Towards an End-to-End Visual-to-Raw-Audio Generation with GAN

Shiguang Liu, *Senior Member, IEEE*, Sijia Li, Haonan Cheng

Abstract—Automatically synthesizing sounds for different visual contents poses a challenge and there is a strong need to facilitate the direct creation of realistic sounds. Different from previous works, in this paper, we propose a novel deep learning based approach, which formulates sound simulation as a regression problem. This allows us to circumvent the complexity of the acoustic theory by a novel, general-purpose neural sound synthesis (V2RA) network. Moreover, the end-to-end architecture of V2RA ensures full training without any extra inputs, which thereby greatly improves the scalability and reusability over previous works. In contrast to conventional visual-to-audio generation methods, the V2RA problem is first established and solved by generative adversarial networks (GANs). Furthermore, our network architecture can directly predict synchronized raw audio signals (unlike most existing approaches that handle the audio through spectrograms) and generate sound in real time. To evaluate the performance of the neural network generator, we specifically introduce two quantitative scores. Various experiments demonstrate that our V2RA network can produce compelling sound results, which thus provides a viable solution for applications such as sound design and dubbing.

Index Terms—Visual to audio, cross media, GAN, audio-visual synchronization.

1 Introduction

Despite over two decades of research in sound synthesis for visual content [1], [2], [3], automatically synthesizing matching sounds remains a demanding task in computer graphics. One reason may be the diversity and complexity of the acoustic theory. Different from current physics-based sound synthesis methods, we formulate automatic sound synthesis as a regression problem from visual signals to audio signals by leveraging advanced deep learning techniques. We propose an automatic visual-to-raw-audio (V2RA) system that can circumvent complex acoustic theory and directly predict raw audio from soundless videos in real time. There are many practical applications, such as automatic Foley processing, image sonification for visually impaired persons and several other joint audio-video processing. Moreover, it requires little expertise to use and reduces theoretical complexity significantly. Lastly, it is an automatic procedure that can help foley artists reduce tedious editing work.

Recently, physically-based modeling of sound has gained increasing attention in computer graphics. Current methods usually build on visual simulation with physical parameters as the input for sound modeling. Advances in this field have enabled successful synthesis of a wide variety of sounds, for example, liquid sound [1], [4], [5], fire sound [6], and impact sound [2]. Generally, there are two main issues for these tasks: i) how to achieve synchronization between visual and audio, and ii) how to improve the timbre of the synthetic audio. Since these physics-based sound simulation methods are generally driven by physical parameters from the animation, the synchronization problem is relatively tractable. However, the acoustic theories that these methods depend on for sound modeling are usually complex and specialized for a single type of sound. To generate realistic sound results, some techniques explore different solutions such as multi-scale simulation [7], parameter estimation [2], complex acoustic bubbles [5], and statistical simulation [8]. Each of these tasks focused on one particular kind of sound, which limits their scalability and reusability, although the goal remains the same, i.e., generating audio from visual content. Moreover, it is difficult for physics-based sound simulation to generate ideal sound for types of animations or movies which suffer from the problem of indirect control. For example, two-dimensional animations, such as keyframe-based (e.g., car movement) or behavioral-based animation (e.g., animal crying), were discussed in [9] that usually lack three-dimensional models.

Granular synthesis methods are also common in sound synthesis technology [10], [11]. The input sound is sampled on a small time scale and split into small grains around 1 to 50 ms. The grains are synthesized in different ways to generate various sounds. However, these methods need to set many parameters manually, which have a significant impact on the generated sounds. Moreover, the generated sound needs to match the visual information. To overcome these limitations, we attempt to bypass the modeling of complex and specialized acoustic equations. Radically different from previous works, we no longer pay attention to separate, special-purpose machinery. Instead, we formulate these simulation tasks as a regression problem. That is, we translate one representation (visual content) into another (raw audio). This also makes it possible to investigate a general-purpose sound synthesis method for different types of animations (with or without a three-dimensional model) and videos. However, the problem of visual-audio regression is rather

- *Shiguang Liu, Sijia Li, and Haonan Cheng are with School of Computer Science and Technology, Division of Intelligence and Computing, Tianjin University, Tianjin 300350, P.R. China. (e-mail: lsg@tju.edu.cn, lisj@tju.edu.cn, tjuchn@tju.edu.cn).*
- *Shiguang Liu is also with Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300350, P.R. China. He is the corresponding author.*

difficult because the video and audio used for the generation have different temporal and spatial scales. High quality audio synthesis occurs at a sampling rate of 44.1 KHz, about 1000 times more frequently than video (Figure 1(a)). To build an end-to-end V2RA generation system, we need to bridge this gap. As shown in Figure 1(b), the principal components of video frames include characteristics such as gradient and boundary that are unusual in audio structure. Therefore, it is challenging to learn the rules of synchronization through mismatched features.

Fortunately, some deep learning based techniques [12], [13], [14] have been proposed to study whether computational models can learn the relationship between visual and audio. These techniques endow us with a preliminary understanding of the deep visual and audio features. However, most of these attempts generated different representations of the sound through an image-to-image translation which is different from our raw audio simulation task. For example, a Long Short-Term Memory (LSTM) based method [12] was developed for hitting and scratching sound synthesis with cochleagram as the representation of the sound. Similarly, a Conditional Generative Adversarial Net (CGAN) based method [13] was proposed to synthesize sounds of the instrument with Log-amplitude of Mel-Spectrum (LMS) for sound representation. Due to the vast differences between visual features and audio features, instead of directly outputting raw audio, these methods generate a time-frequency representation (e.g, cochleagram and LMS) which thereby cannot be directly used for our automatic sound synthesis task. Zhou et al. [14] directly built connections between visual contents and raw audio. However, they use SampleRNN [15] as the sound generator, which is autoregressive and inherently slow during inference.

To overcome the above limitations, we propose an end-to-end network framework that contains a novel video encoding process and a V2RA Generative Adversarial Network (V2RA-GAN). The video encoding process is proposed to bridge the gap of spatial and temporal scales, and the V2RA-GAN can learn the inherent difference and generate matching sound in real time. More precisely, we encode the visual information with $fc6$ feature of VGG19 network [16]. Then we select features at equal intervals and connect them to a 44100-dimensional vector with the same length of audio vectors. The V2RA-GAN is designed for audio generation with several architectural adjustments. To train and test the proposed architecture, we further clean and extend existing datasets [12], [14] to build a more suitable dataset, containing hundreds of videos for different natural scenes such as videos with dogs and fireworks. Furthermore, we introduce two quantitative evaluation scores to assess the audio quality and synchronization rate. Various tests and comparisons show that the proposed network architecture and training procedure enable high quality V2RA generation. In summary, we make the following contributions:

1. We propose an end-to-end, general-purpose system for the automatic, synchronization-aware sound synthesis task, which can generate a wide variety of sounds. The proposed technique can successfully bridge the two large gaps between video and audio and produce convincing results.

2. We propose a video encoding process, which is fully trainable without any extra inputs and therefore greatly improves the scalability, reusability, and generation speed.

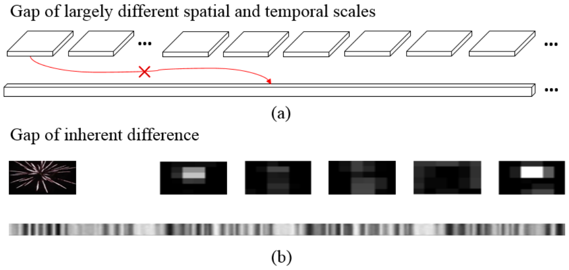

Gap of largely different spatial and temporal scales
(a)
Gap of inherent difference
(b)

Fig. 1. Two challenges of end-to-end V2RA generation. (a) shows the scale structure of video frames (top) and the matching audio (bottom) in the same time interval. (b) shows a frame of video and visualizations of the top 5 features of the VGG19 network (top) and the visualization of the sound corresponding to the video frame (bottom). The gray value of the pixels corresponds to the amplitude of the waveform.

3. As far as we know, our quantitative scores are the first objective measurements for the visual to audio generation task, which can overcome the limitations of previous studies that rely solely on psychological studies.

2 Related Work

In this section, we mainly review existing techniques that are highly related to our work. We classify the related works into two main categories: physics-based techniques and learning-based techniques.

2.1 Physics-based techniques

Traditional sound effect production is a laborious practice for computer animation and movies. To achieve automatic sound generation, several physically based methods are proposed that are synchronized with visual animation. Physics-based sound synthesis has led to improved sound-generation techniques for computer-animated phenomena, including water [1], [4], [5], wind [17], fire [6], [18], [19], [20] and rigid bodies [21], [22], [23], [24]. The general procedure of these methods includes simulating animation models and acoustic models, extracting suitable parameters and rendering both animation and audio. The advantage of the physical methods is flexibility, because sound synthesis and animation production can be adjusted by physical parameters. Moreover, since most of these methods are based on example recordings or acoustic theory, the quality of the synthesis results can usually reach a satisfactory level.

However, there is still an open problem of physics-based approaches for generic objects. To establish connections between video and audio using physics-based methods, one needs to simulate animation models and acoustic models, extract suitable parameters and render both video and audio. In contrast, the learning-based methods are more generic and attempt to explore whether computational models can learn the association between visual and audio directly just like human perception. Our work also adopts a learning-based way to train a model that can automatically and directly generate different types of audio conditioned by input video frames.

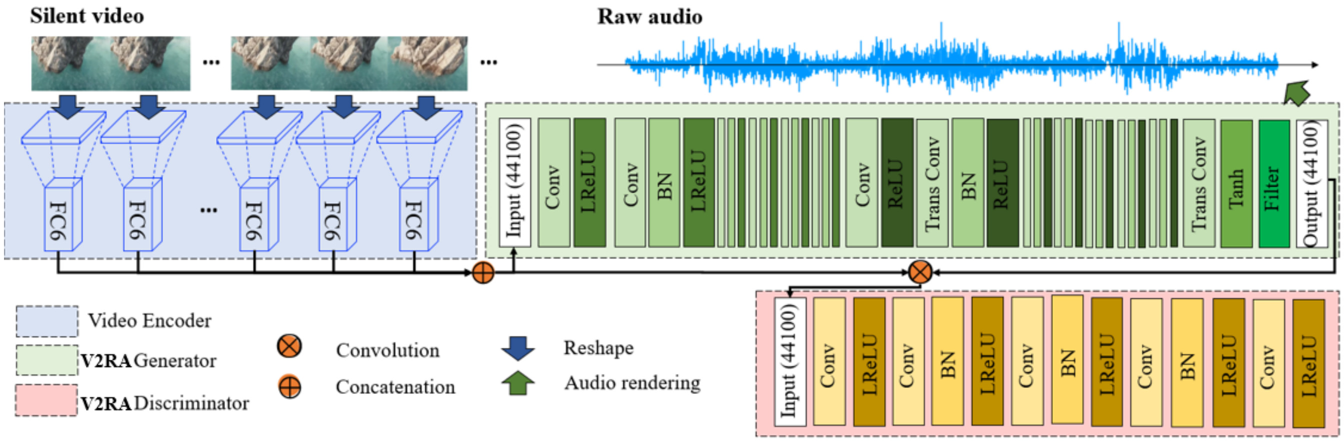


Fig. 2. The framework of our method. We train an end-to-end neural network to map video sequences to raw audio. The network contains three parts: video encoder, V2RA generator and V2RA discriminator. We encode the video frames into 4096-dimensional vectors and concatenate them to 44100-dimensional vectors. An adjusted conditional GAN for high-dimensional input is used for V2RA transformation.

2.2 Learning-based techniques

There is a lot of research devoted to exploring the relationship between visual and audio [25], [26], [27], [28], [29], [30]. Recently, some researchers [12], [13] have begun to explore the visual-to-audio generation task. Owens et al. [12] proposed a neural network which consists of a Convolutional Neural Network (CNN) and a long short-term memory unit (LSTM) for synthesizing hitting sound. Chen et al. [13] developed a generation model between image and sound.

The algorithms in [12] and [13] are based on cochleagram and LMS, respectively. Both of the above learning-based approaches treat audio as image-like spectrograms and therefore cannot directly generate raw audio signals. In essence, these methods are still transforming an image into another image. Building an end-to-end V2RA generation system would be more direct but also more challenging, because audio and video have mismatched temporal scale, different spatial dimensions, and distinct features. To implement a more direct way, Zhou et al. [14] generated natural sound for videos collected in the wild through a SampleRNN model [15]. However, it is inherently slow during inference because SampleRNN is an autoregressive model. To address the aforementioned limitations, we propose a novel solution to synthesize audio directly, which produces convincing results with superior efficiency.

CNNs provide a promising way for identifying the action in videos and have become the common workhorse for a wide variety of tasks [31], [32], [33], [34]. An open problem of CNNs is the requirement of expert knowledge to design effective losses. Fortunately, Generative Adversarial Networks (GANs) [35] provided a discriminator to automatically learn a loss function appropriate for reaching a goal. In recent years, GANs have achieved a good level of success to generate high-dimensional signals and some works [36], [37] have proved the feasibility of using GANs to synthesize sound. Moreover, Donahue et al. [36] demonstrated that GAN is capable of synthesizing audio in an unsupervised setting and it is faster than the autoregressive mode. However, we need both video frames and audio as input for training, so the traditional GANs do not apply to our task. Therefore, in this paper, we explore GANs in the conditional setting and concatenate video information as the condition to control the change of audio. Conditional GANs [38] have been vigorously studied in the last two years and many techniques for translation tasks have been previously proposed. For example, prior and concurrent works have used conditioned GANs on discrete labels [38], [39], text [40], music [41] and images [42]. However, the V2RA generation task is quite different from the works mentioned above in Section 1. We need to adjust the encoding format and design the appropriate filter and loss function, which we will discuss in detail in Section 3. As the first attempt to present a framework with GANs in the conditional setting for an end-to-end V2RA generation task, our work suggests that the V2RA-GAN is capable of synthesizing convincing audio in a supervised setting.

3 V2RA GENERATION

In our work, the V2RA generation task is formulated as a regression problem to map a sequence of video frames to a sequence of raw audio waveform. We solve the problem with three main ingredients, namely a video encoder (the blue part in Figure 2), a V2RA generator (the green part in Figure 2) and a V2RA discriminator (the red part in Figure 2). A silent video is fed into the video encoder to bridge the spatial and temporal gaps. Then, the video features (fc6 features) are generated automatically and the generated video features are concatenated to form a 44100-dimensional vector as the input of the V2RA generator. Additionally, the long vector is convolved with the output of the generator as the input of the V2RA discriminator to determine the quality of the generator output. The V2RA generator and V2RA discriminator together constitute the V2RA-GAN, which aims to transform the video features into raw audio signals.

We discuss our video encoding process in Section 3.1. Then, in Section 3.2, we describe the V2RA-GAN architecture. Finally, in Section 3.3, we put forward two audio optimization schemes to adjust the audio results since the

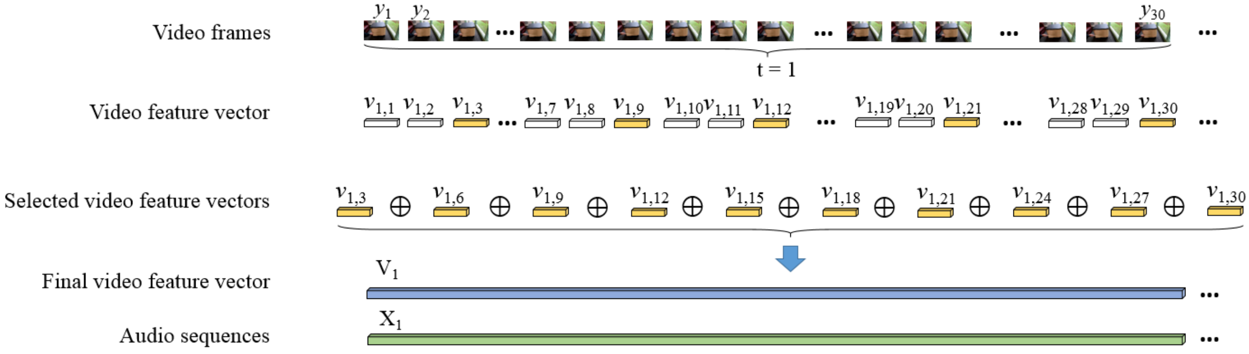

Video frames
y_1 y_2 y_{30}
t = 1
Video feature vector
$v_{1,1}$ $v_{1,2}$ $v_{1,3}$ $v_{1,7}$ $v_{1,8}$ $v_{1,9}$ $v_{1,10}$ $v_{1,11}$ $v_{1,12}$ $v_{1,19}$ $v_{1,20}$ $v_{1,21}$ $v_{1,28}$ $v_{1,29}$ $v_{1,30}$
Selected video feature vectors
$v_{1,3}$ ⊕ $v_{1,6}$ ⊕ $v_{1,9}$ ⊕ $v_{1,12}$ ⊕ $v_{1,15}$ ⊕ $v_{1,18}$ ⊕ $v_{1,21}$ ⊕ $v_{1,24}$ ⊕ $v_{1,27}$ ⊕ $v_{1,30}$
Final video feature vector
V_1
Audio sequences
X_1

Fig. 3. The process of video encoding. Take $SR_{video} = 30$, $SR_{audio} = 44100$ and $\Delta t = 1$ as an example. We concatenate the selected video feature vectors $\{v_{1,q}, v_{1,2q}, ..., v_{1,p\cdot q}\}$ (yellow blocks) to generate the final video feature vector V_1 (blue block) and segment the entire audio sequence into audio vectors X_1 (green block) with equal lengths.

audio channel of the training video usually has some noise-like background sound.

3.1 Video encoder

As the first step of the V2RA generation, the video encoder is proposed to bridge the spatial and temporal gaps between video and audio. As shown in Figure 3, we specifically design a new video encoding method to generate video features that have matching lengths with audio sequences. This makes it easier for networks to learn mapping relationships between video and audio. To design the video encoder for our task, we first formulate the V2RA training problem as:

$$G(y_1, ..., y_m) \rightarrow x_1, ..., x_n, x \in X, y \in Y, \quad (1)$$

where X represents the audio set, Y represents the frame set, $y_1, ..., y_m$ (ranging from 0 to 255) represents input video frames, $G(y_1, ..., y_m)$ represents output raw audio values (ranging from -1 to 1) and $x_1, ..., x_n$ denotes real raw audio values (ranging from -1 to 1). An ideal V2RA model is to meet the following goal:

$$min \left[Dis(G\left(y_1, ..., y_m\right), (x_1, ..., x_n))\right]. \quad (2)$$

Here, $Dis(\cdot)$ is a distance function to calculate the similarity between audio sequences which we will introduce in Section 4.3.

The collected videos usually have different resolutions, for unified processing, we reshape the video frames into the scale of $256 \times 256 \times 3$. However, the scale of $1s$ audio is usually $1 \times 44100 \times 1$. To bridge the spatial gap, we first encode the video frames to have the same length as audio sequences. For each video frame y_i, we construct the video feature vector v_i through a VGG19 network [16] and v_i is transformed to a $1 \times 4096 \times 1$ vector. Then we need to bridge the temporal gap. In the same time interval Δt, $m = \Delta t \cdot SR_{video}$ and $n = \Delta t \cdot SR_{audio}$, where SR_{video} and SR_{audio} are the sampling rates of video and audio, respectively. Thus, $n \gg m$ due to different sampling rates of video and audio. To make sure the final video feature vector V has the correct temporal length, for each time interval Δt, we concatenate p video feature vectors of q neighbor frames (yellow blocks in Figure 3) as follows:

$$V_t = v_{t,q} \bigoplus v_{t,2q} \bigoplus ... \bigoplus v_{t,p\cdot q}. \quad (3)$$

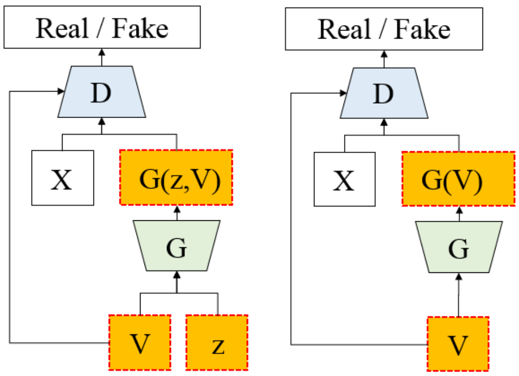


Fig. 4. Training a conditional GAN to map a video to audio. Unlike previous conditional GANs (left) with Gaussian noise z (red dotted box), we use V as both the input and the condition (right).

Here, $\bigoplus$ represents the concatenation operator, $p = Floor[SR_{audio}/\ 4096]$ is the number of video feature vectors and $q = Floor[SR_{video}\ /p]$ represents the interval of video feature vectors. For the missing parts, we pad zeros in equal intervals to ensure a complete match of sequence lengths. Thus, the V2RA problem can be reformulated as:

$$G\left(V_1, V_2, ..., V_{\Delta t}\right) \rightarrow X_1, X_2, ..., X_{\Delta t}, \quad (4)$$

where $X_t = \{x_{t,1}, x_{t,2}, ..., x_{t,SR_{audio}}\}, t \in \{1, 2, ..., \Delta t\}$. Therefore, V_t and X_t have the same length as shown in Figure 3 and the video frames are encoded in the same spatial and temporal intervals as the audio.

3.2 V2RA-GAN architecture

After we generate the video features $V_t (t \in \{1, 2, ..., \Delta t\})$ and audio vectors $X_t (t \in \{1, 2, ..., \Delta t\})$, a V2RA-GAN is proposed to transform the video features to audio vectors. Different from GANs that learn a generative model of data, to solve the V2RA generation problem, we take V_t as the condition to force the output audio vectors conditioned on input videos. Thus, the V2RA-GAN is in the conditional setting and contains both V_t and X_t as input in the training process. Similarly with other types of GANs, the V2RA-GAN also contains two parts, namely a V2RA generator and a V2RA discriminator. The V2RA generator G of V2RA-GAN learns a mapping from video feature vectors V_t to audio sequences X_t, i.e., $G\left(V_1, V_2, ..., V_{\Delta t}\right) \rightarrow X_1, X_2, ..., X_{\Delta t}$. And the V2RA discriminator D is trained to determine if $G(V_t)$ is real or fake under the condition V_t. Since the final output is a set of raw audio signals, our V2RA-GAN differs from the architectures of prior works which were popularized for image synthesis.

Large receptive fields. Different from image synthesis, high quality audio synthesis occurs at a sampling rate of 44.1 KHz which is much higher than that of a video clip. Therefore, it is required to adapt the V2RA generator and V2RA discriminator according to the characteristics of audio. The input of the V2RA generator is V_t and the input of the V2RA discriminator is $V_t \bigotimes G(V_t)$, both of whose lengths are 44100. This suggests that filters with larger receptive fields are needed to process the vector. Therefore, we modify the convolution operation and transposed convolution operation of the conditional generator to widen its receptive field. Specifically, we use longer one-dimensional filters of different lengths instead of two-dimensional filters. The reason for choosing the filters of variable length is to make the output vector shape an integer after the convolution operation and transposed convolution operation. The details of the filter length can be found in Table 1 and Table 2.

Audio filter. Since the frequency of the sampled environmental sound has a wide distribution, there exists undesirable frequency information in the output results. Inspired by [36], we add a filter with a long window (513 samples) to filter out undesired frequencies in the last layer of the generator. The length 513 (see Table 1) is chosen to keep the length of the output audio sequence unchanged. We tested the effect of the audio filter as described in Section 4.5 and the results show that adding the audio filter greatly improved the audio quality.

Loss function. Typically, the loss function of a traditional conditional GAN [38] is the cross-entropy loss function which can be expressed as:

$$\begin{aligned}\min_G \max_D \mathcal{L}(D,G) = &\mathbb{E}_{X,V\sim p_{data}(X,V)}[log(D(X,V))] \\ &+\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),V\sim p_{data}(V)}[log(1-D(G(\boldsymbol{z},V),V))]\end{aligned}, \tag{5}$$

where D is trained to maximize the loss and G is trained to minimize the objective. As the conditional information, V is fed into both the discriminator and the generator as an additional input layer. To improve the training stabilization and the quality of the generated samples in G, we adjust the loss function. First, we use the least-squares (LS) loss function to replace the cross-entropy loss function, since LS loss can avoid the problem of vanishing gradients as proven in [43]. Although it is only proved in the image translation domain, according to the comparison results in Section 4.5, the LS loss function in V2RA generation also has a better effect for audio. Therefore, the formulation in Equation (5) changes to

$$\begin{aligned}\min_D \mathcal{L}(D) = &\frac{1}{2}\mathbb{E}_{X,V\sim p_{data}(X,V)}[(D(X,V)-1)^2] \\ &+\frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),V\sim p_{data}(V)}[(D(G(\boldsymbol{z},V),V))^2]\end{aligned}, \tag{6}$$

$$\begin{aligned}\min_G \mathcal{L}(G) = &\frac{1}{2}\mathbb{E}_{\vec{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),V\sim p_{data}(V)} \cdot \\ &[(D(G(\boldsymbol{z},V),V)-1)^2]\end{aligned}. \tag{7}$$

Our task is formulated as a regression problem. Therefore, we chose common regression loss to constraint our training procedure. Specifically, we chose the $L1$ loss to minimize the distance between the generated results and ground truth because the $L1$ loss is more robust to exception values. Our training videos are mostly in the wild and there is much noise in the videos. Therefore, we add the $L1$ loss for its robustness to the generator to generate more realistic results. Adding such a term has also been proven to be effective in the image translation domain [42]. We have also experimented on the effects of adding the L_1 loss function and found that $L_1 + LS$ achieves the best performance. The magnitude of the L_1 norm is controlled by a new hyperparameter λ. Finally, contrary to standard GAN formulations, we only provide noise by the use of dropout and no longer use $\boldsymbol{z}$ as input as shown in Figure 4 (yellow part). This is because the generator is conditioned on the input video features V, which can provide variability of the generator even without $\boldsymbol{z}$. Moreover, this design has been successful in image translation processes [42] and we prove that it is also feasible for the audio generation task in our experiments. Considering all of the above observations, our final loss function is designed as follows:

$$\begin{aligned}\min_D \mathcal{L}(D) = &\frac{1}{2}\mathbb{E}_{X,V\sim p_{data}(X,V)}[(D(X,V)-1)^2] \\ &+\frac{1}{2}\mathbb{E}_{V\sim p_{data}(V)}[(D(G(V),V))^2]\end{aligned}, \tag{8}$$

$$\begin{aligned}\min_G \mathcal{L}(G) = &\frac{1}{2}\mathbb{E}_{V\sim p_{data}(V)}[(D(G(V),V)-1)^2] \\ &+\lambda\mathbb{E}_{X,V\sim p_{data}(X,V)}[\|X-G(V)\|_1]\end{aligned}. \tag{9}$$

In Table 1 and Table 2, we list the full architectures for our V2RA-GAN generator and discriminator, respectively. We use convolution layers with leaky ReLU activations and strided convolution layers with ReLU activations. BatchNorm is not applied to the first convolution layer in both the generator and the discriminator.

3.3 Audio optimization schemes

The audio results generated by V2RA-GAN may have an undesirable timbre. Here, we design two audio optimization schemes that can help improve the audio quality.

Audio de-noising The first optimization scheme is a de-noising algorithm inspired by [44]. To generate clear audio, we first reformulate the audio signal $G[n]$ synthesized from the V2RA-GAN into two parts as:

$$G[n] = f[n] + \epsilon[n], \tag{10}$$

where $f[n]$ denotes the clear signal and $\epsilon[n]$ denotes noise. More technical details of the solution can be found in [44].

Various experiments showed that the above procedure is robust to signal structures with low frequency (e.g., the bird category in our dataset as shown in Figure 5(a)), which can not achieve good performance in categories such as dog or wood. Because the noise and the high-frequency information of sounds in these categories are mixed together (as shown in Figure 5(b)), the de-noising strategy alone is not suitable for them.

Peak-match For sounds like a dog barking, the noise and the high-frequency information of sounds are mixed.

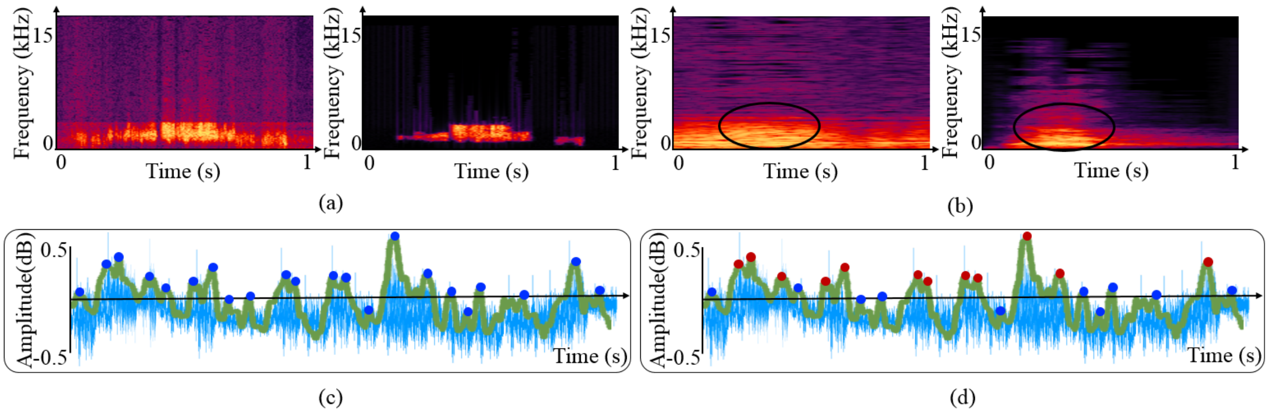


Fig. 5. Spectrum comparison and peak comparison. (a) shows the spectra of a bird's chirping with (left) and without noise (right). (b) shows the spectra of a dog's barking with (left) and without noise (right). (c) shows all the searched peaks and (d) shows the selected peaks (red dots).

TABLE 1
The generator architecture of V2RA-GAN.

Operation	Kernel Size	Output Shape
Input V Uniform(-1,1)		(n,44100)
Conv1D (Stride=3)	(9,c,d)	(n,14700,d)
Conv1D (Stride=3)	(9,d,2d)	(n,4900,2d)
Conv1D (Stride=4)	(12,2d,4d)	(n,1225,4d)
Conv1D (Stride=5)	(15,4d,8d)	(n,245,8d)
Conv1D (Stride=5)	(15,8d,8d)	(n,49,8d)
Conv1D (Stride=7)	(21,8d,8d)	(n,7,8d)
Conv1D (Stride=7)	(21,8d,8d)	(n,1,8d)
Trans Conv1D (Stride=7)	(21,8d,8d)	(n,7,8d)
Trans Conv1D (Stride=5)	(15,8d,8d)	(n,49,8d)
Trans Conv1D (Stride=5)	(15,8d,8d)	(n,245,8d)
Trans Conv1D (Stride=4)	(12,8d,4d)	(n,1225,4d)
Trans Conv1D (Stride=3)	(9,4d,2d)	(n,4900,2d)
Trans Conv1D (Stride=3)	(9,2d,d)	(n,14700,d)
Trans Conv1D (Stride=3)	(9,d,c)	(n,44100,c)
Tanh		(n,44100,c)
Conv1D (Stride=1)	(513,c,c)	(n,44100,c)

TABLE 2
The discriminator architecture of V2RA-GAN.

Operation	Kernel Size	Output Shape
Input X and $G(V)$		(n,44100,2c)
Conv1D (Stride=4)	(16,2c,d)	(n,11025,d)
Conv1D (Stride=4)	(16,d,2d)	(n,2756,2d)
Conv1D (Stride=4)	(16,2d,4d)	(n,689,4d)
Conv1D (Stride=1)	(4,4d,8d)	(n,688,8d)
Conv1D (Stride=1)	(4,8d,8d)	(n,687,8d)

Algorithm 1 Peak-match.

```
Input:
    Audio sequence $G_V$ and reference audio $Ref$;
Output:
    The optimized sound $G_O$;
$len \leftarrow length(G_V)$;
$[GL_{peak}, GV_{peak}] \leftarrow findpeaks(G_V)$;
$[RefL_{peak}, RefV_{peak}] \leftarrow findpeaks(Ref)$;
Delete $GV_{peak} < 0.1$ and $RefV_{peak} < 0.1$
Initialize $G_O(1 : len)$ with zero;
$N_{peak} \leftarrow length(GV_{peak})$;
for $i = 1 : N_{peak}$ do
    StartPoint $\leftarrow RefL_{peak}(i)$;
    EndPoint $\leftarrow RefL_{peak}(i) + interval$;
    $G_O(GL_{peak}(i) : GL_{peak}(i)+interval) \leftarrow GV_{peak}(i) \times Ref(StartPoint : EndPoint)$;
end for
Normalize $G_O$;
```

Therefore, to improve the audio quality of these categories, we further propose a peak-match method.

The main idea of the peak-match process is to search for and match the effective sound part (the circled part in Figure 5(b)). Owing to the synchronization, that is, the change of the audio has already been accurately obtained, we can search the peak of the waveform and match the corresponding peak from recordings. The recordings we utilized are from the Adobe Audition Sound Effect [1]. The process is summarized in Algorithm 1.

4 Experiments

In this section, we first introduce the datasets in Section 4.1 and the training details in Section 4.2. Next, we propose two quantitative indicators for audio quality and evaluate the results based on both indicators in Section 4.3. Then, we compare our technique with state-of-the-art methods in Section 4.4. We further discuss the influence of network

[1] https://offers.adobe.com/en/na/audition/offers/audition_dlc/AdobeAuditionDLCSFX.html?cq_ck=1407955238126&wcmmode=disabled

architecture adjustments on performance and audio optimization effects in Section 4.5 and Section 4.6, respectively. Finally, we compare the results from a user study in Section 4.7.

4.1 Dataset

To explore the generality of our proposed V2RA networks, we tested the method on a variety of tasks and categories, including animated videos and videos in the wild. The information of the testing datasets [12], [14] is listed as follows:

- *VEGAS* [14][2]: This dataset contains 28109 videos that include both video and audio channels. There are 10 categories in the dataset and the total length is 55 hours.
- *Greatest Hits dataset* [12][3]: This dataset contains 977 videos (including both video and audio channels) of human probing (e.g., hitting) objects with a drumstick.

We chose videos of fireworks and dogs from *VEGAS* and impact sounds of cloth, wood, gravel and plastic from the *Greatest Hits dataset*. Besides, we collected bird videos from a free HD stock video website[4]. We randomly chose clean videos (with less background noise) from each category for training and testing. For example, we filtered out videos with background music or videos with voice overs. We chose these categories because they are sensitive to synchronicity. Furthermore, the match between video and audio is easier to observe in these categories. Finally, we chose about 10-30 videos for each categories from the exisiting datasets. For the bird videos, we only get limited videos due to the difficulty of collection and collected about 360s in total. For each task, we split each dataset into the training and testing sets (75% training and 25% testing). During training, we utilized *VEGAS*, *Greatest Hits dataset* and bird videos for V2RA model training. During testing, we tested not only wild videos in the test set, but also animated videos from previous research [2].

4.2 Training details

The proposed model is trained on each category independently. The difficulty increases sharply if we train our model on mixed categories. The model needs to classify the video first and then generate the corresponding audio. This means the more complex correspondence between visual and audio the higher requirements for the generator. The complex problem will result in lower quality sound results. Therefore, we train a separate model for each category. We sample the videos at 30 FPS (300 frames for 10 seconds) and sample the audio at 44.1kHz, i.e., 441000 times per 10 seconds. We use 44.1kHz audio to preserve richer details. We randomly select 25% of the videos from each category for testing, leaving the remaining videos for training with no overlap between paired data. The audio files used for training are the audio that comes with the video in the dataset. The audio files utilized for peak-match are randomly selected from the Adobe Audition Sound Effect dataset. During training, we apply the Adam Stochastic Optimization [45] with a learning rate of 0.001 and a dropout rate of 50%. In Table 3, we list the value of hyperparameters we used in our experiment. All the experiments are trained on a single NVIDIA Quadro 5000 GPU. At test time, we can generate a second audio in 0.03s on this GPU.

TABLE 3
V2RA-GAN hyperparameters and their values.

Name	Value
Num channels (c)	1
Batch size (n)	64
Model size (d)	64
Loss	LS
D updates per G updates	2
Optimizer	Adam

TABLE 4
ODG value and subjective impairment scale.

Different Grades	Description of Impairments
0	Imperceptible
-1	Perceptible but not annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

4.3 Evaluating the V2RA generation quality

How to quantitatively evaluate the performance of a neural network generator for audio has always been a challenging topic. Donahue et al. [36] used the inception score [46] for quantitative measures. However, they pointed out that the inception score was not an accurate proxy for human perception. Therefore, we explore a quantitative measurement based on the study of perception and psychology [47], [48] and put forward two quantitative indicators in terms of audio quality and audio similarity.

TABLE 5
Quantitative evaluation.

Audio	OSG
Figure 6(a) middle & Figure 6(b) middle	9.0417
Figure 6(a) bottom & Figure 6(b) bottom	**3.8867**
Ground truth & [2]	33.0061
Ground truth & Ours	**31.7959**

[2] http://bvision11.cs.unc.edu/bigpen/yipin/visual2sound_webpage/visual2sound.html

[3] http://andrewowens.com/vis/

[4] http://www.videezy.com

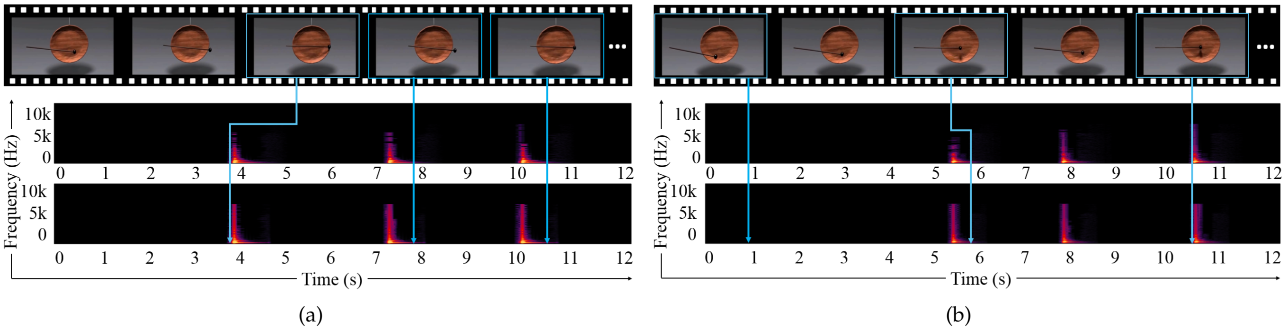


Fig. 6. Comparisons with physics-based methods for two wood scenarios (a) and (b). In each group, the top row is the input video, the middle row and the bottom row show the results of [2] and ours, respectively.

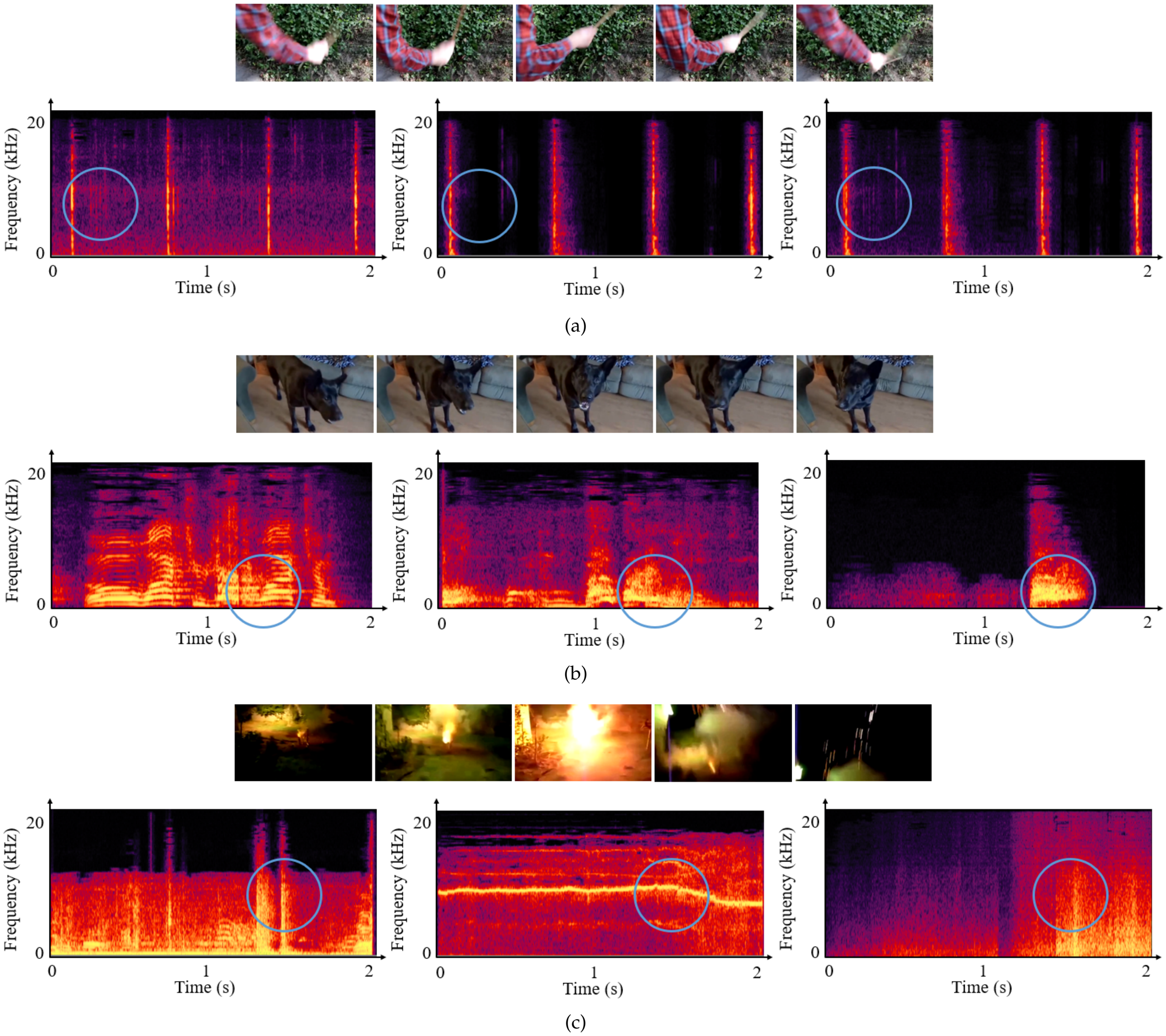


Fig. 7. Comparisons of the spectra of sound generated by different learning-based methods. (a) is the comparison of the ground truth, the sound result of [12] and the sound result of our method. (b) and (c) are the comparisons of the ground truth, the sound result of [14] and the sound result of our method. In each sub-figure, the top row is the video sequence and the bottom row is the spectra of different results (from left to right are the ground truth, compared result, and our result). All the results were normalized.

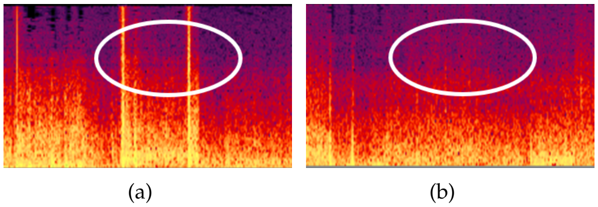


Fig. 8. The effect of filter layers (in the fireworks category). (a) is the result with filter layers. (b) is the result without filter layers. We can observe that the result with filter layers is clearer.

4.3.1 *Quality assessment*

Sound quality is typically an assessment of the accuracy, enjoyability, or intelligibility of audio output from an electronic device. By comparing subjective tests and Perceptual Evaluation of Audio Quality (PEAQ) values in [48], we adapt the PEAQ algorithm for audio quality assessment which is used for voice over Internet Protocol. The quality assessment algorithm contains a model that outputs variables combined with a trained neural network to give a single metric. This metric, namely objective difference grade (ODG), can measure the degradation of a test input relative to a reference input. Specifically, the neural network has been trained to give good matches to the subjective impairment scale as shown in Table 4.

$$ODG = b_{min} + (b_{max} - b_{min})sig(D_I), \tag{11}$$

where $b_{min} = -3.98$, $b_{max} = 0.22$, $sig(.)$ is an asymmetric sigmoid and D_I is a distortion index. More calculation details can be found in [49], [50].

4.3.2 *Similarity assessment*

To evaluate the similarity between the ground truth and synthesized audio, we propose the objective similarity grade (OSG) which contains three acoustic features. The choice of acoustic features is inspired by a psychological research [47], that investigated the acoustical correlates of similarity and categorization judgments of environmental sounds. Thus, we combine three acoustic features that have the strongest correlation with the correlation coefficient r. The correlation coefficient can be found in [47]. The final OSG is a combination of three parts: maximum modulation spectrum (MMS), mean spectral flux (MSF) and root mean square (RMS). In particular,

$$OSG = r_1 \cdot MMS + r_2 \cdot MSF + r_3 \cdot RMS, \tag{12}$$

where $r_1 = 0.57$, $r_2 = 0.52$ and $r_3 = 0.45$. The smaller the OSG value, the more similar the two audio clips are, and the OSG value of two of the same audio clips is 0.

4.3.3 *Nearest Neighbor Retrieval*

We expect that our results have significant differences for different categories. Therefore, we conducted a nearest neighbor retrieval experiment as a baseline to show that our model can learn good correspondence between visual and audio. The audios in the test set make up the retrieval dataset and we use visual features to retrieve corresponding audio in the dataset. In this section, we consider the question: ”Is the retrieved audio in the same category with the video?” The average classification accuracy is 15.71% that is better than chance (14.29%), which also shows that our model is able to learn reasonable correspondence between visual and audio.

4.4 Comparisons with state-of-the-art techniques

In this section, we compare our V2RA generation technique to both physics-based methods and learning-based methods.

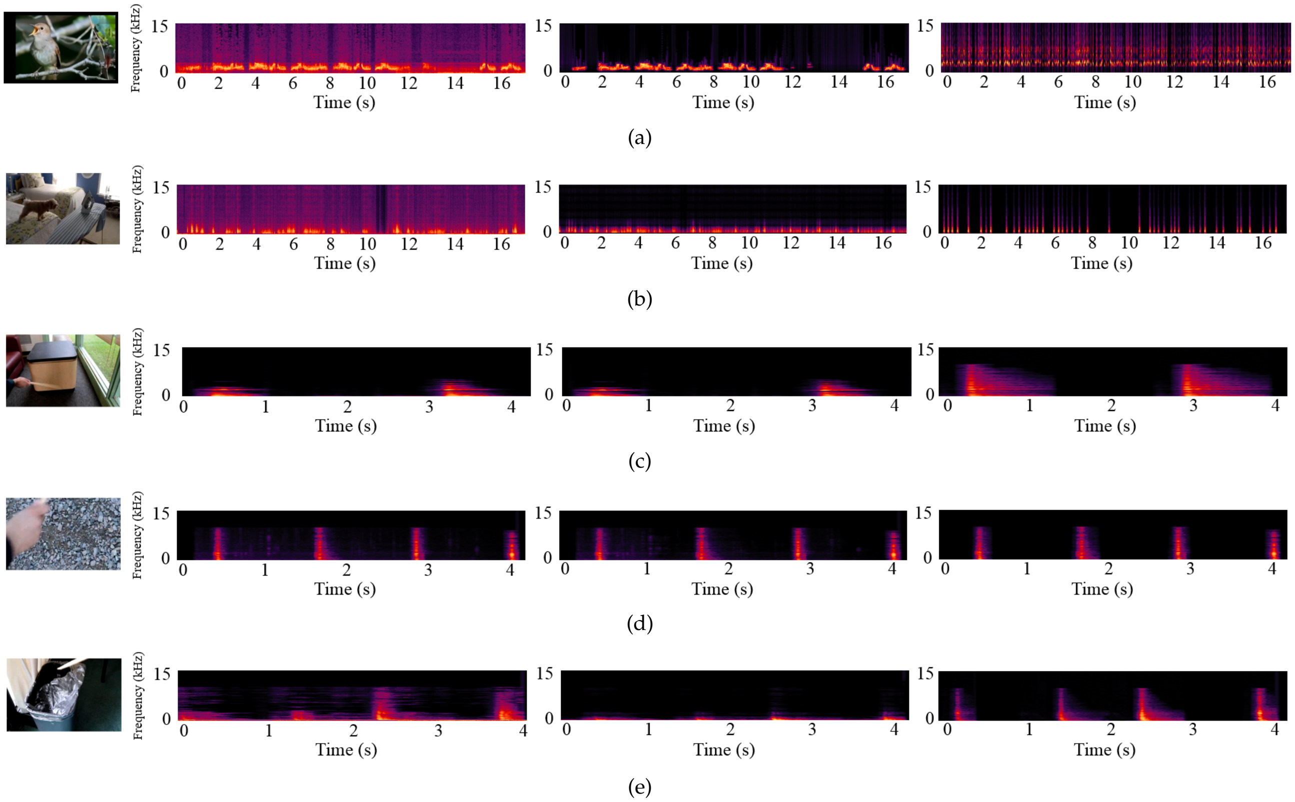
4.4.1 *Comparisons with physics-based methods*

As discussed in Section 2, physics-based methods can usually produce highly realistic sounds especially the example-guided methods that synthesize the results by adjusting and splicing samples. Thus, these methods have obvious advantages in timbre. Therefore, to show the timbre quality of our results, we compared our method with an example-guided physics-based sound synthesis method [2] for wood scenes. Figure 6 illustrates the audio results for two wood scenes synthesized by [2] and our method. We can find that the timbre of the two results are very similar.

We also quantitatively evaluated the audio results as shown in Table 5. The first two rows represent our similarity measurement for each type of method, and it can be observed that the audio produced by our method has a higher timbre consistency (lower OSG value). Then we compared the results synthesized by the two methods with real wood sound recordings. The results show that our method has a better match with the ground truth. Moreover, the method in [2] is a specialized method for generating impact sounds only, but our V2RA generation is a general method which can achieve comparable sound effects.

4.4.2 *Comparisons with learning-based methods*

We further compared our results with the state-of-the-art learning-based methods [12], [14]. The method in [12] can only synthesize cochleagram (a type of time-frequency representation) rather than raw audio. So this method needs extra input (cochleagram) to achieve ideal performance. In contrast, our method is end-to-end and no additional input is required. Moreover, our method is general and can directly generate the waveform. From the circled part in Figure 7(a), we can find that our result contains ambient noise that is closer to the ground truth. Zhou et al. [14] proposed a SampleRNN based method for automatic sound synthesis. This method is also end-to-end. However, because their network lacks the deep understanding module of visual content, the temporal match between visual content and audio is not ideal. As shown in Figure 7(b), in the video, the dog barks only once (the rest is background voice in



(a)

(b)

(c)

(d)

(e)

Fig. 9. Optimization schemes effects. (a)-(e) show the spectrum of original results, results after de-noising and results after peak-match, respectively.

the ground truth), while there were two barks (see the circled parts) in the result of [14]. In Figure 7(c), we can observe that redundant frequency bands (in the blue circles) appear which are quite different from the spectrogram of the ground truth. There was an obvious burst in our result, while there was continuous sound in the result of [Zhou et al. 2018]. These results show our method can achieve more synchronicity than previous work. Moreover, the method in [14] is not suitable for real-time applications because of the autoregressive model. As can be seen from Table 6, the ODG values are relatively close which indicates that the quality of the audio generated by the two methods is similar. However, the OSG values have been improved significantly.

4.5 Network architecture evaluation

We further evaluated the architecture of our proposed V2RA-GAN, based on both qualitative and quantitative indicators discussed in Section 4.3. The qualitative evaluation is the same as the aforementioned operation (Section 4.4). The results for our evaluation appear in Table 7. We added skip connections for the generator which is known as 'U-Net' in image processing. However, different from the great performance achieved in the image processing field, the addition of skip connections does not improve the audio quality (see Table 7). This may be due to the different structures between audio and image data. To determine if an audio filter improves the learning procedure, we compared the results with filter layers and without filter layers, and the results (Table 7) show that the filter layer can indeed improve the audio quality. More intuitively, we can observe that the sound will be clearer with filter layers (see Figure 8). We can also observe that adding a Gaussian random vector z does not improve the sound quality. Therefore, we do not use a Gaussian random vector in our design. We also adopt reshape operation instead of random cropping operation.

We also verified the effect of different loss functions on our network. As introduced in Section 3.2, the loss function of our V2RA-GAN is $L_1 + LS$. Thus, we first compared LS with $L_1 + CE$ (cross entropy) loss. From Table 7 we can observe that the $L_1 + LS$ loss function has the best quantitative evaluation score. Comparing the score for the LS loss function and $L_1 + LS$ loss function in Table 7, we argue that adding the L_1 loss function can greatly improve the quality of audio results. Therefore, by evaluating different combinations of L_1 loss, cross-entropy loss and LS loss, it can be concluded that the combination of L_1 loss and LS loss achieves the best performance.

4.6 Validation of the optimization schemes

Figure 9 illustrates the benefit of the optimization schemes proposed in Section 3.3 and Table 8 lists the quantitative evaluations. We present the spectra of five typical cases. From the spectra in Figure 9(a), we can find that the frequency of bird singing and frequency of background noise varies. Thus, the de-noising process is more suitable for signals whose frequency lies in a narrow band. The result of a peak-match algorithm contains some additional high-frequency information. In Figure 9(b), we notice that for sounds like dog barking, results after de-noising may lose part of the frequency information that is mixed with background noise. Therefore, the peak-match process is more suitable for environmental sound whose frequency is in

TABLE 6
Quantitative evaluation for the results by Zhou et al.'s method [14] and our method in Figure 7.

Audio	ODG	MMS	MSF	RMS	OSG
[14]	-0.1710	2.9515	11.5635	0.4626	7.9039
Our method	-0.1789	1.1951	4.3435	0.3373	3.0916
Ground truth	-0.2157	0	0	0	0

TABLE 7
Network architecture evaluation.

Experiment	ODG	MMS	MSF	RMS	OSG
Full system	**-0.2124**	0.4986	0.3903	0.0583	**0.5134**
+ $\boldsymbol{z}$ as input	-0.2237	0.7325	0.4417	0.0832	0.6846
+ skip connections for the generator	-0.3362	3.4587	1.2723	0.0425	2.5429
- audio filter (the last layer of the generator)	-0.2319	11.3408	21.7192	0.0330	11.3408
+ random crop	-0.2202	0.2570	20.6246	0.0230	10.8536
$L_1 + LS$ loss function	**-0.2124**	0.4986	0.3903	0.0583	**0.5134**
$L_1 + CE$ loss function	-0.2350	0.8933	43.8061	0.0670	23.3185
LS loss function	-0.2216	1.2525	58.7788	0.1429	31.3402
L_1 loss function	-0.2370	0.3037	8.4975	0.1209	4.6462
CE loss function	-0.2327	0.1730	9.3176	0.1303	5.0024
ground truth	-0.2343	0	0	0	0

a wide band. Furthermore, we notice from the spectra in Figures 9(c)- 9(e) that there is no significant difference before and after post-processing. This is due to the fact that the training data in these categories contain little environmental noise. As a result, our generated results do not have much background noise either. Moreover, we also compared the effect of different recordings selected by users in the peak-match process. Because of the variety of these sounds, although choosing different recordings will produce different results (see Figure 10), it does not affect the authenticity of the final video. The results can be heard in the accompanying video.

4.7 User study

In addition to the quantitative evaluation, we further conducted a user study to qualitatively evaluate the proposed model with the two optimization schemes and qualitative evaluation is the same as the aforementioned operation (Section 4.4). In the experiment, we compared the video with the audio generated by our method with the real recorded video from the dataset. Specifically, twenty participants were involved in our user study, whose ages are around twenty. We randomly chose two examples from each category and showed them to participants. In each scenario, the participant is asked a question: "How do you rate the quality of this video?" The score for each clip is on a scale from 1 to 5, where 1 is labelled "Very bad" and 5 "Very good". The quality of sound was evaluated from two aspects: sound clarity and synchronization between sound and visual contents. The results are illustrated in Figure 11.

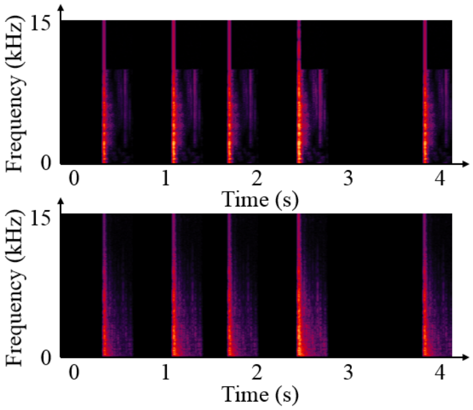


Fig. 10. Comparisons of different recordings utilized in the peak-match process. Although the timbre varies with different recordings (in the gravel category), it does not affect the synchronization of the results.

Observe in Figure 11 that for the videos with clear sound (gravel, wood, cloth and plastic), participants were easier to distinguish which video is ground truth. This is because the real audio is recorded in a lab environment and has been denoised. Therefore, real audio is clearer and easier to achieve high scores. As for the other three categories, participants have trouble distinguishing which video is the real one. Overall, our generated audio achieves good scores and have good quality.

We also conducted another user study to show that our model can learn good correspondence between audio and

TABLE 8
Optimization schemes evaluation.

Categories	Original result (ODG/OSG)	after de-noising (ODG/OSG)	after peak-match (ODG/OSG)
Bird	-0.2347 / 9.6662	-0.1319 / 8.4831	-0.1572 / 10.7734
Dog	-0.2081 / 3.5430	-0.1729 / 3.4476	-0.1697 / 1.1701
Firework	-0.3129 / 1.3317	-0.2744 / 0.7514	-0.2617 / 0.7374
Cloth	-0.0909 / 0.1124	-0.0907 / 0.1120	-0.0907 / 0.1097
Wood	-0.1155 / 0.0308	-0.1074 / 0.0307	-0.1050 / 0.0308
Gravel	-0.1327 / 0.0217	-0.1225 / 0.0223	-0.1314 / 0.0217
Plastic	-0.1156 / 0.0778	-0.1147/ 0.1746	-0.1155 / 0.0658

TABLE 9
Comparison with previous work.

Method	sound quality	synchronization	overall quality
[12]	4.50	4.75	4.40
Ours	4.50	4.75	4.50
[14]	3.35	2.85	3.00
Ours	3.65	4.15	3.35

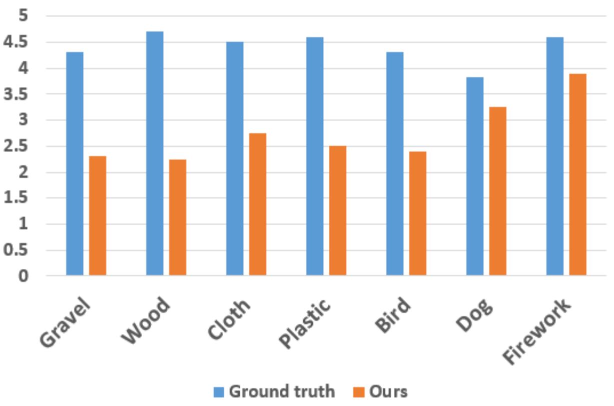


Fig. 11. User study results on the audio quality.

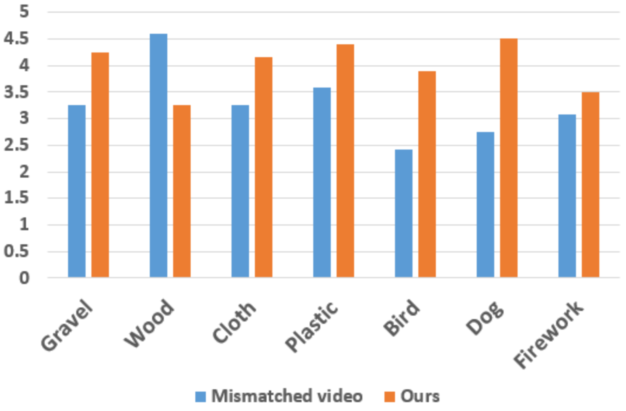


Fig. 12. User study results on the correspondence between audio and visual.

visual. To compare with our generated results, we synthesized mismatched videos artificially. Specifically, for each given video, the mismatched audio was randomly chosen from another video of the same category. The results are illustrated in Figure 12. We can observe that our generated results achieve higher scores overall. This shows that our results have a good match between visual and audio.

We also compared our results with previous work. We compared dog and firework sounds with [14]. We also compared gravel, wood, cloth and plastic sounds with [12]. For a more obvious comparison, we asked each participant three questions: “How much do you rate the sound quality of this video?”, “How much do you rate the synchronization between sound and visual contents?”, and “How much do you rate the overall quality of this video?”. We calculated the average scores of different categories and the results are illustrated in Table 9. It can be observed in Table 9 that our results achieve higher scores overall and have good synchronicity.

5 Conclusions and Future Work

We designed a conditional GAN approach for V2RA generation. To bridge the gaps between video and audio, we proposed a novel video encoding process and a novel V2RA-GAN architecture. Since the generated results were limited by the video quality in the dataset, we provided two audio optimization schemes for audio quality improvement. Various experiments demonstrated that our method can achieve comparable audio results.

Our method is not without limitations. One of the main remaining challenges is how to improve the robustness to different video qualities. Figure 13 shows a failure case of an overexposed video. Since the input video is not clear

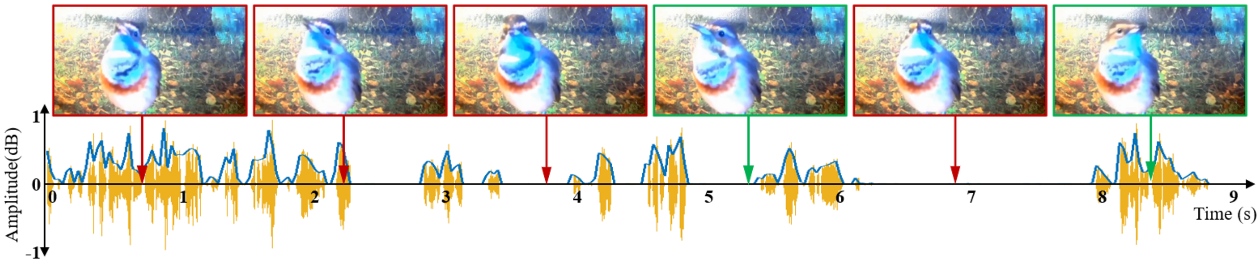


Fig. 13. A failure case. The waveform (yellow) and envelope (blue) show the change of audio. The input video is overexposed, resulting in unclear image contents. The video frames in red boxes are correctly matched, however, the video frames in green boxes are mismatched.

(overexposed in Figure 13), the proposed network cannot accurately detect changes in the video content. As a result, we can observe that the synthesized sound and the video content are out of synchronization. Tackling such types of challenges would need a deeper understanding of video features. Another failure case is testing with videos in mismatched categories. For example, our model can not generate dog barking with bird singing videos as the training set. Overall, we hope that this work can catalyze future investigation of GANs for V2RA generation, and open up more possible directions for future research including environmental sound recognition and environmental scene understanding.

References

[1] C. Zheng and D. L. James, "Harmonic fluids," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 37:1–37:12, 2009.

[2] Z. Ren, H. Yeh, and M. C. Lin, "Example-guided physically based modal sound synthesis," *ACM Transactions on Graphics*, vol. 32, no. 1, pp. 1–16, 2013.

[3] A. Davis, M. Rubinstein, N. Wadhwa, G. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: passive recovery of sound from video," *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 79:1–79:10, 2014.

[4] W. Moss, H. Yeh, J. M. Hong, M. C. Lin, and D. Manocha, "Sounding liquids: automatic sound synthesis from fluid simulation," *ACM Transactions on Graphics*, vol. 29, no. 3, pp. 21:1–21:13, 2010.

[5] T. R. Langlois, C. Zheng, and D. L. James, "Toward animating water with complex acoustic bubbles," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 95:1–95:13, 2016.

[6] Q. Yin and S. Liu, "Sounding solid combustibles: non-premixed flame sound synthesis for different solid combustibles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 2, pp. 1179–1189, 2018.

[7] G. Cirio, A. Qu, and G. Drettakis, "Multi-scale simulation of nonlinear thin-shell sound with wave turbulence," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 110:1–110:14, 2018.

[8] S. Liu, H. Cheng, and Y. Tong, "Physically-based statistical simulation of rain sound," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 123:1–123:14, 2019.

[9] T. Tapio and H. James, "Sound rendering," *Computer Graphics*, vol. 26, no. 2, pp. 211–220, 1992.

[10] B. Truax, "Real-time granular synthesis with a digital signal processor," *Computer Music Journal*, vol. 12, no. 2, pp. 14–26, 1988.

[11] E. R. Miranda, "Granular synthesis of sounds by means of a cellular automaton," *Leonardo*, vol. 28, no. 4, pp. 297–300, 1995.

[12] A. Owens, P. Isola, J. Mcdermott, A. Torralba, E. H. Adelson, and W. T. Freeman, "Visually indicated sounds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2405–2413.

[13] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep cross-modal audio-visual generation," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017, pp. 349–357.

[14] 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.

[15] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *ICLR*, 2017.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[17] Y. Dobashi, T. Yamamoto, and T. Nishita, "Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics." *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 732–740, 2003.

[18] ——, "Synthesizing sound from turbulent field using sound textures for interactive fluid simulation," *Computer Graphics Forum*, vol. 23, no. 3, pp. 539–545, 2004.

[19] J. N. Chadwick and D. L. James, "Animating fire with sound," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 84:1–84:8, 2011.

[20] S. Liu and Z. Yu, "Sounding fire for immersive virtual reality," *Virtual Reality*, vol. 19, no. 3-4, pp. 291–302, 2015.

[21] R. Nordahl, L. Turchet, and S. Serafin, "Sound synthesis and evaluation of interactive footsteps and environmental sounds rendering for virtual reality applications." *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 9, pp. 1234–1244, 2011.

[22] L. Peltola, C. Erkut, P. R. Cook, and V. Valimaki, "Synthesis of hand clapping sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1021–1029, 2007.

[23] D. B. Lloyd, N. Raghuvanshi, and N. K. Govindaraju, "Sound synthesis for impact sounds in video games," in *ACM Symposium on Interactive 3D Graphics and Games*, 2011, pp. 55–62.

[24] G. Cirio, D. Li, E. Grinspun, M. A. Otaduy, and C. Zheng, "Crumpling sound synthesis," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 181:1–181:11, 2016.

[25] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–11, 2018.

[26] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in neural information processing systems*, 2016, pp. 892–900.

[27] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–648.

[28] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, "Temporal video segmentation to scenes using high-level audiovisual features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1163–1177, 2011.

[29] M. S. Hossain, G. Muhammad, B. Song, M. M. Hassan, A. Alelaiwi, and A. Alamri, "Audio–visual emotion-aware cloud gaming framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 2105–2118, 2015.

[30] S. Zhang, S. Zhang, T. Huang, W. Gao, and Q. Tian, "Learning affective features with a hybrid deep model for audio–visual emotion recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 3030–3043, 2017.

[31] N. Takahashi, M. Gygli, and L. V. Gool, "Aenet: learning deep audio features for video analysis," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 513–524, 2017.

[32] X. Yang, T. Zhang, and C. Xu, "Text2video: an end-to-end learning framework for expressing text with videos," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 2360–2370, 2018.

[33] J. Li, X. Liang, S. M. Shen, T. Xu, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2017.

[34] S. Liu and X. Zhang, "Image decolorization combining local features and exposure features," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2461–2472, 2019.

[35] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[36] C. Donahue, J. Mcauley, and M. Puckette, "Synthesizing audio with generative adversarial networks," *arXiv:1802.04208*, 2018.

[37] J. S. Santiago Pascual, Antonio Bonafonte, "SEGAN: speech enhancement generative adversarial network," in *INTERSPEECH*, 2017, pp. 1–5.

[38] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2014.

[39] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *NIPS*, 2015, pp. 1486–1494.

[40] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv:1605.05396*, 2016.

[41] Y. Yu and S. Canales, "Conditional lstm-gan for melody generation from lyrics," *arXiv:1908.05551*, 2019.

[42] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[43] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[44] G. Yu, S. Mallat, and E. Bacry, "Audio denoising by time-frequency block thresholding," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1830–1839, 2008.

[45] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *ICLR*, 2015, pp. 1–15.

[46] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[47] B. Gygi, G. R. Kidd, and C. S. Watson, "Similarity and categorization of environmental sounds," *Perception and Psychophysics*, vol. 69, no. 6, pp. 839–855, 2007.

[48] A. A. D. Lima, F. P. Freeland, R. A. D. Jesus, and B. C. Bispo, "On the quality assessment of sound signals," in *IEEE International Symposium on Circuits and Systems*, 2008, pp. 416–419.

[49] I. Recommendation, "Method for objective measurements of perceived audio quality," *ITU-R BS*, vol. 13871, 2001.

[50] P. Kabal., "An examination and interpretation of ITU-R BS. 1387: perceptual evaluation of audio quality," *TSP Lab Technical Report*, pp. 1–89, 2002.

Shiguang Liu received the Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, P.R. China. He is currently a Professor with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, P.R. China. His research interests include image/video processing, computer graphics, visualization, and virtual reality.

Sijia Li is currently working toward the M.S. degree at the the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, P.R. China. Her research interests include deep learning and cross-modal learning.

Haonan Cheng is currently working toward the Ph.D. degree at the the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, P.R. China. Her research interest is computer graphics.