The Power of Sound (TPoS): Audio Reactive Video Generation with Stable Diffusion

Yujin Jeong[1], Wonjeong Ryoo[2], Seunghyun Lee[2], Dabin Seo[1],
Wonmin Byeon[3], Sangpil Kim,[2,*] and Jinkyu Kim[1,*]
[1] Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea
[2] Department of Artificial Intelligence, Korea University, Seoul 02841, Korea
[3] NVIDIA Research, Santa Clara 95050, USA
[*]Correspondences: S. Kim (spk7@korea.ac.kr) and J. Kim (jinkyukim@korea.ac.kr)

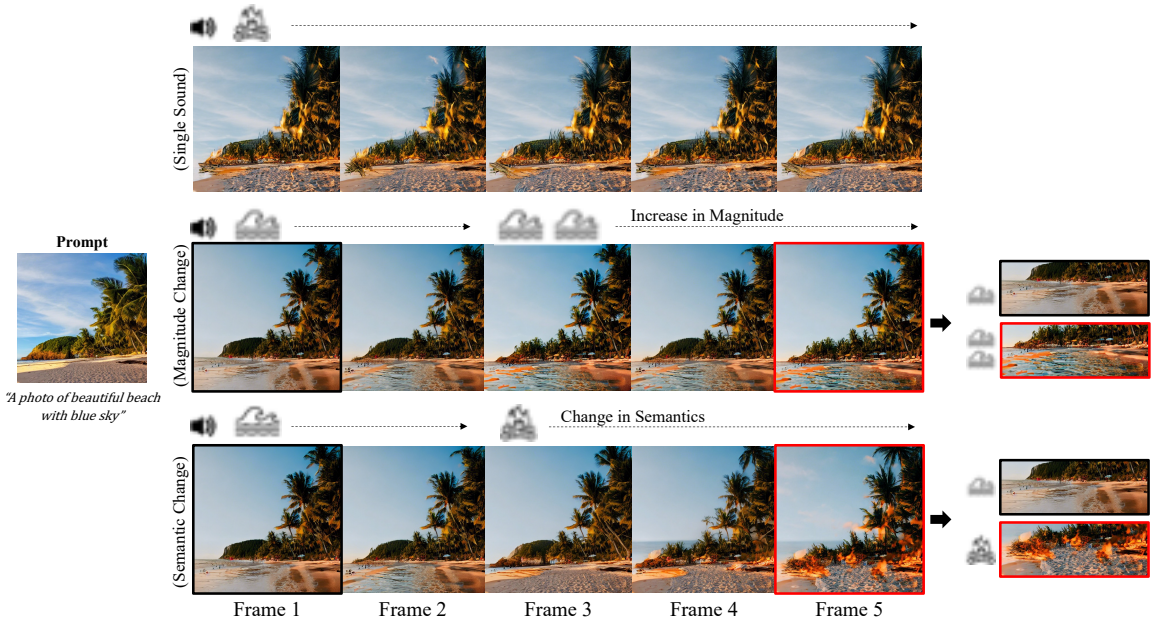

Figure 1: The Power of Sound (TPoS) is a novel framework that generates audio-reactive video sequences. Built upon the Stable Diffusion model, our model first generates an initial frame from a user-provided text prompt (e.g. "a photo of a beautiful beach with a blue sky"), then reactively manipulates the style of generated images corresponding to the sound inputs (e.g. an audio sequence of fireplace). Our model is indeed able to generate a frame conditioned on semantic information of the sound (see 1st and 2nd rows where images are manipulated driven by sound inputs such as fireplace or wave sound), while realistically dealing with temporal visual changes conditioned on changes of sound, e.g., increasing magnitude of sounds (see second row) or wave → fireplace (see last row). TPoS creates visually compelling and contextually relevant video sequences in an open domain.

Abstract

In recent years, video generation has become a prominent generative tool and has drawn significant attention. However, there is little consideration in audio-to-video generation, though audio contains unique qualities like temporal semantics and magnitude. Hence, we propose The Power of Sound (TPoS) model to incorporate audio input that includes both changeable temporal semantics and magnitude. To generate video frames, TPoS utilizes a latent stable diffusion model with textual semantic information, which is then guided by the sequential audio embedding from our pretrained Audio Encoder. As a result, this method produces audio reactive video contents. We demonstrate the effectiveness of TPoS across various tasks and compare its results with current state-of-the-art techniques in the field of audio-to-video generation. More examples are available

at https://ku-vai.github.io/TPoS/

1. Introduction

Recent generative models have demonstrated the potential to generate visually-appealing video frames [32, 11, 10, 23, 36]. They often use a simple text prompt (e.g., "a video of a person on the street on a rainy day") to generate a video which is intuitive for end-users to drive a video generation. Text can effectively convey uni-modal object-wise guidance, such as "a rainy day" or "a person", but it may be challenging if users want to drive it into more complex sequential procedures, i.e., "a video of a person on the street on a rainy day, but a rain suddenly stops, and a wind blows."

In this paper, we leverage sounds to guide the video generation models, i.e., sound-driven video generation. Audio is another modality that can complement texts by effectively providing sequential information (or temporal semantics): e.g., a continuous transition from the sound of light rain to the sound of heavy rain. There have been introduced sound-guided video generation approaches. However, existing sound-guided video generation approaches are limited to specific applications, such as face generation [15, 26, 21, 14, 21], where audio is used to provide a script for the avatar, or other simple synthetic motions (e.g., a video of musicians playing violin or a video of painting motions by an artist) [7, 13, 18, 4].

Recently, Lee *et al*. [19] introduced a sound-guided landscape video generation model, leveraging the latent space of StyleGAN [16]. They focus on using audio only for semantic labels (i.e., a sound of the wind is simply encoded into a meaning of wind) but not temporal semantics – i.e. semantic information that changes over time. Thus, in this work, we focus on leveraging temporal semantics from audio inputs such that our video generator reactively manipulates video frames. Our model temporally aligns the latent space with the given audio sequence (e.g., continuous changes in audio, e.g., weak rain $\rightarrow$ heavy rain, are reflected to generate corresponding video frames).

Our work starts with Stable Diffusion [29], a text-driven image generator with advantages in generating high-resolution images based on the latent diffusion models. Its architectural advantages (i.e., attention mechanism and diffusion process) help leverage audio as a driving condition, generating temporally reactive and consistent video frames. Given the latent space of trained Stable Diffusion, we generate video frames temporally guided by audio sequences with regularizers to ensure temporal consistency (between generated consecutive frames) and correspondence with audio inputs.

Our model consists of two main modules: (i) *Audio Encoder*, which is designed to encode temporal semantics of audio sequences, producing a sequence of the latent vectors. (ii) *Audio Semantic Guidance Module*, which uses the above-mentioned latent vectors as a condition in the diffusion process to generate corresponding image outputs. We apply identity regularizer to produce temporally consistent video frames, while we apply audio semantic guidance to generate audio-reactive video frames.

Figure 2: Limitation of text-driven image manipulation. Given a generated image by Stable Diffusion [29] with a text prompt, "A photo of a person on the street," additional textual conditions of different semantic meanings (i.e. "a little of rain", "rain", and "rain a lot") produce similar images, failing to capture differences. Note that we apply SEGA [2]'s guidance to preserve content identity.

However, we observe that generating multifarious high-quality images solely from a sound input is challenging due to the lack of such a large-scale dataset to train a model. Instead, we first generate an initial frame using pre-trained Stable Diffusion model with a text prompt, then generate the following video frames conditioned on audio inputs. This frees a model from data dependency burdens and enables training with the current relatively small (than image-text modality) audio dataset [19, 5] , focusing on leveraging temporal semantics from audio. We summarize our contributions as follows:

- We propose a novel sound-driven video generation method built upon Stable Diffusion [29] and can generate video frames reactively with audio sequence inputs.
- Our attention-based Audio Encoder produces temporally-aware latent vectors, which are consumed by Stable Diffusion as a per-time manipulation condition, producing audio-reactive video frames.
- Our model regularizes the latent features of diffusion models to produce temporally consistent video frames, preserving identity throughout the generated video.
- We demonstrate the effectiveness of our proposed model using a public dataset Landscape [19], generally outperforming other state-of-the-art sound-driven video generation approaches in terms of video quality metrics and human evaluation.

2. Related Work

Latent Diffusion Models. Recent success [29] suggests that the Latent Diffusion Models (LDM) improve the efficiency of the diffusion process, successfully generating

Table 1: Comparison between existing state-of-the-art audio-driven video generation approaches in terms of whether they consider the following factors: temporal semantics, magnitude changes of sound, and target domains.

Model	Temporal Semantics	Magnitude	Domains (Audio Type)
Sound2Sight [4]	-	✓	Closed
CCVS [18]	-	✓	Closed (Music)
TräumerAI [13]	-	✓	Closed (Music)
Lee *et al*. [19]	✓	-	Closed (Nature)
Ours	✓	✓	Open Domains

high-quality images given a text prompt. One challenge in LDM is that the generation process is too sensitive to the condition, making it difficult to control semantics. Recently, there have been introduced to control semantics with LDM by a semantic mask [1] or by utilizing semantic information of the cross-attention layers [9]. Wu *et al*. [37] used linear combinations of text embeddings and Liu *et al*. [22] proposed composable diffusion models, but they still remained challenging to control fine-grained semantic changes. Recently, Semantic Guidance (SEGA) [2] computed a guidance vector in the latent space, enabling semantic control of diffusion models without further inputs. Inspired by SEGA, we also control the semantics in the latent space with temporally-encoded audio vector sequences.

Text-driven Video Generation. Recent text-to-video generation tools, including Make-A-Video [32], Video Diffusion Models [11], Imagen video [10], and Phenaki [36] have shown promising performance in generating videos from textual descriptions. However, text-to-video generation has its limitations in terms of temporal coherence, which mostly leads to short video duration or a linear video change. Recent text-to-video generation methods, StyleGAN-V [34] and Dreamix [23], made progress addressing these issues. However, conditioning temporal semantics or complex scenarios is still challenging to be obtained from text inputs. Thus, in this paper, we want to explore conditioning a model with audio inputs, which inherently convey such temporal semantics.

Audio-driven Video Generation. Leveraging temporal semantics was not seriously considered in previous audio-driven video generation approaches. Sound2Sight[4] and CCVS [18] generate video frames conditioned on the (non-temporal) context of the given audio, while TräumerAI [13] utilized the magnitude of the given audio. Recently, Lee *et al*. [19] explored a model that can consider audio semantics as a condition to drive a video generator. Also, their dependency on StyleGAN [34]-based embedding space makes it difficult for models to generate transitions in video. In this work, we focus on leveraging temporal semantics from audio inputs such that our generator reactively manipulates video frames.

3. Method

In this work, we propose a novel audio-driven video generation method, which generates video frames conditioned on audio sequences, ensuring temporal consistency between consecutive frames and temporal correspondence with audio inputs. As shown in Figure 3, our model consists of two main parts: (i) *Audio Encoder*, which encodes temporal semantics of audio sequences, producing a sequence of the latent vectors (Section 3.2). (ii) *Audio Semantic Guidance Module*, which uses the above-mentioned latent vectors as a condition in the diffusion process to generate corresponding image outputs, which are temporally consistent (by our identity regularizer) and audio-reactive (see Section 3.3).

3.1. Preliminary: Latent Stable Diffusion

Latent Diffusion Models (LDMs) [29] are the method that uses an encoder to convert a noised latent vector $\mathbf{z}^T$ to a denoised latent vector $\mathbf{z} = \mathbf{x} + \epsilon$, where $\mathbf{z}$ is a latent vector of an input image $\mathbf{x}$ and ϵ is a noise. Stable Diffusion [29] is part of a conditional generation model that can synthesize an image given a condition $\mathbf{y}$. It uses U-Net [30] as denoising autoencoders represented by ϵ_θ. To generate an output image, the autoencoder $\epsilon_\theta(\mathbf{z}^t, t, \tau_\theta(\mathbf{y}))$ takes three inputs: the noised latent vector $\mathbf{z}^t$, a sequence t that is uniformly sampled from the set $1, \ldots, T$, and a conditional input $\mathbf{y}$.

Conditional input $\mathbf{y}$ is first transformed into a latent vector $\mathbf{c}_p$ through a pretrained function τ_θ and then it is fed into a cross-attention layer of the U-Net as the key and value, which is then combined with $\mathbf{z}^t$ by an attention mechanism, where the query is the flattened intermediate representation of the U-Net. Next, the denoising autoencoder ϵ_θ is used to denoise $\mathbf{z}^t$ from $t = T$ to $t = 1$ sequentially. After T denoising steps, the resulting denoised latent vector $\mathbf{z}$ $(= \mathbf{z}^1)$ is transmitted to the decoder, which produces the final output image $\tilde{\mathbf{x}}$.

3.2. Encoding Temporal Semantics from Audio

We use an audio modality as a source of generating temporal conditions. We divide the audio mel-spectogram into uniform snippets (segments) to extract abundant audio features and feed this into the Audio Encoder to extract both temporal semantic features and intensity of audio in end-to-end manner. The figure of our training process is illustrated in Figure 4.

Audio Feature Extraction. Audio inputs are first transformed into a mel-spectrogram representation, denoted as $\mathbf{x}^a \in \mathbb{R}^{d \times w}$, where d represents the number of mel-frequency bins and w is the width of the spectrogram. To incorporate time information, the mel-spectrogram is divided into N segments. Each segment, denoted as $\mathbf{x}^a_n \in \mathbb{R}^{d \times \lceil \frac{w}{N} \rceil}$, where $n \in \{1, \ldots, N\}$, is then fed into a shared feature extraction module, i.e., the pre-trained ResNet18 [8]. The

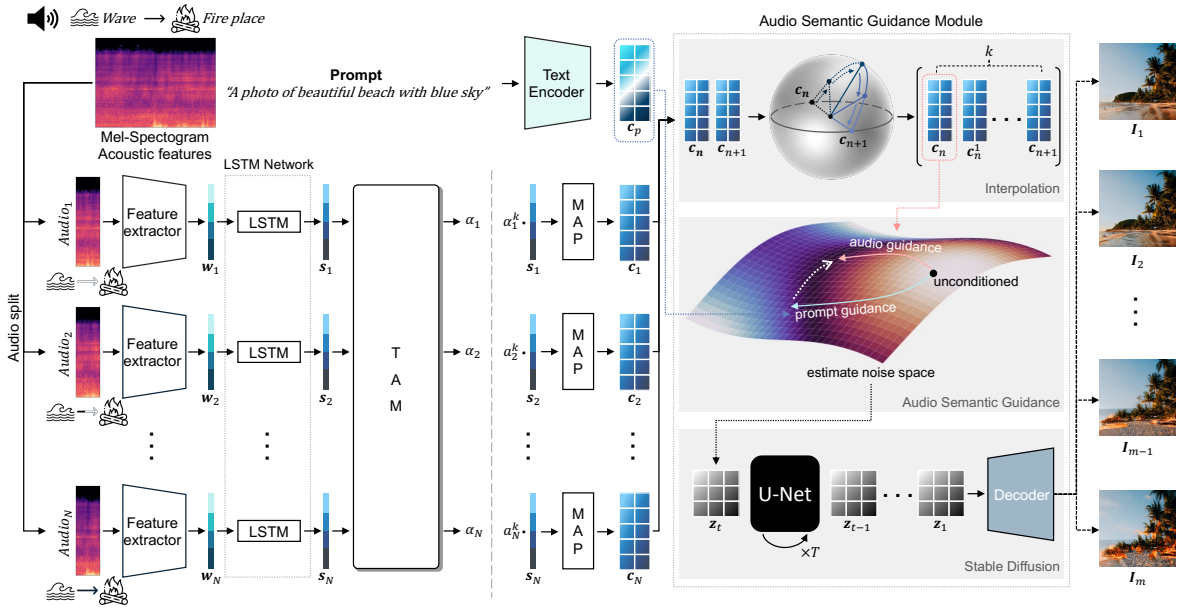

Figure 3: An overview of our proposed TPoS model. Our model consists of two main modules: (i) Audio Encoder, which produces a sequence of latent vectors, encoding temporal semantics of audio input by utilizing CLIP [27] space and highlighting the important temporal features and (ii) Audio Semantic Guidance Module, which is based on the diffusion process, generating video frames that are temporally consistent and audio-reactive.

feature extraction module $f_a(\cdot)$ learns to extract low-level features from each audio segment regardless of its time dependency, i.e., $\mathbf{w}_n = f_a(\mathbf{x}_n^a)$

LSTM-based Temporal Semantic Encoder. As our goal is to generate audio-reactive video frames, it is also important to encode temporal changes (or relations) of the given audio inputs. Similar to Lee *et al.* [19], given audio features $\mathbf{w} \in \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$, which encodes per-segment disjoint audio representation, we apply the standard Long Short-Term Memory (LSTM) network [12] to encode temporal relations or changes between consecutive audio features $\mathbf{w}$. Formally, our LSTM takes the audio feature $\mathbf{w}_{n-1}$ as input and updates its hidden state $\mathbf{h}_n$, producing an output $\mathbf{s}_t$: i.e. $(\mathbf{s}_n, \mathbf{h}_n) = \texttt{LSTM}(\mathbf{h}_{n-1}, \mathbf{w}_{n-1})$. These outputs $\mathbf{s} \in \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$ are then fed into Temporal Attention Module (TAM) to further encode temporal semantics.

Aligning Audio Semantics with Image-Text CLIP Joint Space. As we will use the output $\mathbf{s}$ as a condition to manipulate video frames, it is important to ensure those audio features are well-aligned with other text and visual features in the CLIP [27]-based joint embedding space. Similar to Lee *et al.* [20], given the pre-trained image-text CLIP space, we apply the following loss $\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow t}$ with the InfoNCE loss [24] l_{sim} such that positive pairs (e.g. an audio of raining and a text prompt "raining") are pulled close to each other, while negative pairs are pushed farther away.

$$\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow t} = l_{sim}(\mathbf{s}_N, \texttt{CLIP}_t(\mathbf{t})) + l_{sim}(\texttt{CLIP}_t(\mathbf{t}), \mathbf{s}_N) \quad (1)$$

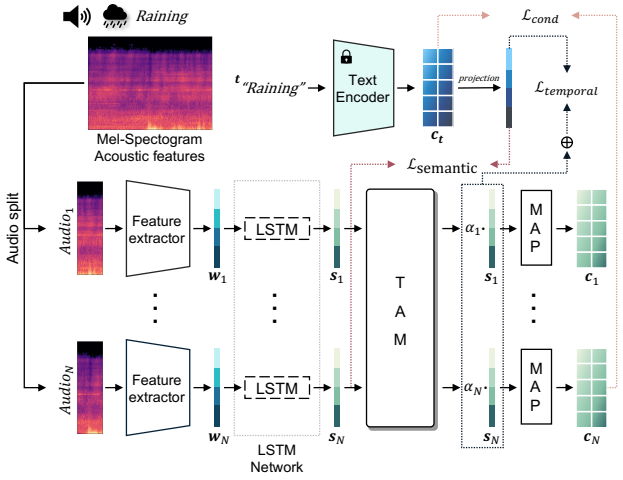

Figure 4: An overview of our Audio Encoder training process. Our model generates temporally-encoded audio embeddings with an LSTM [12] layer and Temporal Attention Module (TAM). Audio input is partitioned into N segments, and each of these is encoded and used as a condition to manipulate audio-reactive video sequences (e.g. light rain $\rightarrow$ heavy rain). This is done by our Mapping Module (MAP), which maps the audio embedding to the latent space of Stable Diffusion.

where $\texttt{CLIP}_t$ is a pre-trained CLIP-based text encoder, which takes a text prompt $\mathbf{t}$ obtained from audio class labels as input, yielding an d-dimensional feature. Note that

we only apply this loss for the final output $\mathbf{s}_N$ for efficient training. Given a set of positive pairs, we apply the following InfoNCE loss $l_{sim}(\mathbf{a}, \mathbf{b})$:

$$l_{sim} = -\log \frac{\exp(\langle \mathbf{a}_i, \mathbf{b}_i \rangle)/\tau}{\sum_j \exp(\langle \mathbf{a}_i, \mathbf{b}_j \rangle)/\tau} \tag{2}$$

where $\langle \mathbf{a}_i, \mathbf{b}_i \rangle$ represents the cosine similarity with temperature τ. Note that we set τ to 0.07.

Augmenting Audio Semantics. Audio data is often limited in volume and diversity; thus, augmentation techniques may be required to extract better-quality audio semantic features, preventing a representation collapse. We use SpecAugment [25] to apply random transformations (such as masking our certain frequency bands or time segments), yielding augmented audio inputs. We further add the InfoNCE loss [24] $\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow a'}$ to pull augmented audio features together.

$$\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow a'} = l_{sim}(\mathbf{s}_N, \mathbf{s}'_N) + l_{sim}(\mathbf{s}'_N, \mathbf{s}_N) \tag{3}$$

where apostrophe indicates augmented view of an original audio data. Finally, we use the semantic loss $\mathcal{L}_{\text{semantic}}$: i.e. $\mathcal{L}_{\text{semantic}} = \mathcal{L}_{\text{CLIP}}^{a \leftrightarrow t} + \lambda_s \mathcal{L}_{\text{CLIP}}^{a \leftrightarrow a'}$ where λ_s is set to 0.6.

Temporal Attention Module (TAM). As shown in Figure 5, we further use an attention-based module to encode temporal semantics from the audio inputs. We empirically observe that adding this module helps improve the quality of video frame generation, which is probably due to the fact that the model becomes flexible to focus on more important temporal information, improving the model's representation power. Formally, we first compute attention weight α_n for a given audio feature $\mathbf{s}_n$ by applying an MLP layer f_{proj} followed by a softmax operation: i.e. $\alpha_n = \exp(f_{\text{proj}}(\mathbf{s}_n))/\sum_n \exp(f_{\text{proj}}(\mathbf{s}_n))$ such that $\sum_n \alpha_n = 1$. We compute the weighted sum of audio features based on attention weights, yielding an attended audio feature $\mathbf{o}^a = \sum_n \alpha_n \mathbf{s}_n$. We add another InfoNCE loss $\mathcal{L}_{\text{temporal}}$ to align the audio features with text:

$$\mathcal{L}_{\text{temporal}} = l_{sim}(\mathbf{o}^a, \texttt{CLIP}_t(\mathbf{t})) + l_{sim}(\texttt{CLIP}_t(\mathbf{t})), \mathbf{o}^a) \tag{4}$$

Lastly, we also minimize the MSE loss between text embeddings (before the projection layer) and the projected audio feature $\texttt{MAP}(\mathbf{o}^a)$: $\mathcal{L}_{\text{cond}} = ||\mathbf{c}_t - \texttt{MAP}(\mathbf{o}^a)||_2^2$. Note that we exclude the feature for the <SOS> token. More details are provided in the supplemental material.

Total Loss. We train our Audio Encoder end-to-end by minimizing the following loss function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{semantic}} + \mathcal{L}_{\text{temporal}} + \mathcal{L}_{\text{cond}} \tag{5}$$

3.3. Generating Video Frames with Stable Diffusion

Initial Frame Generation from Text Prompt. Our model relies on leveraging combinational operations of denoising process. The estimated noise space can be either random or manual by visual input with the help of the diffusion encoder. Based on this guided diffusion, our model first generates the initial frame with a text prompt (e.g. "a photo of beautiful beach with blue sky"). Given this generated image as *content*, we manipulate its *styles* according to audio inputs and generate corresponding video frames, i.e. given a series of latent vectors $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$, we generate m video frames. Note that the number of video frames is controllable by latent vector interpolation, as we will explain later. We follow the standard image generation process with the Stable Diffusion model [29], i.e., we compute a latent vector $\mathbf{c}_p$ in a CLIP-based embedding space given a text prompt, conditioning it to generate an image.

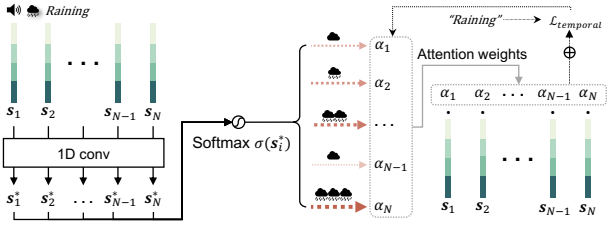

Figure 5: Details of our Temporal Attention Module (TAM). Temporal Attention Module effectively captures the important audio segments (e.g. harsh raining sound) as attention weights, which are guided by CLIP [27]'s text embedding (e.g. "raining") in training phase and express the magnitude of audio in test phase.

Audio Semantic Guidance. We employed the SEGA [2] framework to create video frames that incorporate sound style while preserving the main content identity. We first utilize the combination of attention α_n and audio feature $\mathbf{s}_n$ with normalization of the scale of output feature by multiplying N to produce output $\mathbf{c}_n$: i.e., $\mathbf{c}_n = N^k \alpha_n^k \mathbf{s}_n$, where k is hyper parameter that regulates attention ratio and is set to 1. The Audio Semantic Guidance module generates the n-th video frame by taking audio condition $\mathbf{c}_n$ as input to guide the diffusion models. The denoising autoencoder ϵ_θ is executed $\delta - 1$ times out of T time to form incomplete noise along with the original text prompt meaning. From $t = \delta$, the audio semantic guidance operates through the following equation:

$$\begin{aligned} \tilde{\epsilon_\theta}(\mathbf{z}^\delta, \mathbf{c}_p) :=& \epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_\varnothing) + g(\epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_p) - \epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_\varnothing)) \\ &+ \lambda(\mathbf{z}^\delta, \mathbf{c}_n) \end{aligned} \tag{6}$$

where $\mathbf{z}^\delta$ is denoised latent vector at $t = \delta$, g is the guidance scale of the text prompt, $\lambda(\mathbf{z}^\delta, \mathbf{c}_n)$ is the audio semantic guidance term, and $\mathbf{c}_\varnothing$ represents an unconditioned prompt that does not make any semantic difference. As a result, only the $\lambda(\mathbf{z}^\delta, \mathbf{c}_n)$ term has been added to the original denoising process from $t = \delta$ to $t = 1$. Note that $\mathbf{z}^T$ is fixed through frames in one video. The audio semantic guidance

Table 2: Comparison of the quality of generated video frames with state-of-the-art audio-to-video generations in terms of IS [31], FVD [35], and CLIP [27]-based distances.

Model	Input	IS↑	FVD↓	CLIP↑ $(a \leftrightarrow v)$	CLIP↑ $(t \leftrightarrow v)$
Sound2Sight [4]	1st Frame	1.02 ± 0.02	494.28	0.0364	0.2164
CCVS [18]	1st Frame	1.30 ± 0.20	679.94	0.1251	0.2360
TräumerAI [13]	-	1.47 ± 0.19	736.32	0.1589	0.1778
Sound-guided Video Generation [19]	-	1.16 ± 0.16	544.09	0. 1151	0.1702
Ours (w/o TAM)	Latent Vector	1.27 ± 0.23	483.76	0.1342	0.2370
Ours	Latent Vector	$\mathbf{1.49 \pm 0.38}$	**421.23**	**0.1964**	**0.2436**

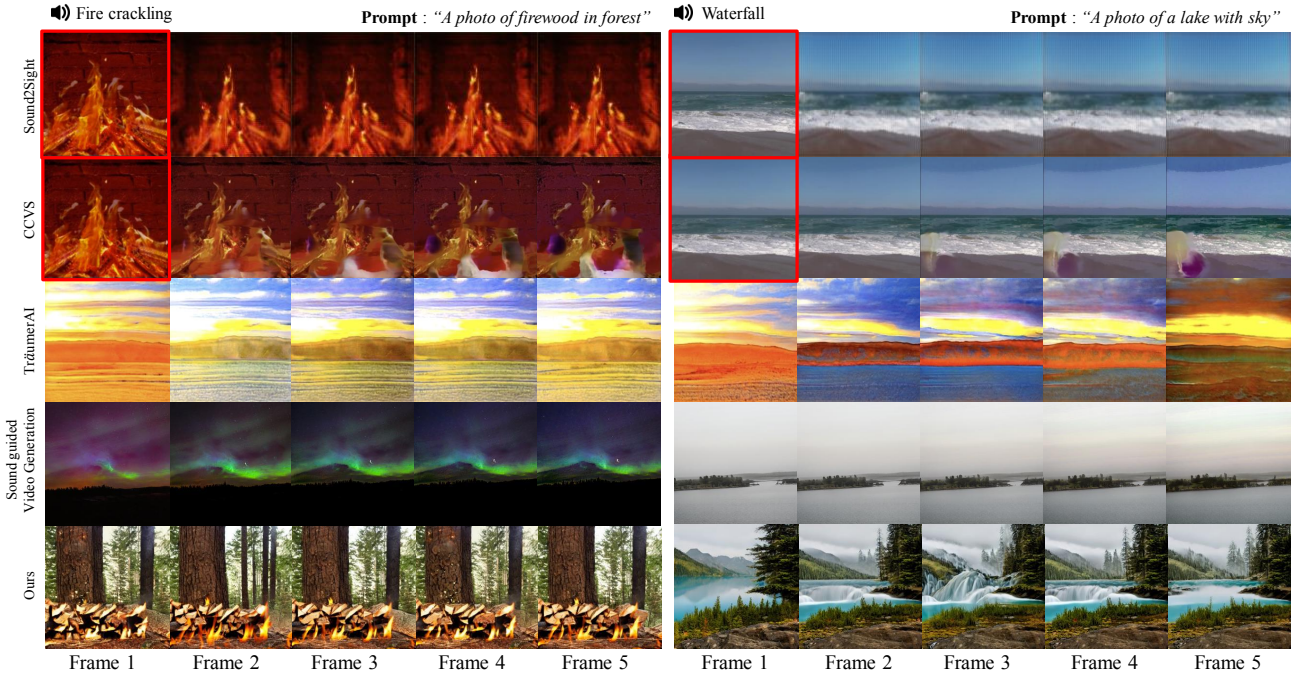

Figure 6: Examples of generated video frames (given fire crackling and waterfall audio) by Sound2Sight [4], CCVS [18], TräumerAI [13], Lee *et al*. [19], and ours. Note that Sound2Sight and CCVS use an initial frame (highlighted in a red box).

$\lambda(\mathbf{z}^{\delta}, \mathbf{c}_n)$ is defined as follows:

$$\lambda(\mathbf{z}^{\delta}, \mathbf{c}_n) = g_s(\epsilon_\theta(\mathbf{z}^{\delta}, \mathbf{c}_n) - \epsilon_\theta(\mathbf{z}^{\delta}_{\varnothing}, \mathbf{c}_{\varnothing})) + \sigma_m \Phi_m \quad (7)$$

where $\sigma_m \in [0, 1]$ is the momentum hyper parameter that scales the momentum Φ_m and g_s focuses on the relevant dimensions to the audio manipulation task (More details can be found in the supplemental material or SEGA [2]). Different from SEGA framework, we only consider positive guidance of audio semantics, which is the semantic difference between the guidance provided by $\mathbf{c}_n$ and the unconditioned guidance $\mathbf{c}_{\varnothing}$.

The generation process conditioned by $\mathbf{c}_n$ is separately working through diffusion processes so that the different semantic meaning or magnitude which presents in $\mathbf{c}_n$ can generate frames independently.

Temporal Frame Interpolation. We use an interpolated latent vector to generate continuous video frames between two consecutive frames. Following the work by Ramesh *et al*. [28], we apply a spherical linear interpolation between all consecutive pairs of $\mathbf{c}_n$ and $\mathbf{c}_{n+1}$, yielding k interpolated latent vectors. These vectors are then used as a condition for the diffusion models to generate temporally-interpolated video frames.

4. Experiments

Datasets. We use two Audio-Video datasets to train our Audio Encoder: VGG-Sound [5] and Landscape [19]. VGG-Sound is an audio dataset with about 170,000 of 10-second clips of audio-video data, which consists of 309 classes. The dataset has numerous 'in the wild' audio data that spans

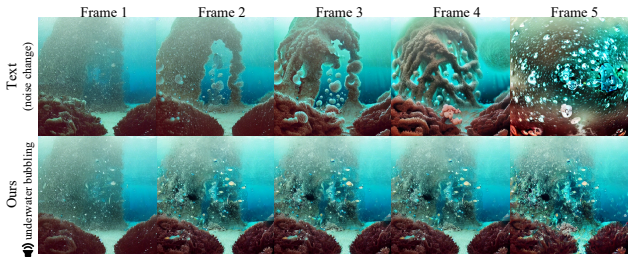

Figure 7: Examples of our video generation conditioned on text prompt (top) and audio (bottom).

a large number of challenging acoustic environments and real application noise characteristics. Since audio of nature sound is the perfect tool to stylize compared to the class such as people talking or sports, we add about 9,000 audio clips of Landscape audio dataset in the training process. For obtaining test sets, class-balanced sampling is applied to Landscape dataset.

Baselines. We compare our methods with existing audio-to-video generation methods, Sound2Sight [4], CCVS [18], and StyleGAN [34] based TräumerAI [13] and Sound-guided Video Generation [19]. We follow experiment methods by Lee *et al*. [19]. All baselines are trained or finetuned with Landscape dataset [19]. Since Sound2Sight and CCVS need first frame to generate the video, we randomly select first frame from the Landscape dataset at the inference task. For TräumerAI and Sound-guided Video Generation, we first pre-train StyleGAN [16] on high fidelity benchmark datasets (LHQ datasets [33]) and then train the model to navigate the latent space by Landscape dataset. Note that randomly initialized vector for TräumerAI [13] and Sound-guided Video Generation [19] are given. For our model, we initially generate image from random noise space and randomly sample the prompt related with landscape to generate the landscape like video for fair comparison.

Evaluation Metrics. We use the following two video quality metrics for our evaluations: (i) Fréchet Video Distance (FVD) [35] and (ii) Inception Score (IS) [31]. FVD is used to assess video quality by measuring the distribution gap between real and synthesized videos in the latent space. Additionally, IS score is commonly employed to evaluate the effectiveness of Generative Adversarial Networks (GANs) by computing the KL-divergence between the label distribution of each image and the marginal label distribution. To implement these, we fine-tune Inflated 3D ConvNet [3] with Landscape [19] dataset for FVD and used pre-trained InceptionNet [17] that is trained on ImageNet [6] dataset. We also measure CLIP [27]-based cosine similarity (CLIP score) between audio and image as well as text and image. To obtain the textual pivot feature, we fed the following prompt "The photo of `<class>`" into CLIP text encoder.

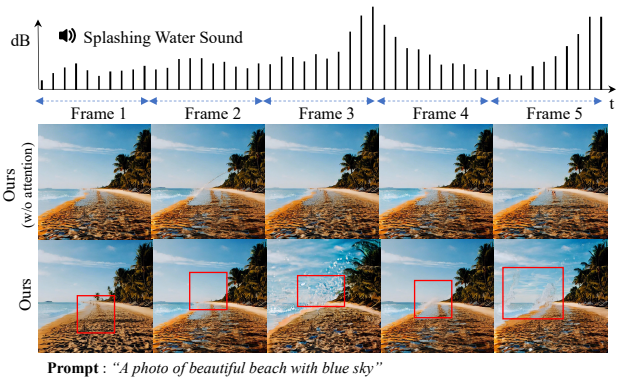

Figure 8: Generated video frames *with* and *without* our Temporal Attention Module. We generate video frames with splashing water sound where its amplitude changes over time (see top).

Quantitative Experiments. For fair comparison with other existing baselines such as TräumerAI [13] and Sound-guided Video Generation [19] which does not get a hint about what to generate, we use prompt that originally does not generate the style of sound. We set fps 20 and generated videos to extract images from all baselines. Table 2 shows that our approach produces the best quality results as video. Additionally, to ensure that the generated videos are semantically related to the sound, we compare the cosine similarity between text-audio and video embedding. Our methods shows a superior performance in terms of multi-modal semantics.

Qualitative Video Quality Comparison. We first evaluate the quality of generated video frames. In Figure 6, we provide typical examples of generated video frames by (from top) Sound2Sight [4], CCVS [18], TräumerAI [13], Lee *et al*. [19], and ours. Note that Sound2Sight [4] and CCVS [18] need the initial frame as input (see red boxes). With audio inputs, such as fire crackling and waterfall, as a condition, ours generally generate temporally-consistent audio-reactive video frames. We observe that Sound2Sight [4] and CCVS [18] often show blurring artifacts, while StyleGAN-based TräumerAI [13] and Sound-guided Video Generation [19] often fail to generate audio-reactive video frames, e.g., they produce landscape scenes that are not semantically aligned with a fire crackling audio. However, ours generate a scene of a waterfall or a fire on firewood, aligning well with audio inputs.

Comparison to Video Generation with Text. To analyze the benefit of audio modality, we conduct a qualitative experiment to compare the effect of audio and text modalities in generating visual content. As shown in Figure 7, we first generate an initial frame with the text prompt "A photo of deep in the sea." Then we generate the next frames with text "underwater bubbling" (top) and underwater bubbling

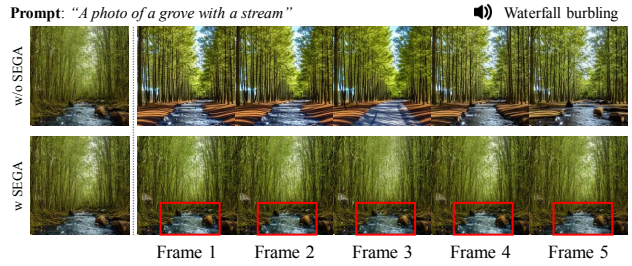

Figure 9: Generated video frames with and without our Audio Semantic Guidance. Edited parts by Audio Semantic Guidance are highlighted with red box (see second row).

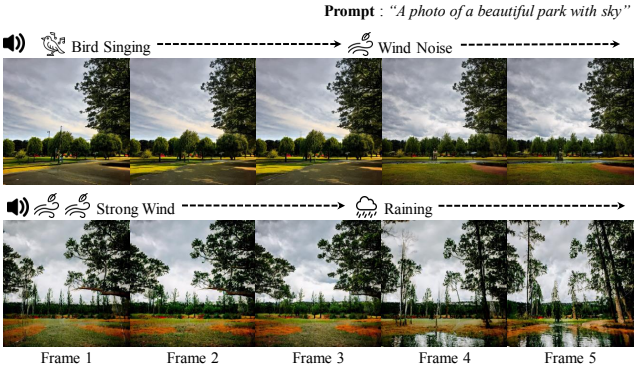

Figure 10: Examples of generated video frames with a sound that changes over time (e.g. bird singing → wind noise).

sound (bottom). It is difficult to make temporal changes conditioned on text unless we train our model with a text-video dataset. Thus, we instead change the noise scale to make temporal changes, preserving identity. However, as shown in Figure 7 (top), it generates distortions or a linear change, but this is not the case for audio. Our model with audio generates visually-appealing video frames.

Effect of Temporal Attention Module. We use Temporal Attention Module (TAM) to improve the representation power to encode temporal semantics better. To analyze this, we perform an ablation study with and without TAM to see its effect on video frame generation. We observe in Figure 8 that our model is indeed able to generate video frames reactive to the audio changes over time (compare the changes along with the given splashing water sound).

Effect of Audio Semantic Guidance. We demonstrate the effect of Audio Semantic Guidance with ablation study. First, we generate video frames without Audio Semantic Guidance by replacing audio semantic guidance term $\lambda(\mathbf{z}^{\delta}, \mathbf{c}_n)$ with $\epsilon_\theta(\mathbf{z}^{\delta}, \mathbf{c}_n)$ (refer the notation in Section 3.3). To remove the effect of Audio Semantic Guidance, we also set δ to T. As illustrated in Figure 9 (top), without Audio Semantic Guidance, the model has a tendency to produce unnecessary changes (e.g. grass → toil) and struggles to

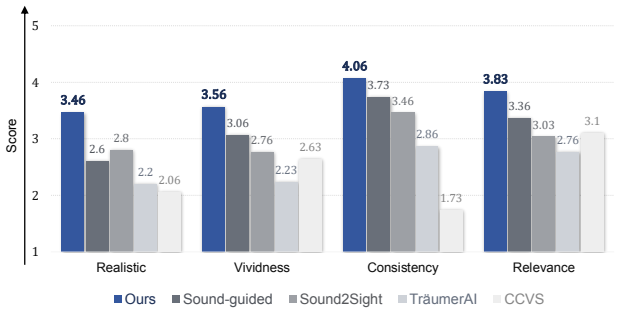

Figure 11: Our human evaluation results. We conduct a user study with 100 participants on Amazon Mechanical Turk (AMT). Participants are shown generated video frames and asked to evaluate them in terms of realistic, vividness, consistency, and relevance. The Likert scale is used (higher is better).

achieve consistent alignment with audio (e.g. producing a road with a waterfall burbling sound). On the other hand, by leveraging Audio Semantic Guidance as in Figure 9 (bottom), we can generate sequential video frames that not only has consistent *content* but also represents natural temporal variations according to audio sound (e.g. producing wave of water with waterfall burbling sound), resulting in enhanced naturalness.

Experiments of Semantic Transition. In Figure 10, we provide examples where audio inputs are changed (e.g., bird singing → wind noise, strong wind → raining). As we observe in that figure, our model successfully adapts to the audio change, generating video frames accordingly. This may confirm that our model is indeed conditioned on the audio sequence and can generate audio-adaptive video frames.

User study. Further, we conduct a human evaluation to evaluate the video quality by human judges. We recruit 100 participants from Amazon MTurk. Participants are shown video frames generated by five different audio-driven video generation models: Sound2Sight [4], CCVS [18], TräumerAI [13], Lee *et al*. [19], and ours. Participants are asked to evaluate the given video frames in terms of realism, vividness, consistency, and relevance. We use a five-point Likert scale where ideal video frames will get all five points. More details are provided as supplemental material. We observe in Figure 11 that our proposed method outperforms the other approaches in all categories. These results are consistent with our quantitative and qualitative results.

Visual Conditioning. Our model is capable of producing an initial frame using visual input (such as images) with the trained diffusion encoder. Figure 12 demonstrates that our framework leverages latent space of a diffusion model with a combination of visual input, text prompt and audio. We first randomly choose visual input (e.g., sun light photo) and

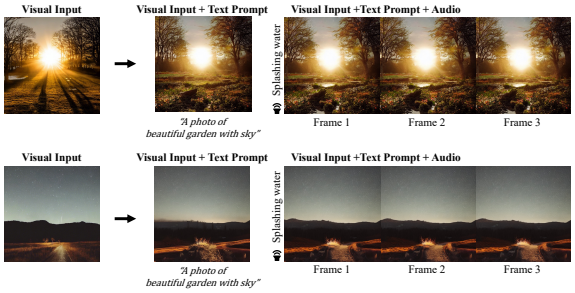

Figure 12: Example of generated video with visual input, text prompt and audio sound. (e.g., 1st row: sun light photo conditioned by text "A photo of beautiful garden with sky" and splashing water sound.)

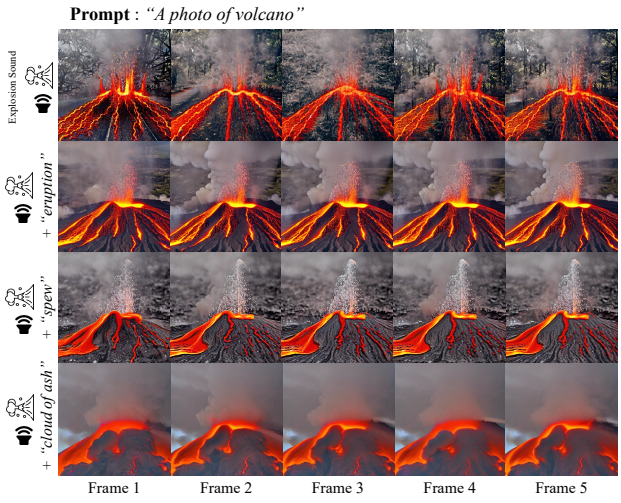

Figure 13: Example of generated videos with audio-text joint condition (e.g., 2nd row: conditioned with text "eruption" and explosion sound)

generate an initial frame with the text prompt (e.g., "A photo of beautiful garden with sky"). Then we incorporate an audio sound (e.g, splashing water) to generate videos. Shown in Figure 12, it can produce videos featuring a garden with water where the sun is centered (1st row) or where a black mountain in the background (2nd row).

Text-Audio Joint Conditioning. As our model is built upon the Stable Diffusion model, it is also possible to use text and audio as a condition together. In Figure 13, we provide an example where we generate video frames conditioned on a sound of an explosion along with texts, such as "eruption", "spew", or "cloud of ash." (see 2nd-4th rows) Preserving temporal semantics, our model successfully generates video frames guided by text as well.

Face Generation. Our Audio Semantic Guidance enables detailed adjustment in certain areas, such as face generation. Figure 14 shows examples of human faces conditioned on

Figure 14: Examples of face generation with our methods. We use giggling and sobbing sound to manipulate face expressions of human face.

giggling sound (1st and 3rd rows) and sobbing sound (2nd row). Our model excels in preserving the core facial attributes while manipulating emotional expressions.

5. Conclusion

In this paper, we propose The Power of Sound (TPoS), a novel audio-to-video generation with Stable Diffusion. Our work extends the usage of audio modality on generation models, and broaden the methods of using Stable Diffusion by generating realistic videos by our Audio Encoder. Superior performances are achieved over widely-used audio-to-video benchmarks, reflected by objective evaluations and User study, hence attributing towards the new formulation of video generation with audio modality.

Acknowledgements. This work was supported by the National Research Foundation of Korea grant (NRF-2021R1C1C1009608, 10%), Basic Science Research Program (NRF-2021R1A6A1A13044830, 5%), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(2022-0-00043, 30%), and the MSIT(Ministry of Science and ICT), Korea, under the ICT Creative Consilience program(IITP-2023-2020-0-01819, 5%) supervised by the IITP. W. Ryoo, S. Lee, and S. Kim are supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program(Korea University), 30%), the National Research Foundation of Korea grant (NRF-2022R1F1A1074334, 15%), Culture, Sports, and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2023 (Project Name: 4D Content Generation and Copyright Protection with Artificial Intelligence, Project Number: R2022020068, 5%), and supported by the Google Cloud Research Credits program(code- MU6X-FKAV-N3A3-1X0T).

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. 3

[2] Manuel Brack, Patrick Schramowski, Felix Friedrich, Dominik Hintersdorf, and Kristian Kersting. The stable artist: Steering semantics in diffusion latent space. *arXiv preprint arXiv:2212.06013*, 2022. 2, 3, 5, 6

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 7

[4] Moitreya Chatterjee and Anoop Cherian. Sound2sight: Generating visual dynamics from sound and context. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 701–719. Springer, 2020. 2, 3, 6, 7, 8

[5] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. 2, 6

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[7] Sarah Gross, Xingxing Wei, and Jun Zhu. Automatic realistic music video generation from segments of youtube videos. *arXiv preprint arXiv:1905.12245*, 2019. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[9] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3

[10] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2, 3

[11] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 2, 3

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[13] Dasaem Jeong, Seungheon Doh, and Taegyun Kwon. Träumerai: Dreaming music with stylegan. *arXiv preprint arXiv:2102.04680*, 2(4):10, 2021. 2, 3, 6, 7, 8

[14] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Qianyi Wu, Wayne Wu, Feng Xu, and Xun Cao. Eamm: One-shot emotional talking face via audio-based emotion-aware motion model. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2

[15] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Wayne Wu, Chen Change Loy, Xun Cao, and Feng Xu. Audio-driven emotional video portraits. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14080–14089, 2021. 2

[16] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2, 7

[17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 7

[18] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Ccvs: context-aware controllable video synthesis. *Advances in Neural Information Processing Systems*, 34:14042–14055, 2021. 2, 3, 6, 7, 8

[19] 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 *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 34–50. Springer, 2022. 2, 3, 4, 6, 7, 8

[20] Seung Hyun Lee, Wonseok Roh, Wonmin Byeon, Sang Ho Yoon, Chanyoung Kim, Jinkyu Kim, and Sangpil Kim. Sound-guided semantic image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3377–3386, 2022. 4

[21] Borong Liang, Yan Pan, Zhizhi Guo, Hang Zhou, Zhibin Hong, Xiaoguang Han, Junyu Han, Jingtuo Liu, Errui Ding, and Jingdong Wang. Expressive talking head generation with granular audio-visual control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3387–3396, 2022. 2

[22] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 423–439. Springer, 2022. 3

[23] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023. 2, 3

[24] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4, 5

[25] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019. 5

[26] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 484–492, 2020. 2

[27] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4, 5, 6, 7

[28] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 6

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 3, 5

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6, 7

[32] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2, 3

[33] Ivan Skorokhodov, Grigorii Sotnikov, and Mohamed Elhoseiny. Aligning latent and image spaces to connect the unconnectable. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14144–14153, 2021. 7

[34] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022. 3, 7

[35] 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. 6, 7

[36] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 2, 3

[37] Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. *arXiv preprint arXiv:2212.08698*, 2022. 3