INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

GRAVO: Learning to Generate Relevant Audio from Visual Features with Noisy Online Videos

Youngdo Ahn[1*], *Chengyi Wang*[2], *Yu Wu*[3], *Jong Won Shin*[1], *Shujie Liu*[3]

[1]Gwangju Institute of Science and Technology, Korea
[2]Nankai University, China
[3]Microsoft, China

ayoungdo@gm.gist.ac.kr

Abstract

Given a video, previous video-to-audio generation methods use a hierarchical auto-regressive language model to produce a sequence of audio tokens to be decoded into a waveform. The audio generation depends only on the previous audio token and the current image but ignores the surrounding images that may have useful information. To learn the relationships between image frames, in this paper, we introduce GRAVO (**G**enerate **R**elevant **A**udio from **V**isual features with **O**nline videos), which employs multi-head attention (MHA) to encode rich context information and guide the audio decoder to produce more accurate audio tokens. Moreover, two auxiliary losses are introduced to explicitly supervise the MHA behavior, maximizing the similarity between the MHA output vector and the target waveform representation while preserving the original visual semantic information. Experimental results demonstrate that GRAVO surpasses state-of-the-art models on ImageHear and VGG-Sound datasets.

Index Terms: audio generation, multi-modality, translation, language model, pre-trained models

1. Introduction

Deep models for generation tasks have become popular in both academia and industry [1, 2, 3, 4], such as using large-scale data and powerful neural networks to generate realistic visuals based on the text descriptions [5, 6]. Recently, interest has grown in using these models to generate audio based on images [7, 8, 9, 10], which can be useful in various applications, such as creating sound effects for video games and animations, or making images more accessible to blind and visually impaired users.

Prior attempts in this line have developed models with a pre-determined set of (less than 20) sound classes, leveraging labeled data for model training [11, 12, 13]. Such models only work well on the pre-defined sound classes for which they have been trained and perform poorly on any unseen classes due to the limited scope of their training data. Recently, [14] and [15] proposed label-free approaches, allowing open-domain visually guided audio generation. Specifically, conditioned on the output of an image classifier, [14] generates Mel-spectrograms using SpecVQGAN, which is then decoded into a waveform using a neural vocoder. As the most recent state-of-the-art method, Im2Wav [15] takes advantage of a large amount of image-audio pairs from online videos on the web. Im2Wav is a Transformer-based audio Language Model using image representation of the pre-trained CLIP [16] as input. Based on previous acoustic tokens and temporally-aligned visual features, Im2Wav generates audio tokens, which are converted into waveforms using the corresponding audio decoder from the pre-trained VQ-VAE model. Sometimes, the subject of interest in an image and its corresponding audio may not match up perfectly. This can occur when a subject such as an animal is moving out of the camera's view but its sound can still be heard, or an object appears in the center of the current image but does not make any sound. Given the aforementioned mismatches between images and audio, it can be argued that using image representations that are temporally aligned as input could lead to incorrect results during the generation process.

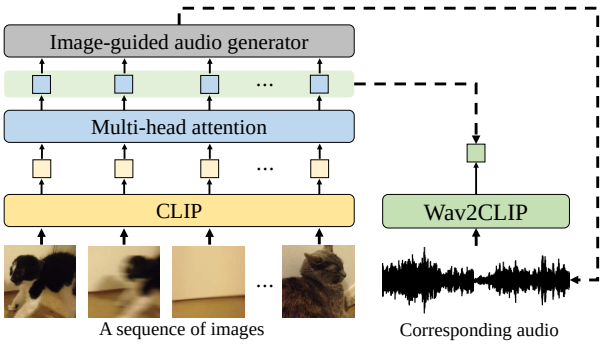


Figure 1: *The overview of GRAVO. CLIP embeddings are extracted from a given sequence of images and transformed through multi-head attention (MHA). The transformed embeddings are used to generate corresponding audio in the same video via the image-guided audio generator. In the training phase, Wav2CLIP embedding is extracted from the target audio and used to guide MHA to extract relevant image features.*

In order to determine which objects in the video are audible and which are not, additional contextual information beyond the current image is needed. Following this motivation, we propose the GRAVO (**G**enerate **R**elevant **A**udio from **V**isual features with **O**nline videos) as shown in Fig. 1, introducing multi-head attention (MHA) on top of the visual features extracted with CLIP model. The introduced MHA mechanism in our GRAVO model enables each frame in the image sequence to have access to all the information across the entire sequence, thus enabling the model to automatically learn the inter-frame relationships and produce more precise information for audio generation. To further enhance the accuracy of our GRAVO model, we also incorporate two supplementary losses that guide the MHA mechanism to learn audio information by utilizing the output of the pre-trained Wav2Clip model [17] as a target for knowledge distillation. One loss is used to encourage the output of the MHA to be close to the Wav2Clip embedding. To retain the original

*This work was done during an internship at Microsoft.

10.21437/Interspeech.2023-1581

visual information while ensuring the alignment between image and audio representation, a second regularizer loss is introduced to minimize the variance of the learned visual features.

We evaluate GRAVO on VGG-Sound [18] and ImageHear [15] datasets, which are comprised of videos and single images, respectively. Experimental results demonstrated that GRAVO outperformed Im2Wav in both audio classification accuracy, with an improvement of 9.89%, and audio-visual similarity score, with an increase of 0.7 on ImageHear. On VGG-Sound, GRAVO achieves higher audio classification accuracy with an improvement of 3.8%, as well as a higher audio-visual similarity score of 0.37. Audio demo samples for the evaluation datasets and generated videos by [6] are available on our website[1].

2. Method

Formally, given an audio-video pairs dataset $\mathcal{D} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^N$, where $\mathbf{x}^i = (x_1^i, \cdots, x_T^i)$ is an audio sample with T time steps, and $\mathbf{y}^i = (y_1^i, \cdots, y_M^i)$ is the corresponding video with M frames. The goal of GRAVO is to generate high-fidelity audio for the given image or image sequence. Figure 2 sketches the overview of GRAVO. The model has three modules: the pre-trained image encoder extracts image representations, the attention-based conditional audio generator then converts these image features into a sequence of discrete tokens, and finally, the pre-trained audio decoder restores the waveform.

2.1. Visual and Audio Representation

We use the pre-trained CLIP [16] model as the image feature extractor and a pre-trained VQ-VAE model as the audio feature extractor.

The CLIP model is designed to learn the correlation between text and image pairs by maximizing their similarity score. Previous research demonstrates that CLIP effectively captures the underlying semantics of images [16] and performs admirably in tasks related to audio generation [15]. Given a sequence of images $\mathbf{x}$, the model produces a corresponding sequence of semantic representations, represented by $\mathbf{f}$.

The GRAVO model utilizes the pre-trained one-dimensional hierarchical VQ-VAE model from Im2Wav [15] as the audio feature extractor. It consists of an audio encoder, a quantizer with multi-level codebooks, and a decoder. The encoder encodes $\mathbf{x}$ into a series of latent vectors, $\mathbf{h}$, which are then divided into two-level representations with shorter sequence lengths, represented by $\mathbf{h} = [\mathbf{h}^{(1)}, \mathbf{h}^{(2)}]$. The quantizer converts these representations into two-level discrete tokens $[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}]$, with each level utilizing its own codebook. The decoder then recovers the audio waveform, which is conditioned on the discrete tokens. During the training of the GRAVO model, all audio in the dataset, $\mathbf{x} \in \mathcal{D}$, is tokenized and used as the predicted targets for the conditional audio generator. Finally, the pre-trained VQ-VAE decoder is employed as the wave reconstructor.

2.2. Conditional Audio Generator

The goal of the conditional audio generator is to predict the discrete audio tokens $\mathbf{z}$ given the sequence of image features $\mathbf{f}$. It comprises two auto-regressive language models for coarse-to-fine generation, referred to as Up and Low. The two language models are applied at different time resolutions. The Low model is responsible for determining the semantic information of the generation, while the Up model is tasked with completing the fine details. Previous research [15], conditions the Low model on the temporally aligned image representation, f_m, at each generation step. However, the synchronization of audio and image within a video is not always exact. Sometimes, an object of interest may not be visible on camera but its sound is still audible and in other cases, an object appearing in the center of the frame might not produce any sound. Therefore, not all image representations contribute equally to audio generation. To address this problem, we propose using a multi-head self-attention module (MHA) over the image representations. This MHA module enables each image representation to attend to the entire sequence, allowing the model to automatically learn and understand the relationships between all elements. At every time step, the Low model is conditioned on both the temporal aligned MHA output representation as well as the mean of them:

$$\mathcal{L}_{\text{Low}} = -\log p\left(z_t^{(2)} | \bar{\mathbf{f}}', f_t', z_{<t}^{(2)}; \theta_{\text{Low}}, \theta_{\text{MHA}}\right) \tag{1}$$

where f_t' is temporal aligned MHA output and $\bar{\mathbf{f}}'$ is the mean vector. The Up model is conditioned on the output of the Low model as well as the mean of the CLIP image representations:

$$\mathcal{L}_{\text{Up}} = -\log p\left(z_t^{(1)} | \bar{\mathbf{f}}, z_t^{(2)}, z_{<t}^{(1)}; \theta_{\text{Up}}\right) \tag{2}$$

During inference, the audio tokens from the Low model are used as the condition of the Up model. Then the audio tokens from the Up model are converted into the waveform through the pre-trained audio decoder.

2.3. Relevance Guided Multi-head Attention

The proposed multi-head attention module improves the receptive field at each generation time step. However, without proper guidance, it can be difficult for the model to identify the most relevant image representation that corresponds to the present audio segment. To enhance the model's performance, we propose the use of two auxiliary losses, which utilize the Wav2CLIP [17] embedding to direct the MHA to learn the audio-relevant image features, allowing for better alignment between image and audio.

Wav2CLIP is an audio representation learning model. It distills the CLIP model into an audio model, resulting in one joint embedding space for different modalities. Given an audio piece, Wav2CLIP can project it to a similar space to the CLIP representation of the relevant image. To guide the MHA to generate audio-relevant representation, we disclose the distance between the MHA output and the Wav2CLIP embedding:

$$\mathcal{L}_{\mathbf{w}} = -\sum_{m=1}^{M} \cos\left(\mathbf{w}, f_m'\right) \tag{3}$$

where $\mathbf{w}$ is the Wav2CLIP embedding.

The above criterion encourages every f_m' in $\mathbf{f}'$ to be close to the Wav2CLIP embedding, which may compromise the original visual semantic information contained within f_m. To prevent this loss of information, we propose another loss as a regularizer, which aims to retain the original visual information while ensuring the alignment between image and audio representation. Such a balance improves the overall performance of the model:

$$\mathcal{L}_{\hat{\mathbf{f}}} = \sum_{m=1}^{M} \|\hat{\mathbf{f}}_m - f_m'\|_2^2 \tag{4}$$

[1] https://GRAVO-demo.github.io

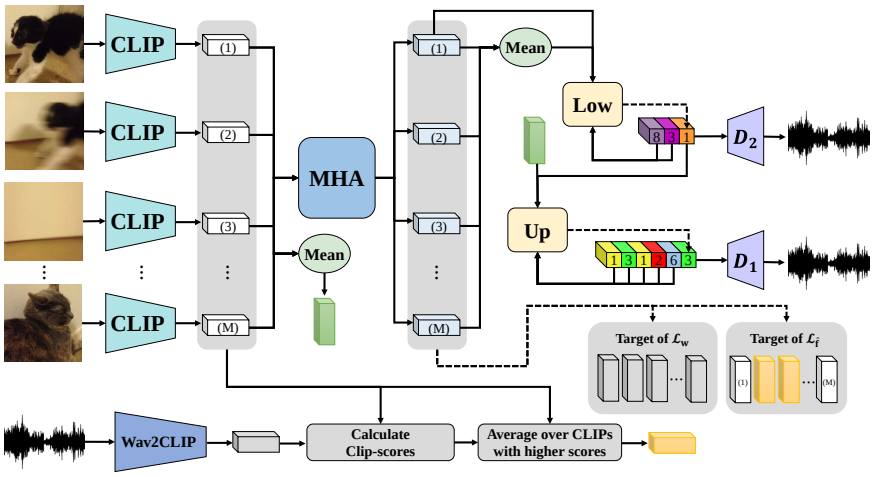


Figure 2: *The overall architecture of GRAVO. The output of multi-head attention (MHA) is used as a condition of Low, which is a causal language model. In the training phase, clip scores are calculated between Wav2CLIP and CLIP embeddings and used to extract a yellow block which is the average embedding over the embeddings with higher clip scores. The yellow block is used as one of the targets of MHA together with the Wav2CLIP embedding. Note that we do not use Wav2CLIP to generate audio in the inference process.*

Here, $\hat{\mathbf{f}}_m$ is either the CLIP representation f_m or the mean of $\mathbf{f}$, which is determined by the similarity score between the CLIP representation and the Wav2CLIP representation:

$$\hat{\mathbf{f}}_m = \begin{cases} f_m, & \text{if } \mathrm{NS}(\mathbf{w}, f_m) > \gamma \cdot \max \mathrm{NS}(\mathbf{w}, \mathbf{f}), \\ \tilde{\mathbf{f}}, & \text{otherwise}, \end{cases} \tag{5}$$

where NS represents z-normalized cosine similarity for each sample along image frames. γ is a threshold that determine whether f_m is audio-relevant visual feature or not. $\tilde{\mathbf{f}}$ is the average of embeddings whose NS score is above $\gamma = 0.9$ multiplied by the maximum value of the normalized scores.

With the proposed two losses, the overall training objective of the Low model is defined as

$$\mathcal{L}_{\text{GRAVO}} = \mathcal{L}_{\text{Low}} + \lambda_{\mathbf{w}} \mathcal{L}_{\mathbf{w}} + \lambda_{\hat{\mathbf{f}}} \mathcal{L}_{\hat{\mathbf{f}}} \tag{6}$$

where $\lambda_{\mathbf{w}}$ and $\lambda_{\hat{\mathbf{f}}}$ are hyper-parameters that we empirically use 0.0001 and 1000, respectively.

2.4. Classifier Free Guidance

We follow the previous work [15] to use the classifier free guidance method to control the trade-off between sample quality and diversity [19, 20]. As in [15], we replace f'_m with a learned-null embedding $f^{\emptyset}$ during training for each sample in the batch with probability $p = 0.5$. During inference on the Low model, we produce audio tokens using the summation of the probabilities with and without visual conditioning.

$$\begin{aligned} \log p(z_t^{(2)}) &= \lambda_{\mathbf{f}^{\emptyset}} + \eta(\lambda_{\mathbf{f}'} - \lambda_{\mathbf{f}^{\emptyset}}), \\ \lambda_{\mathbf{f}'} &= \log p(z_t^{(2)} | \bar{\mathbf{f}}', f'_t, z_{<t}^{(2)}), \\ \lambda_{\mathbf{f}^{\emptyset}} &= \log p(z_t^{(2)} | \bar{\mathbf{f}}^{\emptyset}, f^{\emptyset}, z_{<t}^{(2)}), \end{aligned} \tag{7}$$

where $\bar{\mathbf{f}}^{\emptyset}$ is the mean vector of null embeddings. η is the guidance scale that determines the trade-off between the diversity and quality of the generated audio characteristics. We use $\eta = 3$ that showed great performance in the fields of text-to-image [20] and text-to-audio [7] generation following [15].

3. Experiments

3.1. Data

We train our method on VGG-Sound [18] and evaluate on two datasets: VGG-Sound [18] and ImageHear [15].

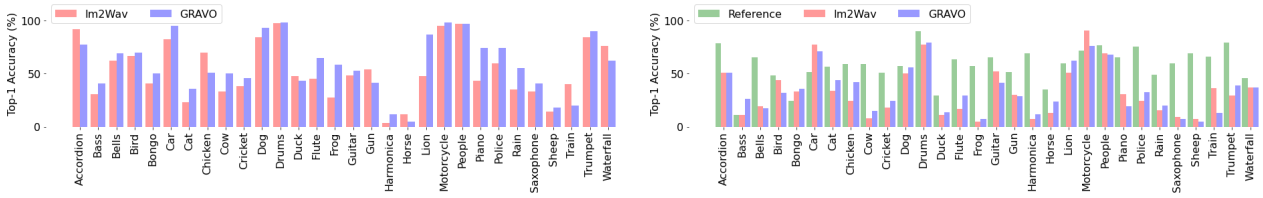
VGG-Sound is a large-scale dataset that is created by extracting audio and visual information from YouTube videos. It contains 200k 10-second videos from 309 classes. We follow the same training and testing split as the original VGG-Sound. The training set is divided into 0.9 and 0.1 ratios for the training and validation. During training, we randomly crop 4 seconds of video from each clip. For evaluation, we use only the initial 4 seconds of each clip. Most of the videos in the dataset have a frame rate of 30 frames per second (fps). For videos with a frame rate lower than 30 fps, we fill in the initial and final parts with the first and last frames, respectively. All the audios are sampled at 16kHz. ImageHear is a dataset that includes 100 images of 30 visual classes. We use this dataset for evaluation only using every single image as a 4-second video clip. Following previous work [15], we generate 120 audios for each class.

3.2. Setup

The VQ-VAE encoder has a total of 5 convolutional layers with stride 2, and the decoder has the reverse operation. After passing through two additional convolutional layers, the first codebook is used to apply an overall downsampling factor of 8. The second codebook is applied after passing through two more convolutional layers, resulting in an overall downsampling factor of 32. This corresponds to a token processing rate of 2000 per second for the Up model and 500 per second for the Low model. Each codebook is composed of 128 sizes, each containing 2048 codes. The auto-regressive models are based on Transformer architecture and sparse attention. Each model is 48 layers with a hidden size of 1024. The MHA module has one layer of 8 heads and 0.1 rates of dropout. We use the pre-trained VQ-VAE, Low, and Up-level language models in our experiment[2].

We use a batch size of 16 on two Tesla V100 GPUs. We search hyper-parameters $\lambda_{\mathbf{w}}$ over $[1, 0.1, 0.0001]$, $\lambda_{\hat{\mathbf{f}}}$ over

[2] https://github.com/RoySheffer/im2wav



(a) *ImageHear* (b) *VGG-Sound*

Figure 3: *Class-wise accuracies for the generated audio. Reference represents the results for real audio. The results of Im2Wav are re-implemented using their pre-trained model to get the exact class-wise accuracy number.*

Table 1: *Image-guided audio generation results for VGG-Sound (left) and ImageHear (right). Reference represents the result for real audio. Result of * from [15].*

Method	FAD↓	KL↓	CS↑	ACC↑	CS↑	ACC↑
Reference	-	-	8.79	58.02	-	-
[14]*	6.64	3.10	4.62	14.44	5.90	22.36
Im2Wav [15]*	6.41	2.53	7.19	35.77	9.53	49.14
GRAVO	**5.96**	**2.38**	**7.56**	**39.57**	**10.23**	**59.03**

Table 2: *Ablation study of GRAVO for VGG-Sound (left) and ImageHear (right).*

$\mathcal{L}_{\mathbf{w}}$	$\mathcal{L}_{\hat{\mathbf{f}}}$	FAD↓	KL↓	tCS↑	CS↑	ACC↑	CS↑	ACC↑
Im2Wav		6.41	2.53	-	7.19	35.77	9.53	49.14
✗	✗	6.06	2.39	11.04	7.41	39.53	9.81	53.06
✓	✗	6.10	2.40	**32.60**	7.43	**41.38**	9.86	56.28
✗	✓	6.37	**2.36**	9.62	7.31	39.36	10.09	55.83
✓	✓	**5.96**	2.38	9.67	**7.56**	39.57	**10.23**	**59.03**

$[1, 10, 1000]$, and γ over $[0.1, 0.5, 0.9]$. Then we set the hyper-parameter that shows effective performance on the validation set. We train the model once for each hyper-parameter and the number of parameters of Im2Wav and GRAVO is 361M and 362M, respectively.

3.3. Evaluation Functions

We evaluate the generated sounds on 4 metrics: Fréchet Audio Distance (FAD); Kullback–Leibler divergence (KL); clip-score (CS); audio classification accuracy (ACC, %).

FAD measures the distance between the generated and real distributions of audio and represents the fidelity of the audio generation. This distance is calculated by extracting features from both the real and generated audio using an audio classifier [21][3]. CS measures the relevance between images and the generated audio using pre-trained CLIP and Wav2CLIP models, respectively. We show the scaled results by multiplying by 100. KL and ACC are measured using PaSST model [22], which is an audio classifier of 527 classes. The KL divergence is computed on top of the classifier output. For ACC, we measure the score after replacing the softmax output of the audio classifier with zero for the general classes which are arranged by [15]. On VGG-Sound, we measure FAD, KL, and CS for all samples in the test set and ACC for 30 classes corresponding to the ImageHear classes.

In the ablation study of GRAVO, we measure the CS for the target audio and the transformed image features through the MHA, which we denote tCS.

4. Results

Table 1 shows the results on two datasets. We compare with SpecVQGAN [14] and Im2Wav [15] as our baselines. It can be seen that our model improves on all metrics by a large margin, especially for the ACC metric. The performance gap is significant on ImageHear. We guess it is because this dataset includes only a single clean object for each sample and GRAVO has learned to generate audio from soundable images well. Fig. 3 represents the class-wise accuracy of 30 classes in ImageHear. Results show that our method outperforms Im2Wav for most of the classes.

We further conduct an ablation study to show the effectiveness of each module. Table 2 shows the results. Firstly, we can see that even without guidance on MHA, GRAVO improves all the metrics over Im2Wav. In terms of tCS, all scores of GRAVO are higher than 8.79, which is a CS of Reference in Table 1. It implies that MHA transforms the CLIP embedding to the related features of the target audio more than the original image sequence. When using $\mathcal{L}_{\mathbf{w}}$ only, tCS and ACC improve while the other metrics keep the same. $\mathcal{L}_{\hat{\mathbf{f}}}$ increases CS and ACC on ImageHear and does not improve on VGG-Sound. Combining the two loss functions results in a better FAD and CS on VGG-Sound, while the other metrics are comparable with the baseline. On ImageHear, the combination leads to the best performance. It proves the proposed two auxiliary losses are very useful.

5. Conclusions

In this paper, we present GRAVO, which generates audio that is relevant to images. To achieve this, the model incorporates an MHA module on top of pre-trained CLIP features to learn the intrinsic relationships between images. We also propose to use Wav2CLIP embeddings to guide the behavior of the MHA module, allowing it to learn features that are relevant to audio in images. Experimental results demonstrate that GRAVO significantly enhances generation quality across multiple metrics.

6. Acknowledgements

This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program) (2022-00155958) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) and by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)).

[3] https://github.com/google-research/google-research/tree/master/frechet_audio_distance

7. References

[1] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[2] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[3] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[4] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

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

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

[7] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[8] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[9] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "Musiclm: Generating music from text," 2023. [Online]. Available: https://arxiv.org/abs/2301.11325

[10] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[11] K. Chen, C. Zhang, C. Fang, Z. Wang, T. Bui, and R. Nevatia, "Visually indicated sound generation by perceptually optimized classification," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 560–574.

[12] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg, "Visual to sound: Generating natural sound for videos in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3550–3558.

[13] P. Chen, Y. Zhang, M. Tan, H. Xiao, D. Huang, and C. Gan, "Generating visually aligned sound from videos," *IEEE Transactions on Image Processing*, vol. 29, pp. 8292–8302, 2020.

[14] V. Iashin and E. Rahtu, "Taming visually guided sound generation," *arXiv preprint arXiv:2110.08791*, 2021.

[15] R. Sheffer and Y. Adi, "I hear your true colors: Image guided audio generation," *arXiv preprint arXiv:2211.03089*, 2022.

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

[17] H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello, "Wav2clip: Learning robust audio representations from clip," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4563–4567.

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

[19] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[20] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.

[21] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[22] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *arXiv preprint arXiv:2110.05069*, 2021.