. For subtle motions, such as playing the trombone or violin, our model still produces smooth and stable movements, even during sustained notes or brief pauses in the audio, where motion cues are weak and baselines tend to jitter or freeze. Additional video visualizations for intensive, moderate, and subtle motions are provided in the supplementary material.



(1) Machine gun shooting

(2) Hammering

(3) Dog barking

(4) Frog croaking

(5) Playing trombone

(6) Playing violin

Figure 4: **Qualitative comparison of KeyVID and baseline methods**. We crop key motions on the audio waveform in (a) and the corresponding ground truth video in (b) as references and compare the generated video clips between models from (c) to (f). KeyVID with keyframe awareness (c) shows better alignment with motion peaks in audio signals—for example, the hammer striking, gunshots producing smoke, or facial movements when dogs bark or frogs croak. For subtle motion scenarios such as playing the trombone or violin, our model is also able to produce smooth, stable motion during the brief pauses or sustained notes.

4.5 Effects of Motion Intensity

To analyze how KeyVID performs across different motion types, we categorize the 15 classes in the AVSync15 dataset into three intensity levels based on their average motion scores: *Subtle*, *Moderate*, and *Intense*, with five classes each. The *Intense* level includes highly dynamic motions such as hammering and dog barking, while the *Subtle* level consists of activities with slow movement, such as playing the violin or trumpet. Fig. 5 compares RelSync scores across these motion intensities for KeyVID, KeyVID-Uniform, and AVSyncD. KeyVID shows increasing improvements over KeyVID-Uniform as motion intensity rises, with RelSync gains of 1.50, 1.59, and 2.01 for *Subtle*, *Moderate*, and *Intense* motions, respectively. This demonstrates the effectiveness of keyframes in capturing audio rapid motion transitions Compared to AVSyncD, KeyVID consistently achieves superior synchronization with RelSync gains of 3.86, 3.18, and 3.07 across all intensity levels.

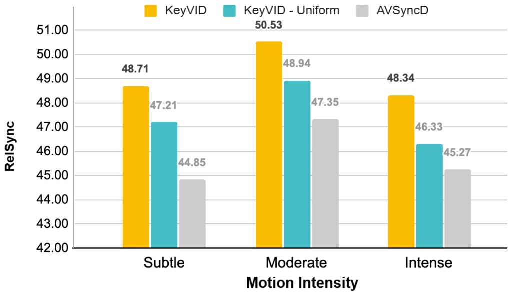


Figure 5: **RelSync scores across motion intensity levels**. **KeyVID** improves audio synchronization score on all motion intensity.

Table 3: **User study results**. Participants voted for the best method based on audio synchronization (**AS**), visual quality (**VQ**), and temporal consistency (**TC**). The numbers represent the percentage of votes each model received for each metric.

Models	AS	VQ	TC
KeyVID	**66.25%**	**65.00%**	**65.00%**
KeyVID-Uniform	17.92%	22.08%	21.67%
AVSyncD	11.67%	7.08%	7.92%
DynamiCrafter	4.17%	5.83%	5.42%

4.6 User Study

We conducted a user study with twelve participants to assess the quality of generated videos. Each participant was shown twenty randomly selected video samples, where each sample contained results from four models presented in a random order with the same inputs. They were asked to choose which video exhibited better audio-visual synchronization, visual quality, and temporal consistency. We aggregated all $12 \times 20 = 240$ votes for each metric and computed the percentage of votes each model received, as shown in Tab. 3. Further details on the user study can be found in Appendix F.

4.7 Open-Domain Audio-Synchronized Visual Animation

We show KeyVID's ability to animate open-domain inputs beyond its training distribution. As illustrated in Fig. 6, we use the first frame from a Sora-generated video clip, where a hammer is held in the air before striking down. We control the visual animation through two distinct hammering audio clips: the first contains metallic strike sounds, while the second captures impacts on a wooden surface. Our model not only successfully generates videos that match the temporal pattern of strikes, but also adapts the motion based on the material properties inferred from the audio: the first video shows hammering on metal nails, while the second shows hammering on a wooden table. These results demonstrate the generalization capability of KeyVID to open-domain inputs and its ability to accurately follow the audio semantics for visual animation.

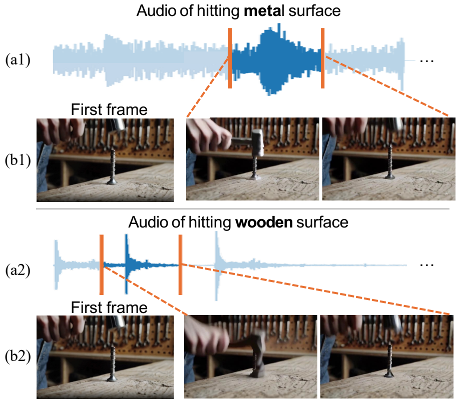

Audio of hitting metal surface
(a1)
...
First frame
(b1)
Audio of hitting wooden surface
(a2)
...
First frame
(b2)

Figure 6: **Open-domain video generation**. Given the same first frame and different audio inputs (a1) and (a2), KeyVID synthesizes videos that align with the audio's semantic meaning and motion pattern in (b1) and (b2).

5 Conclusion

In this paper, we introduced a keyframe-aware audio-synchronized visual animation model which enhances video generation quality and audio alignment, particularly for highly dynamic motions.

Our approach first localizes keyframes from audio and generates corresponding frames using a diffusion model. Then we synthesize intermediate frames to obtain smooth high-frame-rate videos while maintaining memory efficiency. Experimental results demonstrate superior performance across multiple datasets, especially in scenarios with intensive motion. Compared to previous methods, our model significantly improves audio-visual synchronization and visual quality.

Acknowledgement. Although our method does not rely on simple amplitude-based cues, the *keyframe localizer* inevitably learns data-driven statistical correspondences between audio signals and motion change patterns, since it is supervised using optical flow derived motion scores.

References

Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos. In *ECCV*, 2024.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023b.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023a.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models, 2024.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020.

Rui Chen, Yixiao Li, Yifan Zhang, Hao Wang, and Yun Fu. Customizing text-to-video generation with multiple subjects. *arXiv preprint arXiv:2307.23456*, 2023b.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP*, 2023.

Weichen Fan, Chenyang Si, Junhao Song, Zhenyu Yang, Yinan He, Long Zhuo, Ziqi Huang, Ziyue Dong, Jingwen He, Dongwei Pan, et al. Vchitect-2.0: Parallel transformer for scaling up video diffusion models. *arXiv preprint arXiv:2501.08453*, 2025.

Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *ICML*, 2020.

Zichen Geng, Caren Han, Zeeshan Hayder, Jian Liu, Mubarak Shah, and Ajmal Mian. Text-guided 3d human motion generation with keyframe-based parallel skip transformer. *arXiv preprint arXiv:2405.15439*, 2024.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022b.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Siddhant Jain, Daniel Watson, Eric Tabellion, Ben Poole, Janne Kontkanen, et al. Video interpolation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7341–7351, 2024.

Youngjae Jeong, Won Jeong Ryoo, Seung Hyun Lee, Donghyeon Seo, Wonmin Byeon, Sangpil Kim, and Jinkyu Kim. The power of sound (TPoS): Audio reactive video generation with stable diffusion. In *ICCV*, 2023.

Sung-Bin Kim, Arda Senocak, Hyunwoo Ha, Andrew Owens, and Tae-Hyun Oh. Sound to visual scene generation by audio-to-visual latent alignment. In *CVPR*, 2023.

Sourabh Kulhare, Shagan Sah, Suhas Pillai, and Raymond Ptucha. Key frame extraction for salient activity recognition. In *ICPR*, 2016.

Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J Kim. Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers. In *CVPR*, 2024.

Seung Hyun Lee, Gyeongrok Oh, Wonmin Byeon, Chanyoung Kim, Won Jeong Ryoo, Sang Ho Yoon, Hyunjun Cho, Jihyun Bae, Jinkyu Kim, and Sangpil Kim. Sound-guided semantic video generation. In *ECCV*, 2022.

Seungwoo Lee, Chaerin Kong, Donghyeon Jeon, and Nojun Kwak. Aadiff: Audio-aligned video synthesis with text-to-image diffusion. *arXiv preprint arXiv:2305.04001*, 2023.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023a.

Yixiao Li, Hao Wang, Yifan Zhang, and Yun Fu. ID-Animator: Zero-shot identity-preserving human video generation. *arXiv preprint arXiv:2306.67890*, 2023b.

Haiyang Liu, Xingchao Yang, Tomoya Akiyama, Yuantian Huang, Qiaoge Li, Shigeru Kuriyama, and Takafumi Taketomi. Tango: Co-speech gesture video reenactment with hierarchical audio motion embedding and diffusion interpolation. *arXiv preprint arXiv:2410.04221*, 2024.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. *arXiv preprint arXiv:2303.08320*, 2023.

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

Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *arXiv preprint arXiv:2310.15169*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

Alexander Richard, Evgenia Egorova, Stanimir Matuszewski, Florian Bernard, Jürgen Gall, and Gerard Pons-Moll. Audio-driven 3d facial animation from in-the-wild videos. *arXiv preprint arXiv:2306.11541*, 2023. URL `https://arxiv.org/abs/2306.11541`.

Ludan Ruan, Yunzhi Ma, Hongjie Yang, Haoxian He, Bing Liu, Jianlong Fu, Nenghai Yuan, Qin Jin, and Bing Guo. MM-Diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *CVPR*, 2023.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023.

Xusen Sun, Longhao Zhang, Hao Zhu, Peng Zhang, Bang Zhang, Xinya Ji, Kangneng Zhou, Daiheng Gao, Liefeng Bo, and Xun Cao. Vividtalk: One-shot audio-driven talking head generation based on 3d hybrid prior. *arXiv preprint arXiv:2312.01841*, 2023.

Kim Sung-Bin, Lee Chae-Yeon, Gihun Son, Oh Hyun-Bin, Janghoon Ju, Suekyeong Nam, and Tae-Hyun Oh. Multitalk: Enhancing 3d talking head generation across languages with multilingual video dataset. *arXiv preprint arXiv:2406.14272*, 2024.

Zhaohan Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. CoDi: Any-to-any generation via composable diffusion. *arXiv preprint arXiv:2305.11846*, 2023. URL `https://arxiv.org/abs/2305.11846`.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *NeurIPS*, 2022.

Yujie Wei, Shiwei Zhang, Zhiwu Qing, Hangjie Yuan, Zhiheng Liu, Yu Liu, Yingya Zhang, Jingren Zhou, and Hongming Shan. Dreamvideo: Composing your dream videos with customized subject and motion. *arXiv preprint arXiv:2312.04433*, 2023. URL `https://arxiv.org/abs/2312.04433`.

W. Wolf. Key frame selection by motion analysis. In *ICASSP*, 1996.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExT-GPT: Any-to-any multimodal large language model. *arXiv preprint arXiv:2310.14547*, 2023. URL `https://arxiv.org/abs/2310.14547`.

Tao Wu, Yong Zhang, Xintao Wang, Xianpan Zhou, Guangcong Zheng, Zhongang Qi, Ying Shan, and Xi Li. Customcrafter: Customized video generation with preserving motion and concept composition abilities. *arXiv preprint arXiv:2408.13239*, 2024a.

Yifan Wu, Zhen Li, and Lei Zhao. Takin-ada: Towards high-quality audio-driven talking head generation. *arXiv preprint arXiv:2410.14283*, 2024b. URL `https://arxiv.org/abs/2410.14283`.

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *ECCV*, 2024.

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, and Runsheng Xu. Diffusion-based video generation with image prompts. *arXiv preprint arXiv:2305.12345*, 2023.

Xiaodong Yang, Yixiao Li, Yifan Zhang, and Yun Fu. Cogvideox: Extending video generation with advanced controls. *arXiv preprint arXiv:2403.34567*, 2024.

Guy Yariv, Itai Gat, Sagie Benaim, Lior Wolf, Idan Schwartz, and Yossi Adi. Diverse and aligned audio-to-video generation via text-to-video model adaptation. In *AAAI*, 2024.

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, Jianlong Fu, Gong Ming, Lijuan Wang, Zicheng Liu, Houqiang Li, and Nan Duan. NUWA-XL: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023. URL `https://arxiv.org/abs/2303.12346`.

David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023.

Hao Zhang, Qian Jiang, Xiang Li, and Hao Wang. Lingualinker: Multilingual audio-driven talking head synthesis. *arXiv preprint arXiv:2407.18595*, 2024a. URL `https://arxiv.org/abs/2407.18595`.

Jiangning Zhang, Chao Xu, Liang Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. DTVNet: Dynamic time-lapse video generation via single still image. In *ECCV*, 2020.

Lin Zhang, Shentong Mo, Yijing Zhang, and Pedro Morgado. Audio-synchronized visual animation. In *ECCV*, 2024b.

Mingzhe Zheng, Yongqi Xu, Haojian Huang, Xuran Ma, Yexin Liu, Wenjie Shu, Yatian Pang, Feilong Tang, Qifeng Chen, Harry Yang, and Ser-Nam Lim. Videogen-of-thought: A collaborative framework for multi-shot video generation. *arXiv preprint arXiv:2412.02259*, 2024. URL `https://arxiv.org/abs/2412.02259`.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

Appendix

Table of Contents

A Details of Keyframe Localizer

In the Sec. 3.1 of the main paper, we introduce that we need to know the position of the key frame at the beginning of inference by predicting optical motion scores. Here is the detailed structure of this network. The network processes raw audio by converting it into a spectrogram $\mathbf{A} \in \mathbb{R}^{C_A \times T_A}$, where C_A denotes the number of frequency channels and T_A represents the temporal length. The original ImageBind preprocessing pipeline applies a CNN with a kernel stride of $(10, 10)$ to patchify the input spectrogram, producing feature embeddings that are then processed by a transformer-based encoder $f_{\text{audio}} \in \mathbb{R}^{B \times T \times C}$. However, this results in T (*e.g.*, T=19) being misaligned with the temporal resolution of the dense motion curve sequence (*e.g.*, 48).

To address this, we modify the CNN stride to $(10, 4)$, increasing the temporal resolution of extracted features (*e.g.*, increase to 46). The transformer encoder then processes the updated feature sequence:

$$\mathbf{F}_{\text{audio}} = f_{\text{audio}}(\mathbf{A}), \quad \mathbf{F}_{\text{audio}} \in \mathbb{R}^{B \times T' \times C}, \tag{3}$$

where $T' > T$ reflects the increased temporal resolution. Since the transformer relies on positional embeddings, we interpolate the pretrained positional embeddings to match the new sequence length T'_A and keep them frozen during training.

The extracted features are passed through fully connected layers to predict a sequence of confidence scores $\mathbf{s} \in \mathbb{R}^{B \times T'}$, where each s_t represents the likelihood of a keyframe occurring at time step t:

$$\mathbf{s} = \sigma(\mathbf{W}\mathbf{F}_{\text{audio}} + \mathbf{b}), \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{C \times 1}$ and $\mathbf{b} \in \mathbb{R}^{T'_A}$ are learnable parameters, and $\sigma(\cdot)$ is the sigmoid activation function. The model is trained using an L1 loss:

$$\mathcal{L} = \left\| \mathbf{s} - \hat{\mathbf{s}} \right\|_1, \tag{5}$$

where $\hat{\mathbf{s}}$ represents the ground-truth keyframe labels derived from optical flow analysis.

B Details of Keyframe Selection

B.1 Detect Peak and Valley

To identify the local maxima (*peaks*) and minima (*valleys*) from a one-dimensional motion score $\{M(t)\}_{t=1}^{T}$, we perform the following steps:

1. **Smoothing**: Convolve the raw score $M(t)$ with a short averaging filter with a window size 5, producing a smoothed label $\widetilde{M}(t)$. This helps reduce noise and minor fluctuations.
2. **Peak Detection**: Finds all local maxima by simple comparison of neighboring values for $\widetilde{M}(t)$. We force a minimum distance of 5 frames between any two detected peaks and require a prominence (height relative to its surroundings) of at least 0.1. This returns the indices of the local maxima.
3. **Valley Detection**: Repeat the same peak-finding procedure on the negative of the smoothed signal.

B.2 Sample Keyframes

In the main text, we discuss the process of selecting $T_K \ll T$ keyframes based on the motion score $M(t)$ for each frame. Specifically, we first pick the initial frame, then select up to $\frac{T_K}{2} - 1$ peaks among all detected ones (or all peaks if fewer are found). Next, we include a valley between each consecutive pair of selected peaks. Finally, we sample any remaining frames by an evenly distributed (proportional) strategy, which approximates uniform downsampling if few peaks and valleys are present. This approach ensures that smooth motion or weak audio signals, producing limited peaks and valleys, do not degrade the consistency of training for video diffusion models.

Algorithm 1 is the detailed pseudo-code for the full procedure, including both peak and valley selection and the final proportional allocation of remaining key frames.

Algorithm 1: Keyframe Selection Algorithm

```
Input: Motion scores $\{M(t)\}_{t=1}^{T}$, desired keyframe count $T_K \ll T$.
Output: A set of $T_K$ keyframes.
Step 1: Detect peaks and valleys based on $M(t)$.
Step 2: Initialize keyframe list:
    Keyframes $\leftarrow$ {first_frame}.
Step 3: Randomly select peaks
    Choose up to $\left\lfloor \frac{T_K}{2} - 1 \right\rfloor$ from the detected peaks and add to Keyframes.
Step 4: Insert valleys
    for each pair of consecutive peaks in Keyframes do
        Select one valley in between and add it to Keyframes.
Step 5: Compute how many more keyframes are needed:
    $R \leftarrow T_K - |\text{Keyframes}|$.
if $R > 0$ then
    Define a list of $N$ remaining frames (unselected) with some weights $\{w_1, \ldots, w_N\}$.
    $W \leftarrow \sum_{i=1}^{N} w_i$
    for $i \leftarrow 1$ to $N$ do
        $\text{ideal\_share}_i \leftarrow R \cdot \frac{w_i}{W}$;
        $\text{allocated}_i \leftarrow \lfloor \text{ideal\_share}_i \rfloor$;
    $r \leftarrow R - \sum_{i=1}^{N} \text{allocated}_i$ ;          // Remainder after flooring
    if $r > 0$ then
        for $i \leftarrow 1$ to $N$ do
            $\text{frac}_i \leftarrow \text{ideal\_share}_i - \text{allocated}_i$;
        Sort frames by $\text{frac}_i$ in descending order.
        for $j \leftarrow 1$ to $r$ do
            $i^* \leftarrow$ index of the $j$-th largest $\text{frac}_i$;
            $\text{allocated}_{i^*} \leftarrow \text{allocated}_{i^*} + 1$;
    for $i \leftarrow 1$ to $N$ do
        if $\textit{allocated}_i > 0$ then
            Keyframes $\leftarrow$ Keyframes $\cup \{\text{frame}_i\}$;
return Keyframes
```

C Structure of Motion Interpolation

As shown in Fig. 7, we present the pipeline of motion interpolation network as introduced in Sec. 3.3

D Motion score prediction evaluation

Quantitative result. We evaluate the keypoint detected from the predicted motion score with the ground truth score. We calculate the average precision with a distance threshold t. In this way, for each keypoint in ground truth motion score curve, if it can match with a predicted keypoint with distance lower than t, it will be consider as a successful match. The average precision means the the average of $N_{match}/N(total)$ across all instance, denoted as $AP@t$ We achieve the $AP@3 = 60.57\%$ and $AP@5 = 77.92\%$.

Visualization. We provide visualization of modtion score prediction in Fig. 8.

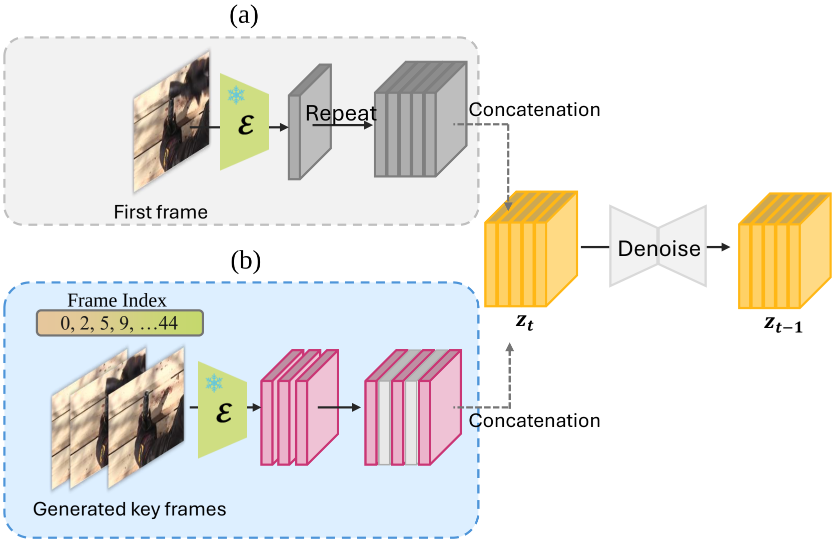


Figure 7: The frame interpolation model shares the same structure as the original keyframe generation model but uses different image features for concatenation. (a) For keyframe generation (Sec. 3.2), the first-frame features are repeated to match the length of the latent vector; (b) For frame interpolation, the condition features from keyframes are padded with zero tensors between keyframe locations to align with the frame length.

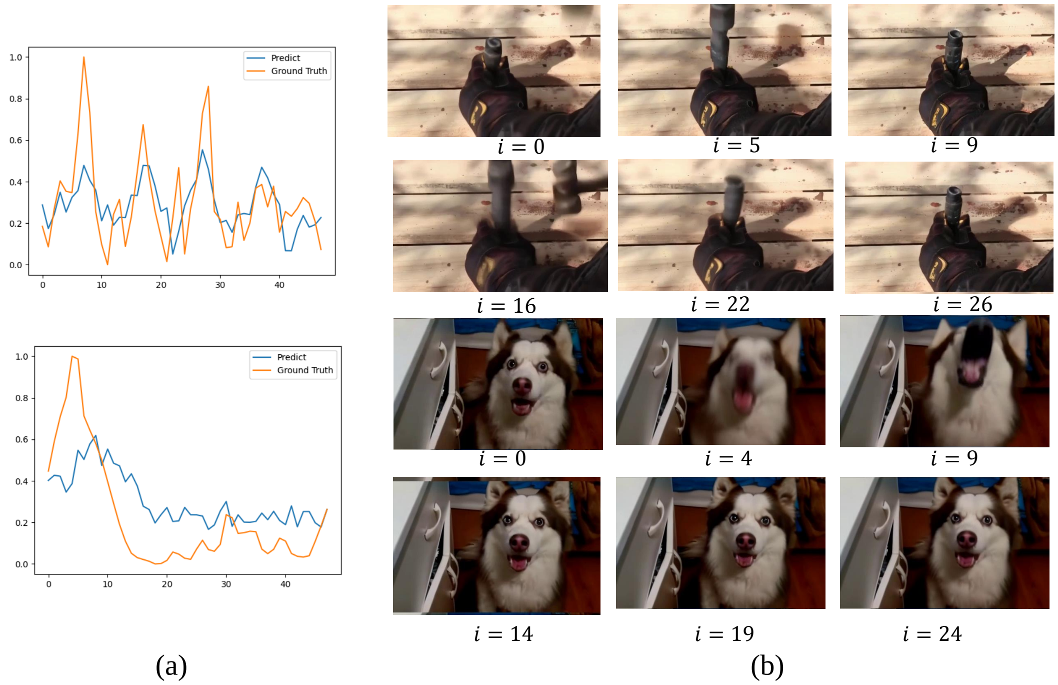


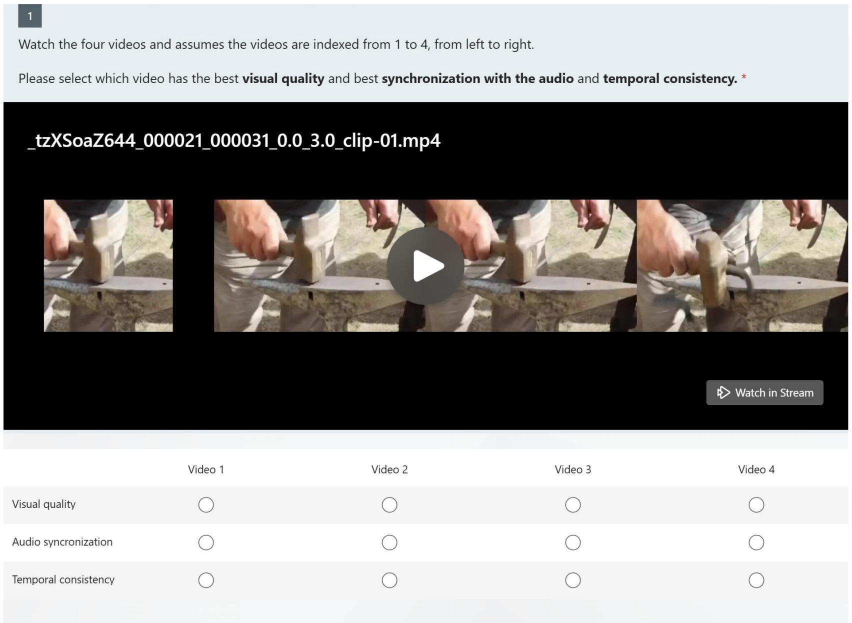
Figure 8: Visualization of (a) Predicted motion score from audio with the ground truth caluate from video data; and (b) the generated video keyframe by diffusion network described in Sec. 3.2 before interpolations.

E More Qualitative Results of Video Generation

As the generation result need to be watch with audio for the best experience, we have put more visualization result into the supplementary as mp4 files.

F Details of User Study

As described in the main paper (Sec. 4.6), we conduct a user study to evaluate the performance of four video generation models in terms of audio synchronization, visual quality, and temporal frame consistency. We invite 12 participants and design an online survey to collect responses. In the survey, we randomly select 20 video instances and present the generation results from four models—KeyVID, KeyVID-Uniform, AVSyncD, and Dynamicrafter—in a row for comparison, with the order randomly shuffled. The videos generated by KeyVID, KeyVID-Uniform, and AVSyncD use the same audio, image, and text conditions, whereas Dynamicrafter generates videos using only text and image conditions. For each instance, participants are asked to select the best video based on three evaluation metrics. This results in a total of $20 \times 12 = 240$ votes for each metric across all models. Sample survey questions are illustrated in Fig. 9.



1

Watch the four videos and assumes the videos are indexed from 1 to 4, from left to right.

Please select which video has the best **visual quality** and best **synchronization with the audio** and **temporal consistency.** *

_tzXSoaZ644_000021_000031_0.0_3.0_clip-01.mp4

	Video 1	Video 2	Video 3	Video 4
Visual quality	○	○	○	○
Audio syncronization	○	○	○	○
Temporal consistency	○	○	○	○

Figure 9: Sample survey question used in the user study.

G Experimental Details

For the experiments of KeyVID on the three datasets *AVSyncD*, *Landscape* , and *TheGreatestHit*, we train at a resolution of 320×512, following Dynamicrafter Xing et al. (2024). During inference, we use DDIM sampling with 90 steps. The temporal length of both the keyframe generation and interpolation models is 12. Since our interpolation module adopts the FreeNoise Qiu et al. (2023) technique, we are able to generate the final 48 frames in a single run. To accommodate this temporal length, we set the window size to 12 and the stride to 6.

H Multimodal Classifier Free Guidance

Similar to Xing et al. (2024), we introduce three guidance scales s_{img}, s_{txt}, and s_{aud} to extend video generation with additional audio control. These scales allow balancing the influence of different conditioning modalities in video generation. The modified noise estimation function is defined as:

$$\begin{aligned}\hat{\epsilon}_\theta\left(\mathbf{z}_t, \mathbf{c}_{\text{img}}, \mathbf{c}_{\text{txt}}, \mathbf{c}_{\text{aud}}\right) = \epsilon_\theta\left(\mathbf{z}_t, \varnothing, \varnothing, \varnothing\right) \\ +s_{\text{img}}\left(\epsilon_\theta\left(\mathbf{z}_t, \mathbf{c}_{\text{img}}, \varnothing, \varnothing\right) - \epsilon_\theta\left(\mathbf{z}_t, \varnothing, \varnothing, \varnothing\right)\right) \\ +s_{\text{txt}}\left(\epsilon_\theta\left(\mathbf{z}_t, \mathbf{c}_{\text{img}}, \mathbf{c}_{\text{txt}}, \varnothing\right) - \epsilon_\theta\left(\mathbf{z}_t, \mathbf{c}_{\text{img}}, \varnothing, \varnothing\right)\right)\end{aligned} \tag{6}$$

$$+s_{\text{aud}}\left(\epsilon_\theta\left(\mathbf{z}_t, \mathbf{c}_{\text{img}}, \mathbf{c}_{\text{txt}}, \mathbf{c}_{\text{aud}}\right) - \epsilon_\theta\left(\mathbf{z}_t, \mathbf{c}_{\text{img}}, \mathbf{c}_{\text{txt}}, \varnothing\right)\right).$$

Here, $\mathbf{c}_{\text{img}}$, $\mathbf{c}_{\text{txt}}$, and $\mathbf{c}_{\text{aud}}$ represent image, text, and audio conditioning, respectively. The newly introduced audio guidance scale s_{aud} enables the model to integrate temporal audio cues, ensuring synchronized motion generation in audio-reactive video synthesis. By adjusting these guidance parameters, we can control the relative impact of each modality in the final video output.

In our experiments, we set the audio guidance scale to 7.5 and the image guidance scale to 2.0 for both the keyframe generation and frame interpolation networks. Since audio guidance is introduced as a new feature, we further compare results across different audio guidance scales ranging from 4.0 to 11.0, as shown in Tab. 4. While higher audio guidance values yield better audio synchronization scores (RelSync and AlignSync), we ultimately select the configuration that provides the best visual quality (FVD and FID) while still achieving competitive audio synchronization performance.

Table 4: Performance metrics for different guidance values.

s_{aud}	**FID↓**	**FVD↓**	**AlignSync↑**	**RelSync↑**
4.0	11.4	270.5	48.18	24.14
7.5	**11.0**	**262.3**	48.33	24.08
9.0	11.1	277.2	48.55	24.16
11.0	11.1	278.6	**48.66**	**24.22**

I Details of Motion Intensity

To analyze motion intensity in AVSyncD, we cluster 15 classes based on their average motion scores across all instances (motion score result in Tab. reftab:motion). The classes are grouped into three motion intensity levels:

- **Subtle**: playing trumpet, playing violin, playing cello, playing trombone, toilet flushing.
- **Moderate**: lions roaring, cap gun shooting, frog croaking, chicken crowing, baby crying.
- **Intensive**: striking bowling, dog barking, hammering, sharpening knife, machine gun.

This classification provides insights into motion intensity distribution within AVSyncD, aiding in evaluating synchronization across different motion levels.

Subtle Motion					
Classes	playing trumpet	toilet flushing	playing cello	playing violin	playing trombone
Motion Score	2.79	4.13	5.32	5.56	6.24
Moderate Motion					
Classes	cap gun shooting	chicken crowing	lions roaring	frog croaking	baby crying
Motion Score	8.47	8.54	9.16	9.24	10.45
Intensive Motion					
Classes	machine gun	sharpen knife	striking bowling	dog barking	hammering
Motion Score	13.16	16.74	17.16	20.32	32.70

Table 5: Motion categories with classes and motion scores.

J LLM Usage

We used large language models (LLMs) to assist in the preparation of this paper. Their role was limited to language editing such as proofreading and rephrasing. All ideas, experiments, and analyses were conceived and conducted by the authors.