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 *]

Abstract

In recent years, video generation 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 audio-to-video generation techniques.

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.

1 Introduction

Recent advancements in generative models have demonstrated their ability to create visually appealing video frames using a simple text prompt (e.g., "a video of a person on the street on a rainy day") as input (Singer et al., 2022; Ho et al., 2022b). However, generating complex sequential procedures through text can be challenging, i.e., "a video of a person on the street on a rainy day, but a rain suddenly stops, and a wind blows." To address this issue, we leverage sounds for the video generation models, i.e., sound-driven video generation. Audio complements text by providing sequential information or temporal semantics, enabling continuous transitions, such as the sound of light rain to the sound of heavy rain. However, existing sound-guided video generation approaches are limited to specific applications, such as face generation (Prajwal et al., 2020) and music videos (Le Moing et al., 2021; Chatterjee & Cherian, 2020) (e.g., a video of musicians playing violin).

Recently, Lee (Lee et al., 2022a) introduced a sound-guided landscape video generation model, leveraging the latent space of StyleGAN (Karras et al., 2021). 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 work starts with Stable Diffusion (Rombach et al., 2022), 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.

[1]Department of Computer Science and Engineering, Korea University [2]Department of Artificial Intelligence, Korea University [3]NVIDIA Research, NVIDIA Corporation. [*]Correspondence to: Sangpil Kim <spk7@korea.ac.kr>, Jinkyu Kim <jinkyukim@korea.ac.kr>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

Our model consists of two main modules: (i) *Audio Encoder*, which is attention based and 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. 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. We summarize our contributions as follows:

- We propose a novel sound-driven video generation method built upon Stable Diffusion (Rombach et al., 2022) 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 (Lee et al., 2022a), generally outperforming other state-of-the-art sound-driven video generation approaches in terms of video quality metrics and human evaluation.

2 Method

As shown in Figure 2, 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 2.1). (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 (Section 2.2). The preliminary of Latent Stable Diffusion is illustrated in Appendix Section B.1.

2.1 Encoding Temporal Semantics from Audio

We use an audio modality as a source of generating temporal conditions. By allowing to intercorporate time information into the 2D stable diffusion model, we are able to generate dynamic sequences such as videos. See Figure 4 in the appendix for a visual illustration of our training process.

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 $\mathbf{x}_n^a \in \mathbb{R}^{d \times \lceil \frac{w}{N} \rceil}$, where $n \in \{1, \ldots, N\}$, is then fed into a shared feature extraction module, i.e., the pre-trained ResNet18 (He et al., 2016). The 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. Similar to Lee (Lee et al., 2022a), 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 (Hochreiter & Schmidhuber, 1997) 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, producing an output $\mathbf{s}_t$: i.e. $(\mathbf{s}_n, \mathbf{h}_n) = \texttt{LSTM}(\mathbf{h}_{n-1}, \mathbf{w}_{n-1})$.

Aligning Audio Semantics with Image-Text CLIP Joint Space. As we will use the output $\mathbf{s} \in \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$ 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 (Radford et al., 2021)-based joint embedding space. Similar to Lee (Lee et al., 2022b), given the pre-trained image-text CLIP space, we apply the following loss $\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow t}$ with the InfoNCE loss (Oord et al., 2018) 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.

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

where $\texttt{CLIP}_t$ is a pre-trained CLIP-based text encoder, which takes a text prompt $\mathbf{t}$ as an 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 $l_{sim}(\mathbf{a}, \mathbf{b})$:

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

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

We add audio augmentation loss for better quality audio semantic features which denotes $\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow a'}$. (See Appendix Section B.2) 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). We further use an attention-based module to encode temporal semantics from the audio inputs. 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 $\mathbf{o}^a = \sum_n \alpha_n s_n$. Note that we normalize the scale of output feature $\mathbf{o}^a$ by multiplying N in inference stage. We add another InfoNCE loss $\mathcal{L}_{\text{temporal}}$ to align the audio features with text:

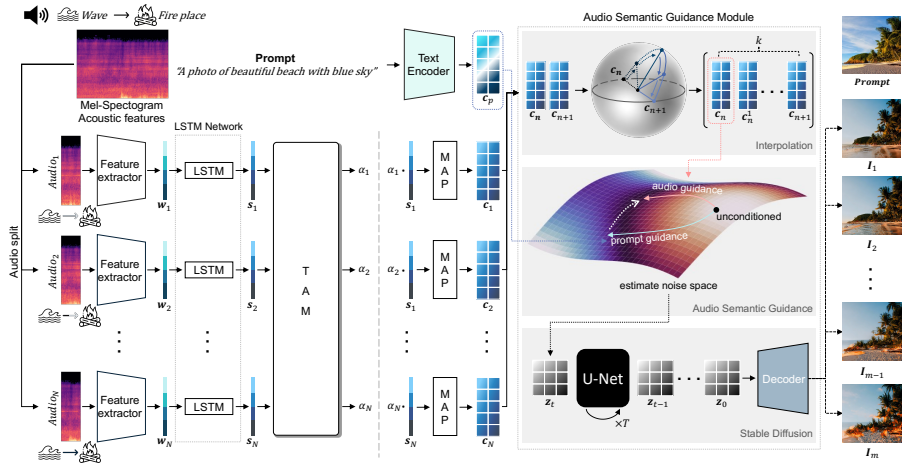

Figure 2: An overview of our proposed TPoS model. Our model consists of two main modules: (i) Audio Encoder, which produces a sequence of latent vectors, encoding temporal semantics of audio input by utilizing CLIP (Radford et al., 2021) space and highlighting the important temporal features and (ii) Audio Semantic Guidance Module, which is based on the diffusion process, generating video frames that are temporally consistent and audio-reactive.

$$\mathcal{L}_{\text{temporal}} = l_{sim}(\mathbf{o}^a, \texttt{CLIP}_t(\mathbf{t}))) + l_{sim}(\texttt{CLIP}_t(\mathbf{t})), \mathbf{o}^a) \quad (3)$$

Lastly, we also minimize the MSE loss between text embeddings (before the projection layer) and the projected audio feature $\texttt{MAP}(\mathbf{o}^a)$: $\mathcal{L}_{\text{cond}} = ||\mathbf{c}_t - \texttt{MAP}(\mathbf{o}^a)||_2^2$. More details are provided in Appendix Section B.2.

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

$$\mathcal{L} = \mathcal{L}_{\text{semantic}} + \mathcal{L}_{\text{temporal}} + \mathcal{L}_{\text{cond}} \quad (4)$$

2.2 Generating Video Frames with Stable Diffusion

Initial Frame Generation from Text Prompt. Our model first generates the initial frame with a text prompt (e.g. "a photo of beautiful beach with blue sky"). We follow the standard image generation process with the Stable Diffusion model (Rombach et al., 2022), i.e. we compute a latent vector $\mathbf{c}_p$ in a CLIP-based embedding space given a text prompt. 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 explain in Appendix Section B.3.

Audio Semantic Guidance. We use SEGA (Brack et al., 2022) to generate sound-styled video frames while preserving content identity. Attention weight $\mathbf{a}_n$ and audio feature $\mathbf{s}_n$ are combined with N as normalization and a hyperparameter k to produce output $\mathbf{c}_n$: $\mathbf{c}_n = N^k \mathbf{a}_n^k \mathbf{s}_n$. Note that we set $k = 1$ in our paper. The Audio Semantic Guidance module use $\mathbf{c}_n$ to generate the nth video frame, guiding diffusion models $\epsilon_\theta(z^t, t, \mathbf{c}_p)$ by additionally adding the audio semantic guidance term $\lambda(\mathbf{z}^\delta, \mathbf{c}_n)$ during the denoising process from $t = \delta$ to $t = 1$, where δ is a hyperparameter. Details of audio semantic guidance are in Appendix Section B.3.

Table 1: Comparison of the quality of generated video frames with state-of-the-art audio-to-video generations in terms of IS (Salimans et al., 2016), FVD (Unterthiner et al., 2018), and CLIP (Radford et al., 2021)-based distances.

Model	IS↑	FVD↓	CLIP↑ $(a \leftrightarrow v)$	CLIP↑ $(t \leftrightarrow v)$
Sound2Sight (Chatterjee & Cherian, 2020)	1.02 ± 0.02	494.28	0.0364	0.2164
CCVS (Le Moing et al., 2021)	1.30 ± 0.20	679.94	0.1251	0.2360
TräumerAI (Jeong et al., 2021)	1.47 ± 0.19	736.32	0.1589	0.1778
SVG (Lee et al., 2022a)	1.16 ± 0.16	544.09	0. 1151	0.1702
Ours	$\mathbf{1.49 \pm 0.38}$	**421.23**	**0.1964**	**0.2436**

3 Experiments

Baselines. We compare our methods with existing audio-to-video generation methods, Sound2Sight (Chatterjee & Cherian, 2020), CCVS (Le Moing et al., 2021), TräumerAI (Jeong et al., 2021) and Sound-guided Video Generation (Lee et al., 2022a). All baselines are trained or fine-tuned on the Landscape dataset (Lee et al., 2022a). Details of experiments are in Appendix Section D.

Quantitative Experiments. Table 1 shows that our approach produces the best quality results as video. Two video quality metrics are used for evaluations, Fréchet Video Distance (FVD) (Unterthiner et al., 2018) and Inception Score (IS) (Salimans et al., 2016). Additionally, to ensure that the generated videos are semantically related to the sound, we compare the cosine similarity between text-audio and video embedding with CLIP (Radford et al., 2021). Our methods shows a superior performance in terms of multi-modal semantics.

Qualitative Video Quality Comparison. In Figure 11, we show examples of generated video frames by baselines and our method. We observe blurring artifacts in Sound2Sight (Chatterjee & Cherian, 2020) and CCVS (Le Moing et al., 2021). Furthermore, StyleGAN-

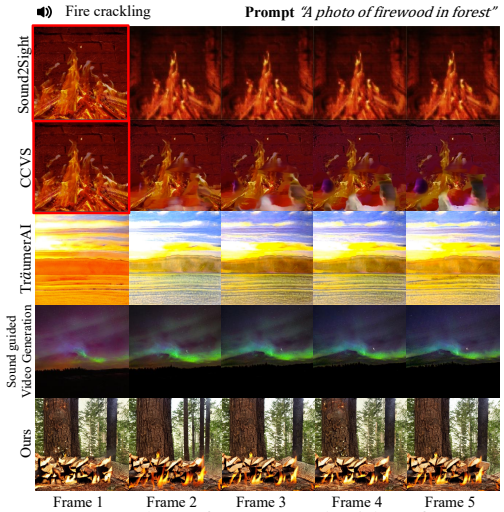

Figure 3: Examples of generated video frames given fire crackling audio by baselines and ours. Note that Sound2Sight and CCVS use an initial frame (red box).

based TräumerAI (Jeong et al., 2021) and Sound-guided Video Generation (Lee et al., 2022a) often fail to generate semantically-aligned audio-reactive video frames. However, our method generate a scene of a fire on firewood, aligning well with audio inputs. User study further demonstrates that our video quality surpass existing methods in Appendix Section E.

4 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, 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, 15%), Basic Science Research Program (NRF-2021R1A6A1A13044830, 15%), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(2022-0-00043, 20%). This work are results of a study on the "Leaders in INdustry-university Cooperation 3.0" Project, supported by the Ministry of Education and National Research Foundation of Korea (tel:1345370620, 50%).

References

Avrahami, O., Fried, O., and Lischinski, D. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022.

Brack, M., Schramowski, P., Friedrich, F., Hintersdorf, D., and Kersting, K. The stable artist: Steering semantics in diffusion latent space. *arXiv preprint arXiv:2212.06013*, 2022.

Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Chatterjee, M. and Cherian, A. 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*, pp. 701–719. Springer, 2020.

Chen, H., Xie, W., Vedaldi, A., and Zisserman, A. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 721–725. IEEE, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., and Cohen-Or, D. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022b.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jeong, D., Doh, S., and Kwon, T. Träumerai: Dreaming music with stylegan. *arXiv preprint arXiv:2102.04680*, 2(4):10, 2021.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34: 852–863, 2021.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Le Moing, G., Ponce, J., and Schmid, C. Ccvs: context-aware controllable video synthesis. *Advances in Neural Information Processing Systems*, 34:14042–14055, 2021.

Lee, S. H., Oh, G., Byeon, W., Kim, C., Ryoo, W. J., Yoon, S. H., Cho, H., Bae, J., Kim, J., and Kim, S. Sound-guided semantic video generation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pp. 34–50. Springer, 2022a.

Lee, S. H., Roh, W., Byeon, W., Yoon, S. H., Kim, C., Kim, J., and Kim, S. Sound-guided semantic image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3377–3386, 2022b.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. 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*, pp. 423–439. Springer, 2022.

Molad, E., Horwitz, E., Valevski, D., Acha, A. R., Matias, Y., Pritch, Y., Leviathan, Y., and Hoshen, Y. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Prajwal, K., Mukhopadhyay, R., Namboodiri, V. P., and Jawahar, C. 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*, pp. 484–492, 2020.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Ronneberger, O., Fischer, P., and Brox, T. 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*, pp. 234–241. Springer, 2015.

Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Skorokhodov, I., Sotnikov, G., and Elhoseiny, M. Aligning latent and image spaces to connect the unconnectable. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14144–14153, 2021.

Skorokhodov, I., Tulyakov, S., and Elhoseiny, M. 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*, pp. 3626–3636, 2022.

Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Villegas, R., Babaeizadeh, M., Kindermans, P.-J., Moraldo, H., Zhang, H., Saffar, M. T., Castro, S., Kunze, J., and Erhan, D. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

Wu, Q., Liu, Y., Zhao, H., Kale, A., Bui, T., Yu, T., Lin, Z., Zhang, Y., and Chang, S. Uncovering the disentanglement capability in text-to-image diffusion models. *arXiv preprint arXiv:2212.08698*, 2022.

A Related Work

Latent Diffusion Models. Recent success (Rombach et al., 2022) suggests that the Latent Diffusion Models (LDM) improve the efficiency of the diffusion process, successfully generating 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 (Avrahami et al., 2022) or by utilizing semantic information of the cross-attention layers (Hertz et al., 2022). Wu (Wu et al., 2022) used linear combinations of text embeddings and Liu (Liu et al., 2022) proposed composable diffusion models, but they still remained challenging to control fine-grained semantic changes. Recently, Semantic Guidance (SEGA) (Brack et al., 2022) 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 (Singer et al., 2022), Video Diffusion Models (Ho et al., 2022b), Imagen video (Ho et al., 2022a), and Phenaki (Villegas et al., 2022) 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 (Skorokhodov et al., 2022) and Dreamix (Molad et al., 2023), 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(Chatterjee & Cherian, 2020) and CCVS (Le Moing et al., 2021) generate video frames conditioned on the (non-temporal) context of the given audio, while TräumerAI (Jeong et al., 2021) utilized the magnitude of the given audio. Recently, Lee (Lee et al., 2022a) explored a model that can consider audio semantics as a condition to drive a video generator. Also, their dependency on StyleGAN (Skorokhodov et al., 2022)-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.

Table 2: 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	Input	Temporal Semantics	Magnitude	Domains (Audio Type)
Sound2Sight (Chatterjee & Cherian, 2020)	1st Frame	-	✓	Closed
CCVS (Le Moing et al., 2021)	1st Frame	-	✓	Closed (Music)
TräumerAI (Jeong et al., 2021)	-	-	✓	Closed (Music)
Lee (Lee et al., 2022a)	-	✓	-	Closed (Nature)
Ours	Latent Vector	✓	✓	Open Domains

B Method Details

B.1 Preliminary: Latent Stable Diffusion

Latent Diffusion Models (LDMs) use an encoder to convert a noised latent vector z^T to a denoised latent vector $z = x + \epsilon_\theta$, with z being a latent vector of an input image x and ϵ representing noise. Stable Diffusion (Rombach et al., 2022) is a part of a conditional generation model that uses U-Net(Ronneberger et al., 2015) as denoising autoencoders, denoted as ϵ_θ. To generate an output image, the autoencoder takes three inputs: noised latent vector z^t, a sequence t, and a conditional input y. The autoencoder $\epsilon_\theta(z^t, t, \tau_\theta(y))$ sequentially denoises z^t from $t = T$ to $t = 1$, where y is first transformed into a latent vector 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. The resulting denoised latent vector $z(= z^1)$ is transmitted to the decoder, which produces the final output image $\tilde{x}$.

B.2 Details of Audio Encoder

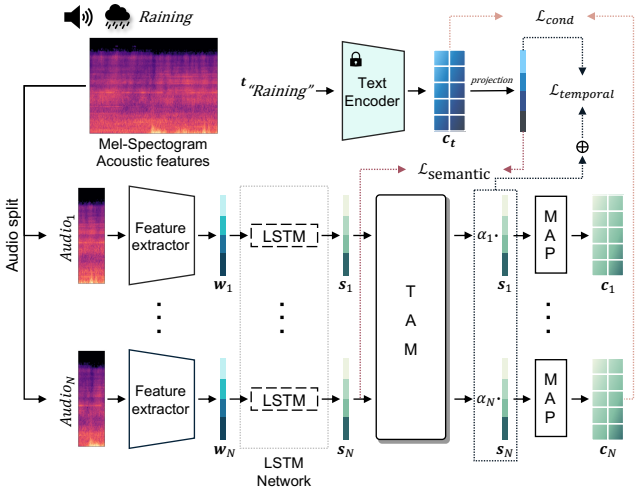

Figure 4: An overview of our Audio Encoder training process. Our model generates temporally-encoded audio embeddings with an LSTM (Hochreiter & Schmidhuber, 1997) layer and Temporal Attention Module (TAM). Audio input is partitioned into N segments, and each of these is encoded and used as a condition 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.

Enriching Audio Semantic Features by Augmentation. 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 (Park et al., 2019) to apply random transformations (such as masking our certain frequency bands or time segments), yielding augmented audio inputs. We further add the InfoNCE loss (Oord et al., 2018) $\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow a'}$ to pull augmented audio features together.

$$\mathcal{L}_{\text{CLIP}}^{a \leftrightarrow a'} = l_{sim}(\mathbf{s}_N, \mathbf{s}'_N) + l_{sim}(\mathbf{s}'_N, \mathbf{s}_N) \tag{5}$$

where apostrophe indicates augmented view of an original audio data.

Mapping Module. The Mapping Module, denoted as `MAP` in our main paper, consists of several MLP layers, which consist of Linear-Linear-Dropout-GELU layers. The purpose of this module is to align the audio embeddings with textual prompt in Stable Diffusion (Rombach et al., 2022). The prompt is converted into a sequence vector via the conditional encoder in Stable Diffusion, which is transformers as CLIP-L/14 (Radford et al., 2021) Text Encoder. Since audio embeddings from the Temporal Attention Module is not sequence-like vectors, we use the Mapping Module to broaden the dimensions like text embeddings (e.g. from `<SOS>` token to `<EOS>` token). To achieve this, MSE loss is used to align the audio embeddings (e.g. raining sound) with the text sequence embeddings of the audio class (e.g. "Raining") from CLIP-L/14. Specifically, to obtain sequence-like vectors, the `<SOS>` token is removed from the text embeddings, which is the same for all prompts. Later, we concatenate the `<SOS>` token with the converted audio embeddings to feed the audio condition into Stable Diffusion in the inference stage.

B.3 Details of Video Frame Generation

Temporal Frame Interpolation. We use an interpolated latent vector to generate continuous video frames between two consecutive frames. Following the work by Ramesh . (Ramesh et al., 2022), 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.

Details of Audio Semantic Guidance Module. The denoising autoencoder ϵ_θ is executed $\delta - 1$ times out of T times to form incomplete noise along with the original text prompt meaning. From $t = \delta$, the audio semantic guidance operates through the following equation:

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

where z^δ is denoised random noised 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 $t = \delta$. Note that z^T is fixed through frames in one video. The audio semantic guidance $\lambda(\mathbf{z}^t, \mathbf{c}_n)$ is defined as follows:

$$\lambda(\mathbf{z}^\delta, \mathbf{c}_n) = g_s \psi(\mathbf{z}^\delta, \mathbf{c}_p, \mathbf{c}_n) + s_m \Phi_m \tag{7}$$

where $s_m \in [0, 1]$ is the momentum hyper parameter that scales the momentum Φ_m. To determine the audio semantic guidance direction in the Stable Diffusion latent space, the semantic difference $\psi(\mathbf{z}^\delta, \mathbf{c}_p, \mathbf{c}_n)$ between the guidance provided by $\mathbf{c}_n$ and the unconditioned guidance $\mathbf{c}_\varnothing$ is used:

$$\psi(\mathbf{z}^\delta, \mathbf{c}_p, \mathbf{c}_n) = \epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_n) - \epsilon_\theta(\mathbf{z}^\delta_\varnothing, \mathbf{c}_\varnothing) \tag{8}$$

The data distribution function $\psi(\mathbf{z}^t, \mathbf{c}_p, \mathbf{c}_n)$ is used to identify the audio part that needs to be modified from the original prompt. First, the difference between the concept-conditioned and unconditioned estimates, denoted as ψ, is scaled. Then, the values in the upper and lower tail are used as the dimension that represent the specified concept. Therefore, the location to be changed can be obtained, and it can be expressed as:

$$g_s = \begin{cases} s_e, & \text{where } |\psi| \geq \eta_\lambda(|\psi|) \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

where $\eta_\lambda(|\psi|)$ indicates that it will cause a change in the distribution by (100 - ψ)%, and s_e determines the degree of the semantic guidance effect.

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.

C Implementation Details

Training details Our end-to-end Audio Encoder model is trained using a combination of Adam (Kingma & Ba, 2014) optimizer and SGD (Ruder, 2016) optimizer. While the Mapping module is trained with Adam optimizer, the remaining modules are trained with SGD optimizer. We distribute the inputs evenly across 4 NVIDIA GeForce RTX 3090 GPUs and train the entire model for 24 epochs. We use the VGG-sound dataset (Chen et al., 2020) and Landscape (Lee et al., 2022a) for training our model. The Audio Encoder is trained with hyperparameters such as a learning rate of 0.001, a batch size of 160, a weight decaying parameter of 0.0005, dropout of 0.2 and a momentum of 0.9 for the SGD optimizer. We stress that our Audio Encoder has not been further fine-tuned for any specific task or experiment.

Inference details of Audio Semantic Guidance. To implement the Audio Semantic Guidance module following SEGA (Brack et al., 2022), three hyperparameters, namely δ, s_e, and ψ, are required. (The notation for each hyperparameter is defined in our main paper.) The parameter δ controls the degree of preservation of the original prompt. Lower values of δ correspond to a greater preservation of the original prompt and $\delta = T$ means no preservation of the original prompt. In our experiments, we set δ between 800 and 950 to balance the preservation of the original prompt with the visualization of the effect of audio semantics. The s_e hyperparameter represents the degree of the scale of audio semantics effects. We set s_e between 2.5 and 8 in our experiments. However, the s_e hyperparameter is not related to the areas that need to be changed. Instead, it is related to the ψ parameters. ψ is a value between 0 and 1, and we set it between 0.8 and 0.99. Note that we did not adjust these hyperparameters when creating a single video. Since the distribution of each data is different, we need to make modifications to the hyperparameters for the purpose of improving the visualization of audio semantics and ensuring that the original prompt remains preserved.

D Experiments details

Datasets. We use two Audio-Video datasets to train our Audio Encoder: VGG-Sound (Chen et al., 2020) and Landscape (Lee et al., 2022a). 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 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.

Baselines Setup. To generate videos using Sound2Sight (Chatterjee & Cherian, 2020) and CCVS (Le Moing et al., 2021), we randomly select the first frame from the Landscape dataset (Lee et al., 2022a) since they need first frame to generate the video. For TräumerAI (Jeong et al., 2021) and Sound-guided Video Generation (Lee et al., 2022a), we pre-train StyleGAN (Karras et al., 2021) with the LHQ dataset (Skorokhodov et al., 2021) and then train the models on the Landscape dataset (Lee et al., 2022a). To ensure a fair comparison, we randomly sample a prompt related to landscapes for our method to generate landscape-like videos.

Evaluation Metrics. Two video quality metrics are used for evaluations, Fréchet Video Distance (FVD) (Unterthiner et al., 2018) and Inception Score (IS) (Salimans et al., 2016). FVD measures the distribution gap between real and synthesized videos in the latent space and is implemented by fine-tuning Inflated 3D ConvNet (Carreira & Zisserman, 2017) with the Landscape (Lee et al., 2022a) dataset. IS is used to evaluate GAN-generated images by computing KL-divergence and is implemented using pre-trained InceptionNet (Kay et al., 2017) trained on the ImageNet (Deng et al., 2009) dataset. Additionally, CLIP (Radford et al., 2021)-based cosine similarity is measured between audio and image as well as text and image, and a textual pivot feature is obtained by feeding the prompt "The photo of `<class>`" into CLIP text encoder. For fair comparison with other existing baselines such as TräumerAI (Jeong et al., 2021) and Sound-guided Video Generation (Lee et al., 2022a) which does not get a hint about what to generate, we use prompt that originally does not generate the semantics of sound. We set fps 20 and generated videos to extract images from all baselines.

E User Study

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 (Chatterjee & Cherian, 2020), CCVS (Le Moing et al., 2021), TräumerAI (Jeong et al., 2021), Lee (Lee et al., 2022a), and ours. Participants are asked to evaluate the given video frames in terms of realism, vividness, consistency, and relevance between audio and video on a five-point scale, ranging from "1 - very unrealistic" to "5 - very realistic," "1 - very unvivid" to "5 - very vivid," "1 - very inconsistent" to "5 - very consistent," and "1 - very irrelevant" to "5 - very relevant," respectively. Specifically, we ask participants "On a scale of 1 to 5, how realistic the video is? Please rate the realism, with 1 being very unrealistic and 5 being very realistic", "On a scale of 1 to 5, how vibrant does the video appear? Please rate the vividness, with 1 being not vibrant at all and 5 being extremely vibrant.", "On a scale of 1 to 5, how well does the movement in the video match the audio levels? Please rate the consistency, with 1 being very inconsistent and 5 being very consistent.", and "On a scale of 1 to 5, how relevant video with the audio sound? Please rate the relevance, with 1 being not relevant and 5 being very relevant.". The order of videos within each question is randomized to prevent participants from inferring the unique quality of each baseline. We observe in Figure 5 that our proposed method outperforms the other approaches in all categories. These results are consistent with our quantitative and qualitative results.

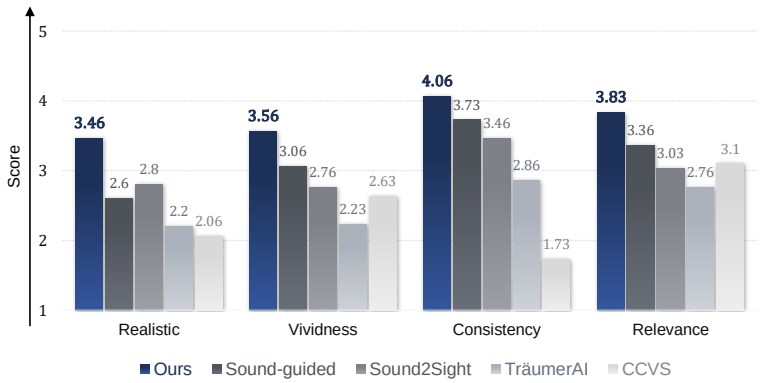

Figure 5: Our human evaluation results. We conduct a user study with 100 participants on Amazon Mechanical Turk (AMT). Participants are shown generated video frames and asked to evaluate them in terms of realistic, vividness, consistency, and relevance. The Likert scale is used (higher is better).

F More Qualitative Results

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 6 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 that is shown in Figure 7. We generate video frames without Audio Semantic Guidance by following equation:

$$\tilde{\epsilon_\theta}(\mathbf{z}_n^\delta, \mathbf{c}_p) := \epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_\varnothing) + g(\epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_p) - \epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_\varnothing)) + \epsilon_\theta(\mathbf{z}^\delta, \mathbf{c}_n) \tag{10}$$

where $\delta = T$, which implies not preserving the prompt. Audio Semantic Guidance $\lambda(\mathbf{z}^\delta, \mathbf{c}_n)$ has been converted to $\epsilon_\theta(\mathbf{z}^T, \mathbf{c}_n)$ to remove the effect of Audio Semantic Guidance (Refer the notation in our main paper).

By leveraging Audio Semantic Guidance, we can generate sequential video frames that has consistent content (e.g. grass on the ground) yet is able to represent natural temporal variations (e.g. wave of water) by audio sound (e.g. waterfall burbling sound). On the other hand, without Audio Semantic Guidance, the prompt would struggle to maintain the content, leading the substantial changes (e.g. grass $\rightarrow$ toil) in the context. In addition, it is unable to manipulate the audio content based on its semantic meaning, which could lead to incorrect manipulations. (e.g. water $\rightarrow$ road). By guiding Stable Diffusion to generate audio semantics while maintaining consistent content, we can manipulate specific areas with audio semantics, resulting in enhanced naturalness.

Experiments of Semantic Transition. In Figure 8, we provide examples where audio inputs are changed (e.g., bird singing $\rightarrow$ wind noise, strong wind $\rightarrow$ 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.

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 9, 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 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 9 (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.

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 10, 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.

Comparison to Baselines. In Figure 11, we provide more examples of generated video frames by (from top) Sound2Sight (Chatterjee & Cherian, 2020), CCVS (Le Moing et al., 2021), TräumerAI (Jeong et al., 2021), Lee (Lee et al., 2022a), and ours. Furthermore, we compare our methods with StyleGAN (Karras et al., 2021) based TräumerAI (Jeong et al., 2021) and Sound Guided Video Generation (Lee et al., 2022a) in Figure 12. StyleGAN based methods both face challenges in effectively aligning audio semantics with latent space of StyleGAN despite of fine-tuning. On the contrary, our model can express the audio semantic meanings in multiple domains thanks to the rich latent space of Stable Diffusion models. Furthermore, compared to other baselines, our model is able to manipulate certain areas (e.g. fire on the stove top) via Audio Semantic Guidance through multiple denoising steps in Stable Diffusion. Our experiment reveals that our method can generate videos that have significant relevance and consistency with audio sound.

Additional Qualitative Examples. Figure 13 and Figure 14 show our model can generate video frames in diverse domains. Furthermore, Figure 15 and Figure 16 demonstrate the semantic consistency between sound and video. Morevover, our model can generate multiple high-fidelity frames naturally by the interpolation in Figure 17. Lastly, We further provide the whole sequences of video frame in Figure 18 at fps 15. It is important to note that we did not conduct any additional training on our Audio Encoder or Stable Diffusion.

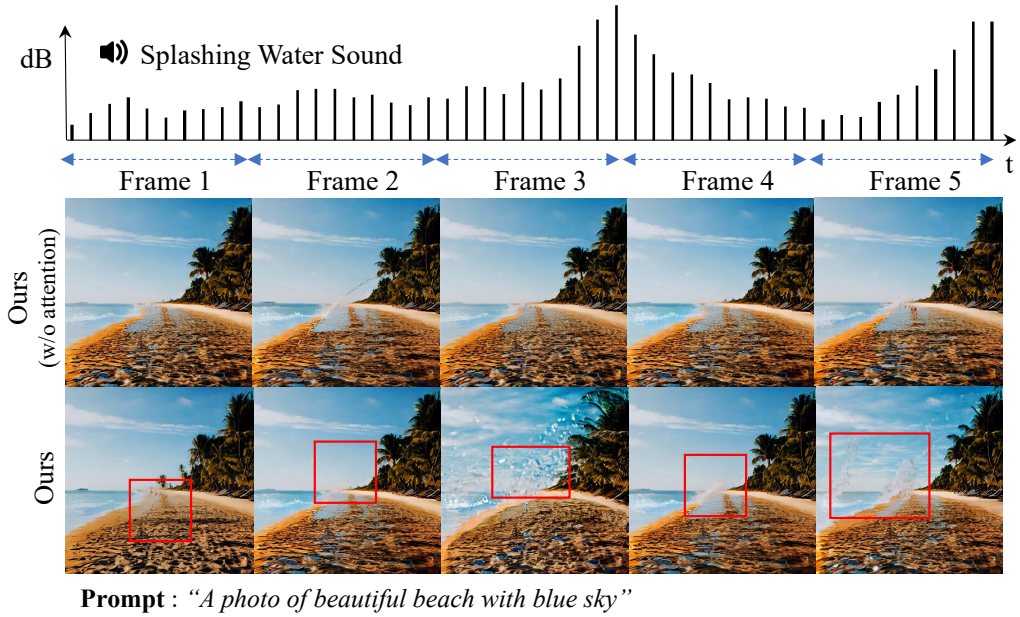

Figure 6: Generated video frames *with* and *without* our Temporal Attention Module. We generate video frames with splashing water sound where its amplitude changes over time (see top).

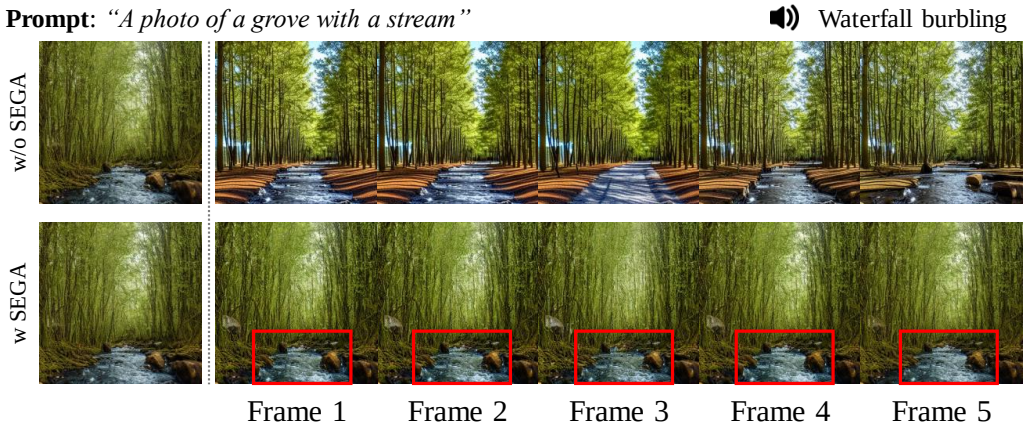

Figure 7: Generated video frames with and without our Audio Semantic Guidance. We generate video frames with waterfall burbling sound where the accurate modifications is highlighted with red box (see second row).

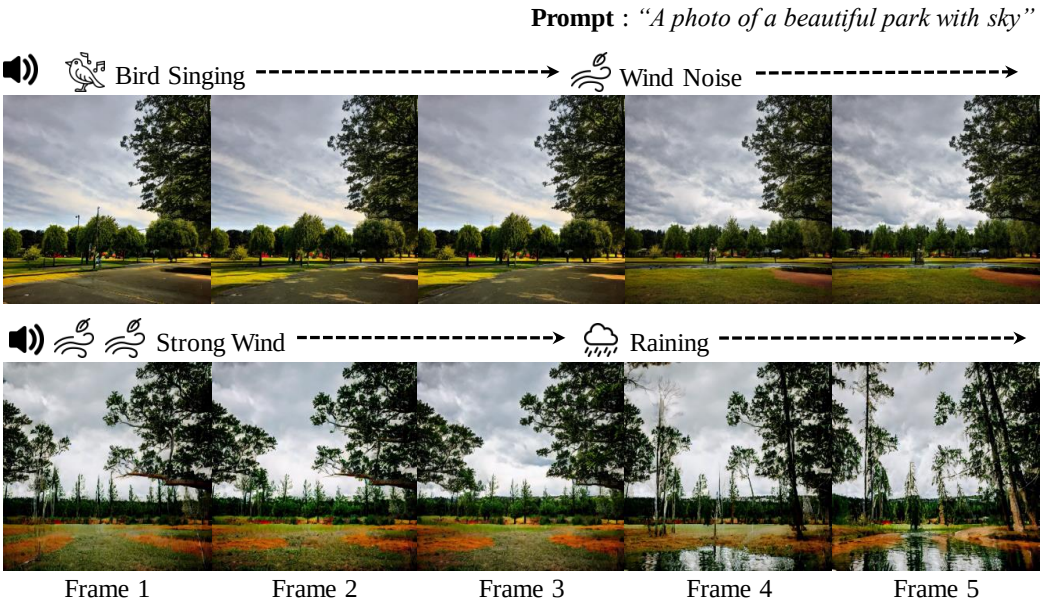

Figure 8: Examples of generated video frames with a sound that changes over time (e.g. bird singing → wind noise).

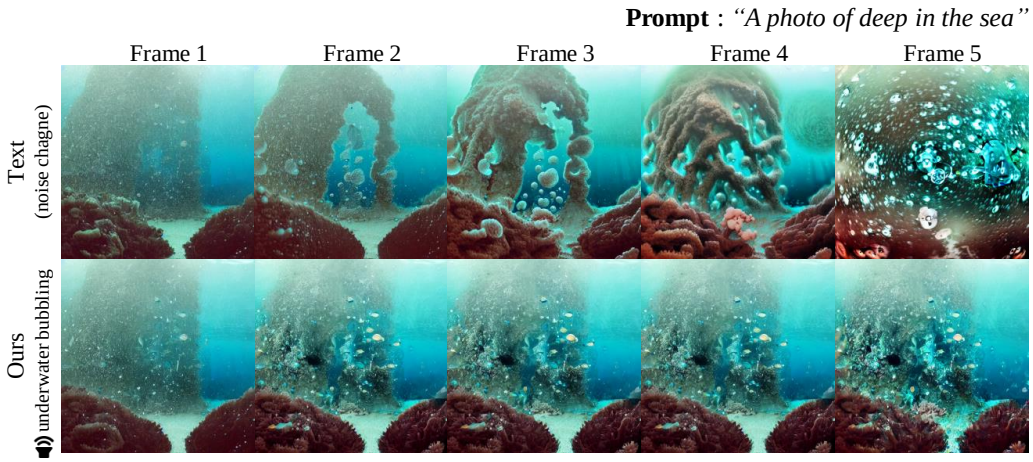

Figure 9: Examples of our video generation conditioned on text prompt (top) and audio (bottom).

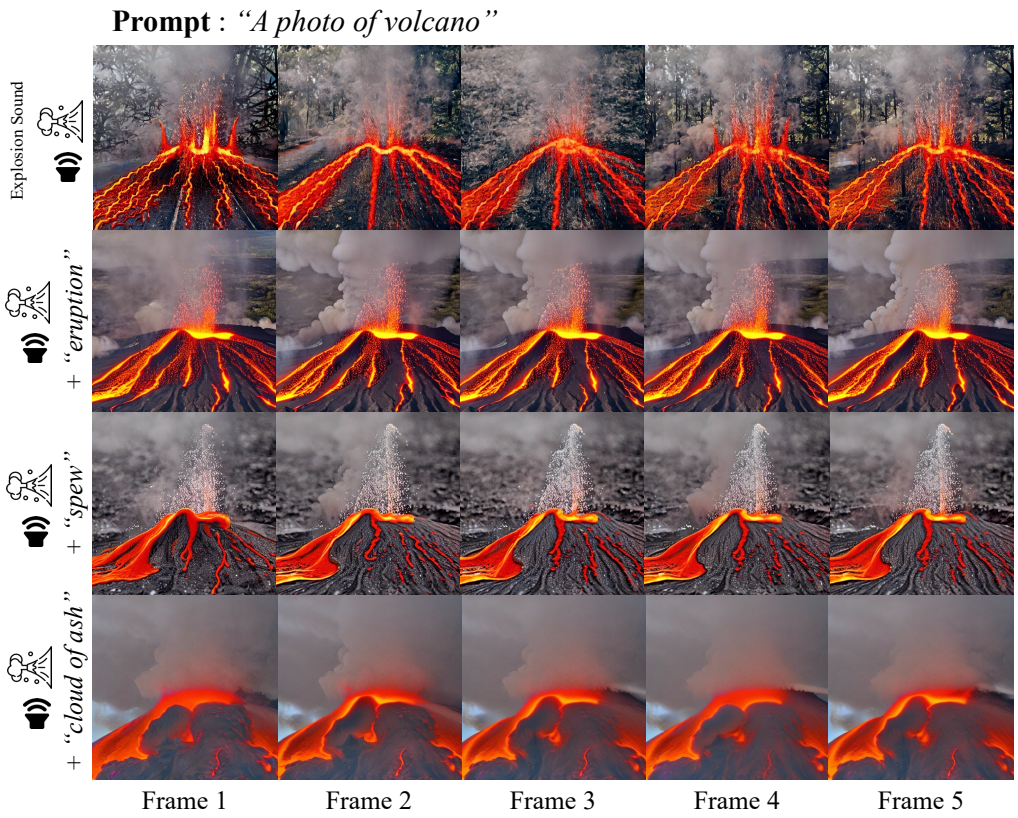

Figure 10: Example of generated videos with audio-text joint condition (e.g., 2nd row: conditioned with text "eruption" and explosion sound)

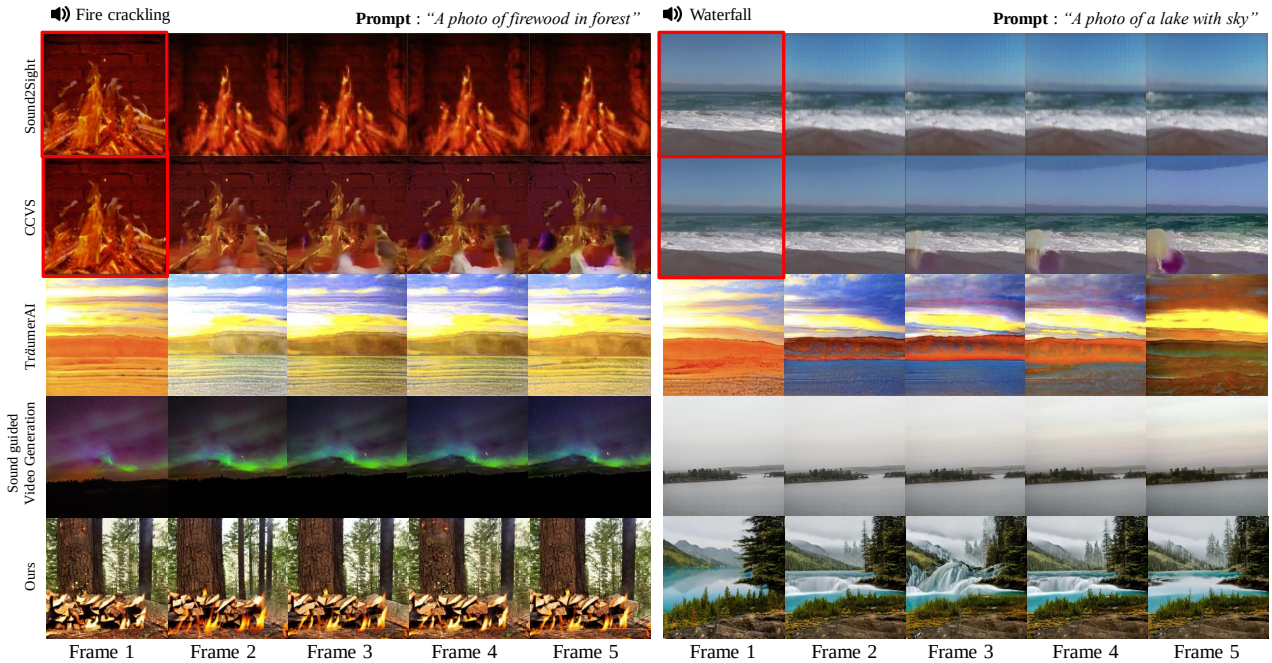

Figure 11: Examples of generated video frames (given fire crackling and waterfall audio) by Sound2Sight (Chatterjee & Cherian, 2020), CCVS (Le Moing et al., 2021), TräumerAI (Jeong et al., 2021), Lee (Lee et al., 2022a), and ours. Note that Sound2Sight and CCVS use an initial frame (highlighted in a red box).

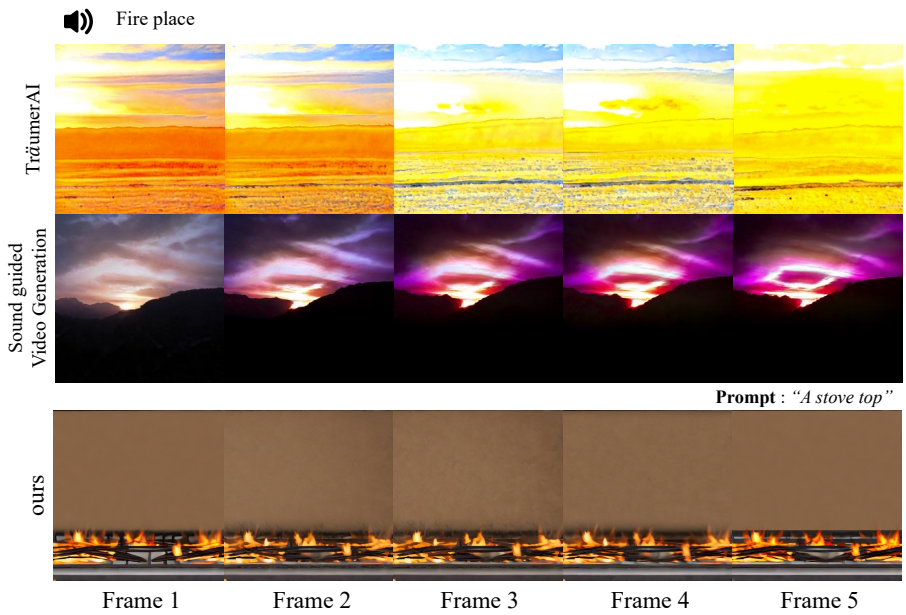

Figure 12: Comparison with StyleGAN (Skorokhodov et al., 2022)-based method. First row and second row represent video frames from TräumerAI (Jeong et al., 2021) and Sound guided Video Generation (Lee et al., 2022a). The last row shows video frames which are generated from our model.

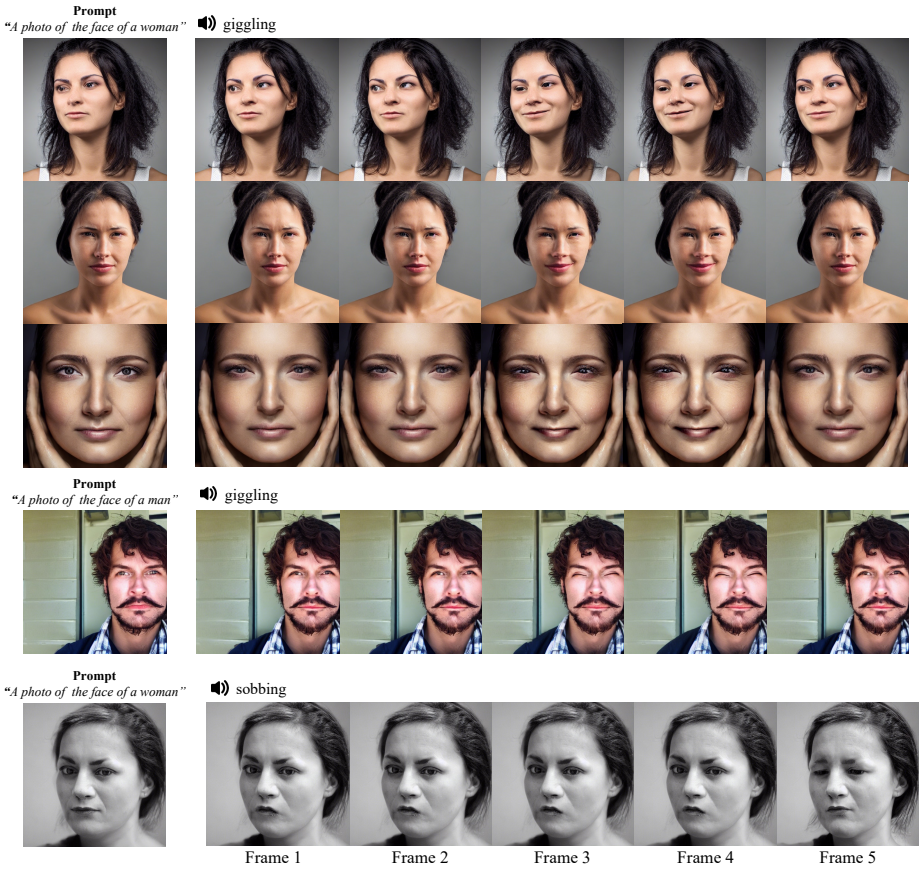

Figure 13: Examples of face generation with our methods. The sound of giggling and sobbing are used.

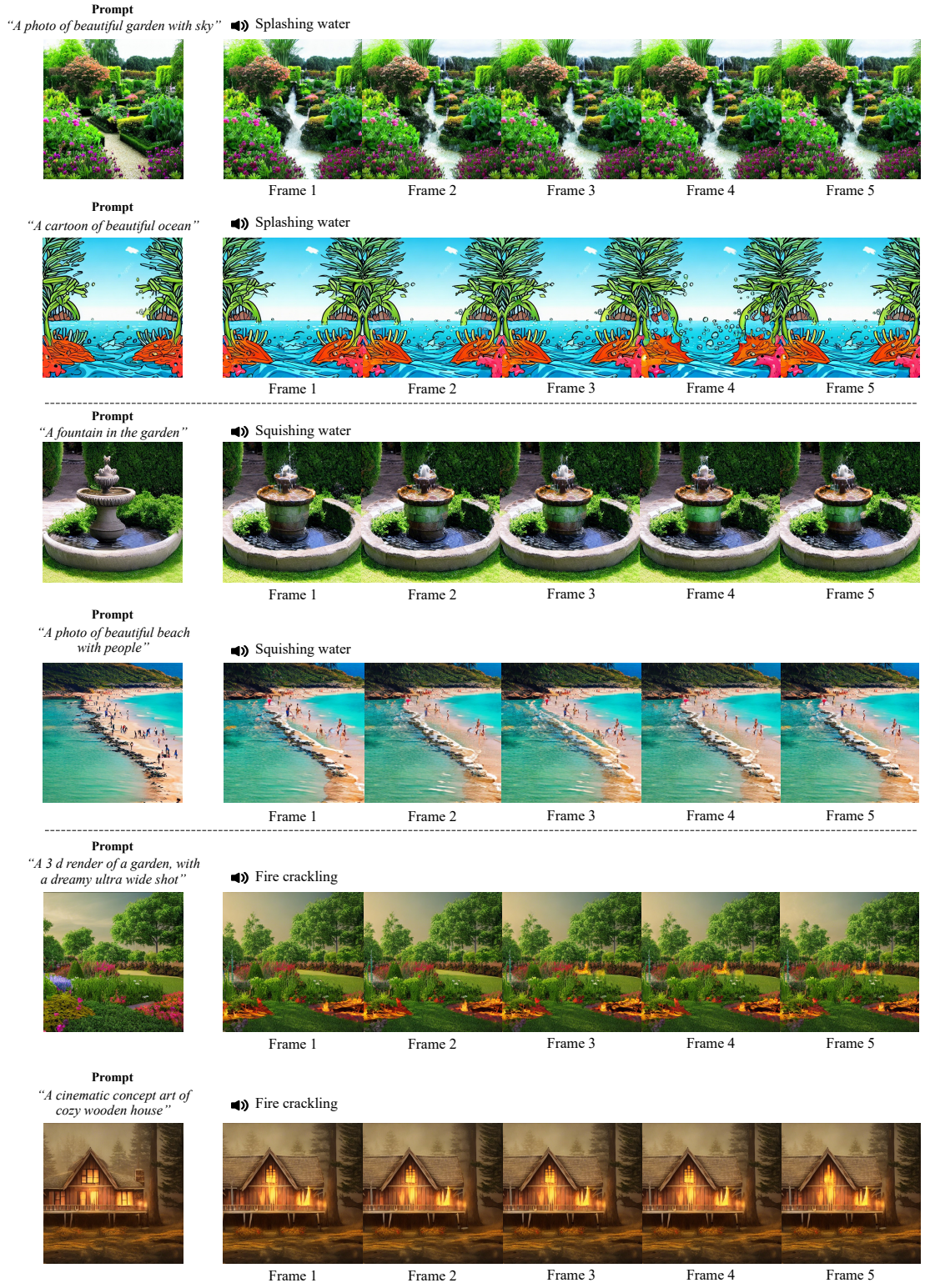

Figure 14: Examples of diverse examples in open domains. The sound of splashing water, squishing water and fire crackling are used.

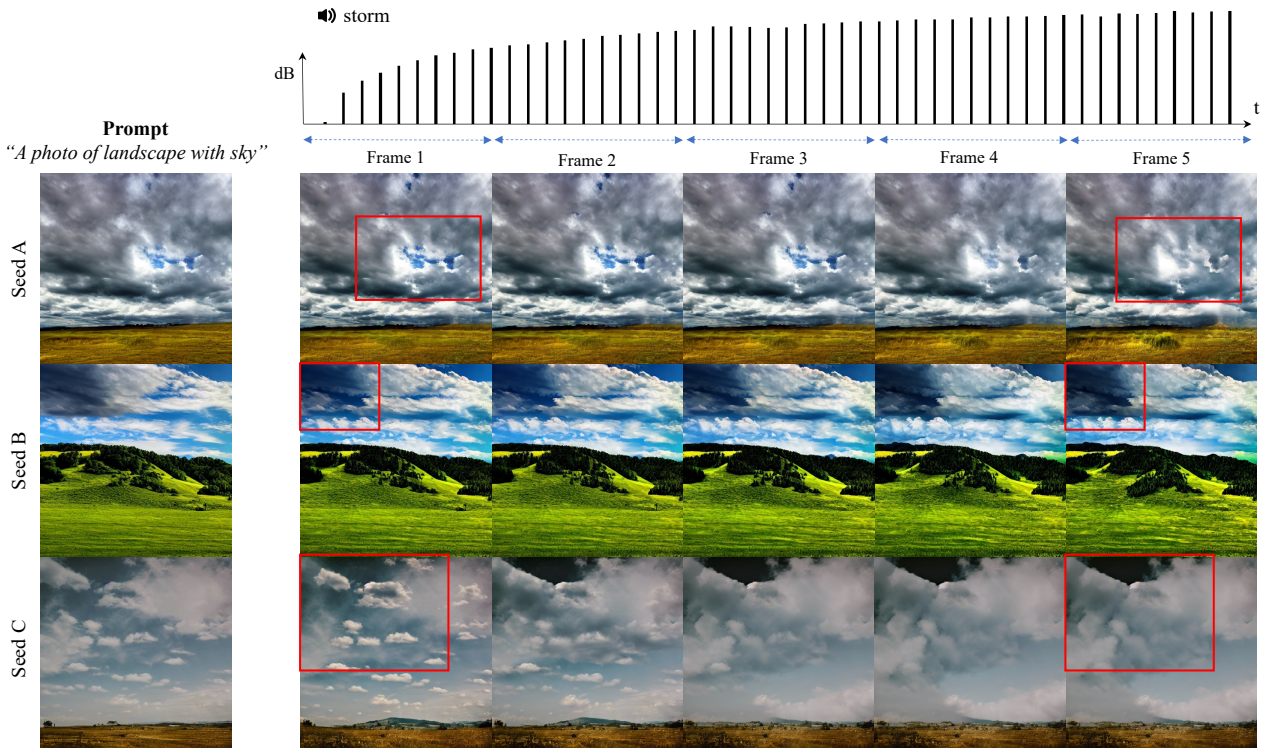

Figure 15: Example of video frames with multiple seed numbers. We regulate the prompt and audio sound as a given input feature and change a seed number randomly. The video frames are temporally consistent with the magnitude of audio.

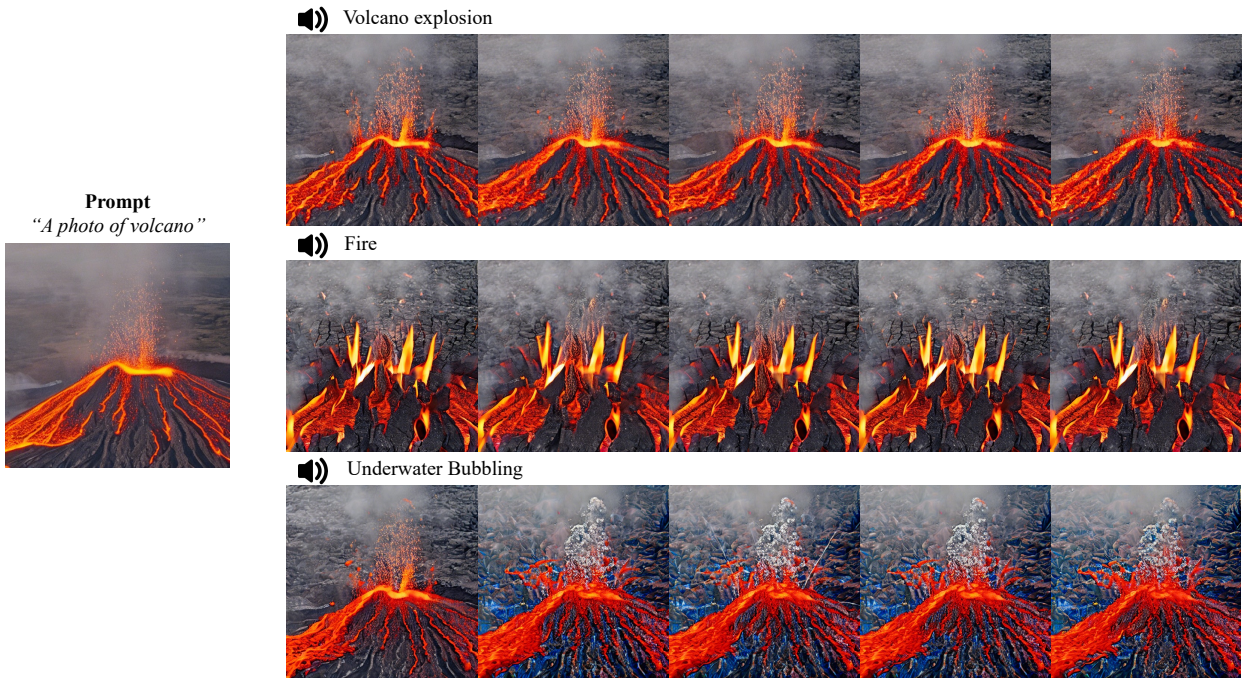

Figure 16: Example of video frames with different sound. The video frames are consistent and relevant with the audio semantics.

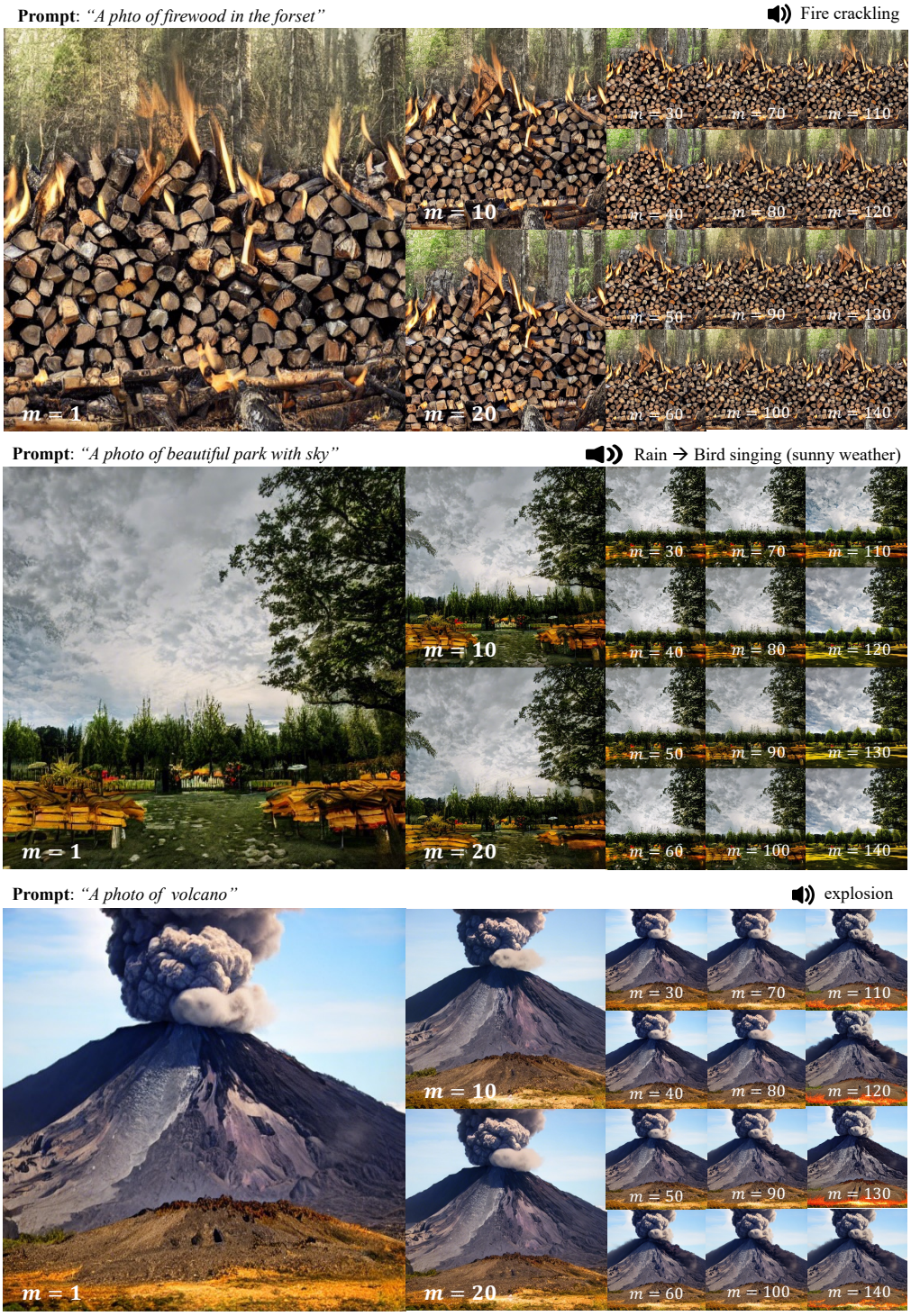

Figure 17: Example of video frames with interpolation module. A number of video frames are generated reactively by audio sound.

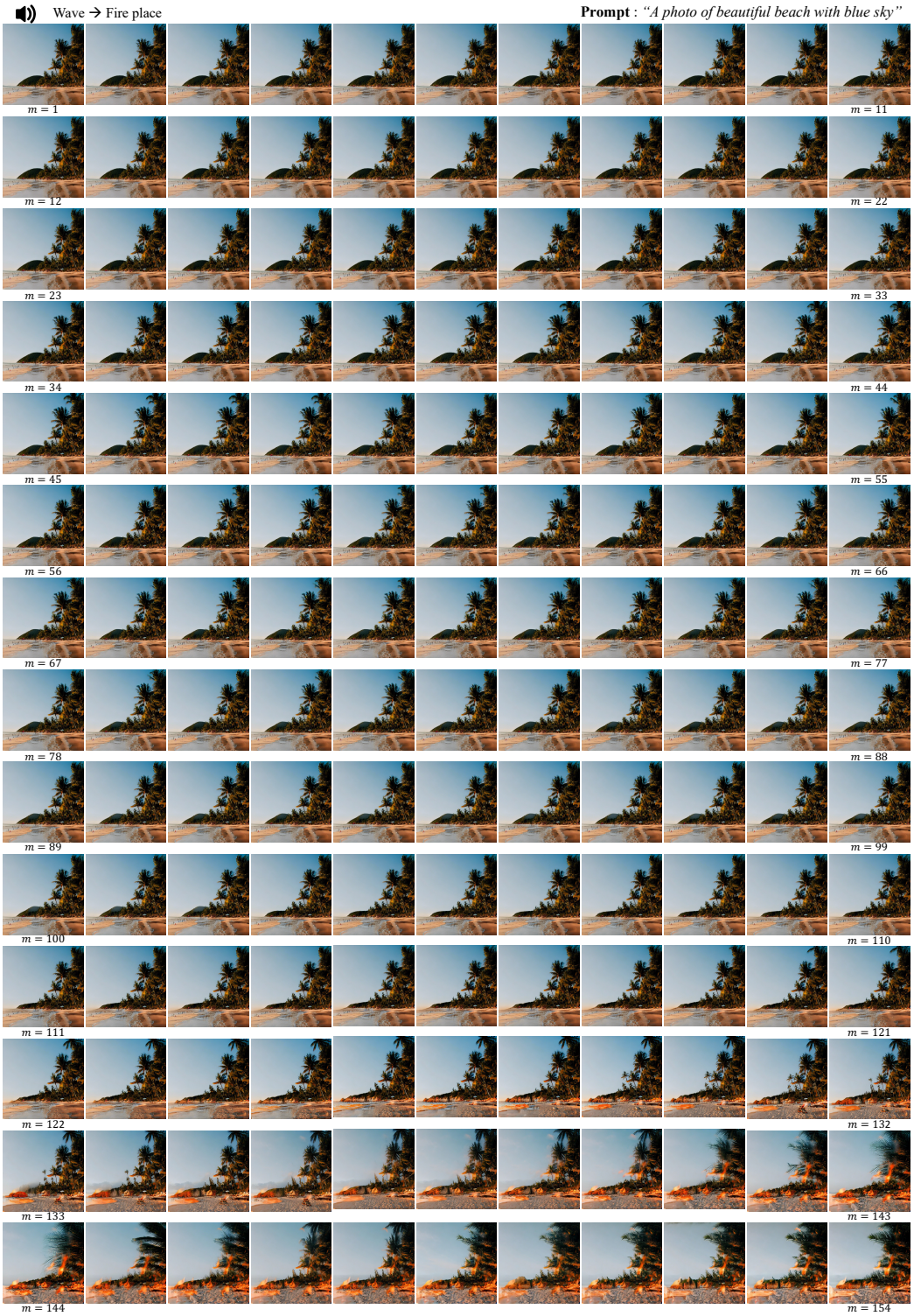

Figure 18: Whole sequence of video frames conditioned by the sound that has a semantic change from wave to fire place.