Deep Cross-Modal Audio-Visual Generation

Lele Chen*
Computer Science
University of Rochester
lchen63@cs.rochester.edu

Sudhanshu Srivastava*
Computer Science
University of Rochester
ssrivas6@cs.rochester.edu

Zhiyao Duan
Electrical and Computer Engineering
University of Rochester
zhiyao.duan@rochester.edu

Chenliang Xu
Computer Science
University of Rochester
chenliang.xu@rochester.edu

ABSTRACT

Cross-modal audio-visual perception has been a long-lasting topic in psychology and neurology, and various studies have discovered strong correlations in human perception of auditory and visual stimuli. Despite work on computational multimodal modeling, the problem of cross-modal audio-visual generation has not been systematically studied in the literature. In this paper, we make the first attempt to solve this cross-modal generation problem leveraging the power of deep generative adversarial training. Specifically, we use conditional generative adversarial networks to achieve cross-modal audio-visual generation of musical performances. We explore different encoding methods for audio and visual signals, and work on two scenarios: instrument-oriented generation and pose-oriented generation. Being the first to explore this new problem, we compose two new datasets with pairs of images and sounds of musical performances of different instruments. Our experiments using both classification and human evaluation demonstrate that our model has the ability to generate one modality, i.e., audio/visual, from the other modality, i.e., visual/audio, to a good extent. Our experiments on various design choices along with the datasets will facilitate future research in this new problem space.

CCS CONCEPTS

• **Computing methodologies → Image representations**; **Neural networks**;

KEYWORDS

cross-modal generation, audio-visual, generative adversarial networks

ACM Reference format:
Lele Chen, Sudhanshu Srivastava, Zhiyao Duan, and Chenliang Xu. 2017. Deep Cross-Modal Audio-Visual Generation. In *Proceedings of Thematic-Workshops'17, Mountain View, CA, USA, October 23–27, 2017,* 9 pages.

*These authors contributed equally to this work.

ThematicWorkshops'17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-5416-5/17/10…$15.00
https://doi.org/10.1145/3126686.3126723

https://doi.org/10.1145/3126686.3126723

1 INTRODUCTION

Cross-modal perception, or intersensory phenomenon, has been a long-lasting research topic in numerous disciplines such as psychology [3, 28, 30, 31] , neurology [27], and human-computer interaction [14, 29], and recently gained attention in computer vision [17], audition [10] and multimedia analysis [6, 19]. In this paper, we focus on the problem of cross-modal audio-visual generation. Our system is trained with pairs of visual and audio signals, which are typically contained in videos, and is able to generate one modality (visual/audio) given observations from the other modality (audio/visual). Fig. 1 shows results generated by our system on a musical performance video dataset.

Learning from multimodal input is challenging—despite the many works in cross-modal analysis, a large portion of the effort, e.g., [6, 19, 21, 32], has been focused on indexing and retrieval instead of generation. Although joint representations of multiple modalities and their correlations are explored, these methods only need to retrieve samples that exist in a database. They do not, for example, need to model the details of the samples, which is required in data generation. On the contrary, the generation task requires generating novel images and sounds that are *unseen* or *unheard*, and is of great interest to many applications, such as creating art works [8, 33] and zero-shot learning [2]. It requires learning a complex generative function that produces meaningful outputs. In the case of cross-modality generation, this function has to map from one modality space to the other modality space, making the problem even more challenging and interesting.

Generative Adversarial Networks (GANs) [7] have become an emerging topic in deep generative models. Inspired by Reed et al.'s work on generating images conditioned on text captions [23], we design conditional GANs for cross-modal audio-visual generation. Different from their work, we make the networks to handle intersensory generation—generate images conditioned on sounds and generate sounds conditioned on images. We explore two different tasks when generating images: instrument-oriented generation (see Fig. 1) and pose-oriented generation (see Fig. 10), where the latter task is treated as fine-grained generation comparing to the former.

Another key aspect to the success of cross-modal generation is being able to effectively encode and decode information contained in different modalities. For images, Convolutional Neural Networks (CNNs) are known to perform well in various tasks. Therefore, we

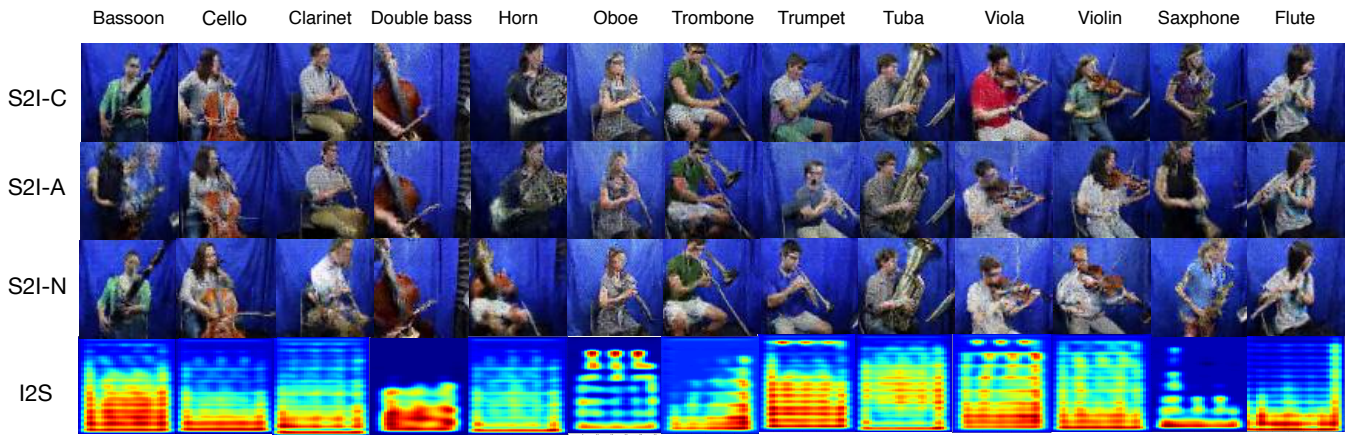

Figure 1: Generated outputs using our cross-modal audio-visual generation models. Top three rows are musical performance images generated by our Sound-to-Image (S2I) networks from audio recordings. S2I-C is our main model. S2I-A and S2I-N are variations of our main model. Bottom row contains the log-mel spectrograms of generated audio of different instruments from musical performance images using our Image-to-Sound (I2S) network. Each column represents one instrument type.

train a CNN, use the output of the fully connected layer before softmax as the image encoder and use several deconvolution layers as the decoder/generator. For sounds, we also use CNNs to encode and decode. The input to the networks, however, cannot be the raw waveforms. Instead, we first transform the time-domain signal into the time-frequency or time-quefrency domain. We explore five different transformations and find that the log-mel spectrogram gives the best result.

To explore this new problem space, we compose two datasets, e.g., Sub-URMP and INIS. The Sub-URMP dataset consists of paired images and sounds extracted from 72 single-instrument musical performance videos of 13 kinds of instruments in the University of Rochester Multimodal Musical Performance (URMP) dataset [11]. In total 80,805 images are extracted and each image is paired with a half-second long sound clip. The INIS dataset contains ImageNet [4] images of five musical instruments, e.g., drum, saxophone, piano, guitar and violin. We pair each image with a short sound clip of a solo performance of the corresponding instrument. We conduct experiments to evaluate the quality of our generated images and sound spectrograms using both classification and human evaluation. Our experiments demonstrate that our conditional GANs can, indeed, generate one modality (visual/audio) from the other modality (audio/visual) to a good extent at both the instrument-level and the pose-level. We also compare and evaluate various design choices in our experiments.

The contributions are three-fold. First, to our best knowledge, we introduce the problem of cross-modal audio-visual generation and are the first to use GANs on intersensory generation. Second, we propose new network structures and adversarial training strategies for cross-modal GANs. Third, we compose two datasets that will be released to facilitate future research in this new problem space.

The paper is organized as follows. We discuss related work and background in Sec. 2. We introduce our network structure, training strategies and encoding methods in Sec. 3. We present our datasets in Sec. 4 and experiments in Sec. 5. Finally, we conclude our paper in Sec. 6.

2 RELATED WORK

Our work differs from other various work in cross-modal retrieval [6, 19, 21, 32] as stated in Sec. 1. In this section, we further distinguish our work from that in multimodal representation learning. Ngiam et al. [16] learn a shared representation between audio-visual modalities by training a stacked multimodal autoencoder. Srivastava and Salakhutdinov [26] propose a multimodal deep Boltzmann machine to learn a joint representation of images and their text tags. Kumar et al. [9] learn an audio-visual bimodal compositional model using sparse coding. Our work differs from them by using the adversarial training framework that allows us to learn a much deeper representation for the generator.

Adversarial training has recently received a significant amount of attention [1, 5, 7, 13, 20, 23, 24]. It has been shown to be effective in various tasks, such as generating semantic segmentations [12, 25], improving object localization [1], image-to-image translation [8] and enhancing speech [18]. We also use adversarial training but on a novel problem of cross-modal audio-visual generation with music instruments and human poses that differs from other works.

2.1 Background

Generative Adversarial Networks (GANs) are introduced in the seminal work of Goodfellow et al. [7], and consist of a generator network G and a discriminator network D. Given a distribution, G is trained to generate samples that are resembled from this distribution, while D is trained to distinguish whether the sample is genuine. They are trained in an adversarial fashion playing a min-max game against each other:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x\sim p_z(z)}[\log(1-D(G(z)))] \quad , \tag{1}$$

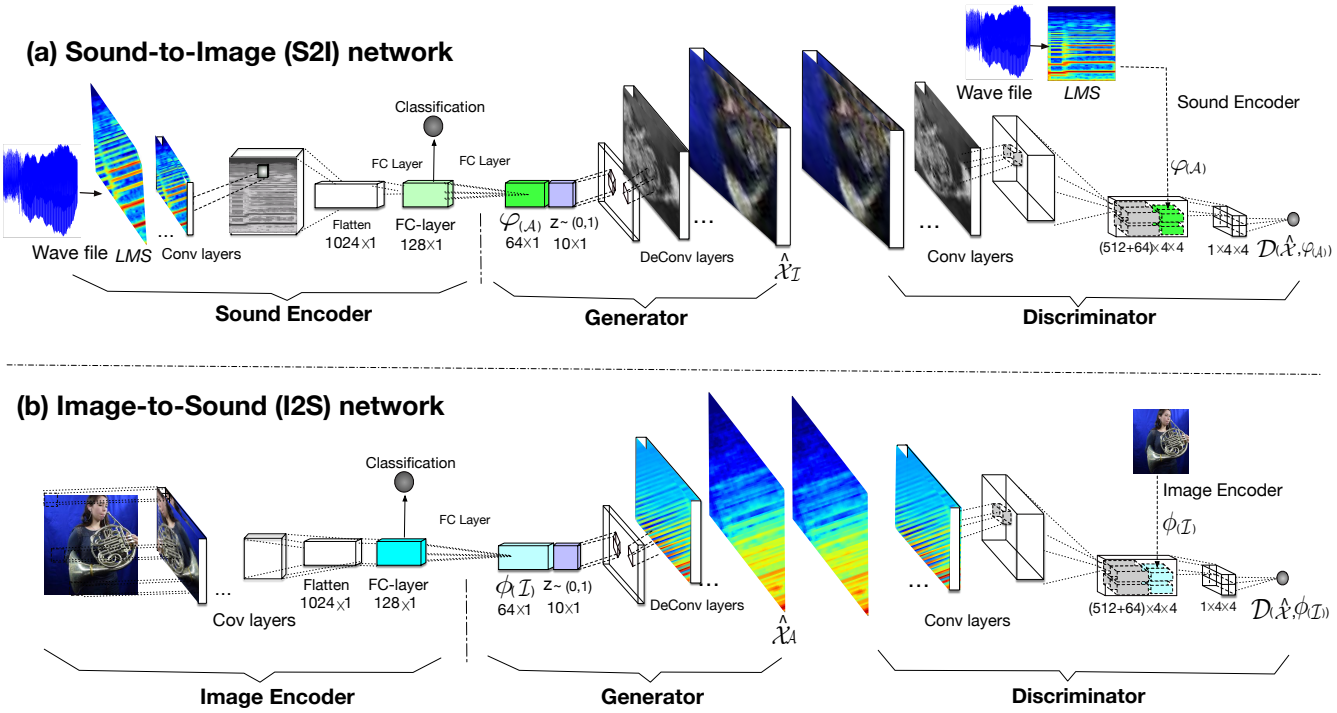

Figure 2: The overall diagram of our model. This figure consists of (a) an S2I GAN network and (b) an I2S GAN network. Each network contains an encoder, a generator and a discriminator, respectively.

where p_{data} is the target data distribution and z is drawn from a random noise distribution p_z.

Conditional GANs [5, 15] are variants of GANs, where one is interested in directing the generation conditioned on some variables, e.g., labels in a dataset. It has the following form:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{x\sim p_z(z)}[\log(1 - D(G(z|y)))] \quad , \tag{2}$$

where the only difference from GANs is the introduction of y that represents the condition variable. This condition is passed to both the generator and the discriminator networks. One particular example is [23], where they use conditional GANs to generate images conditioned on text captions. The text captions are encoded through a recurrent neural network as in [22]. In this paper, we use conditional GANs for cross-modal audio-visual generation.

3 CROSS-MODAL GENERATION MODEL

The overall diagram of our model is shown in Fig. 2, where we have separate networks for Sound-to-Image (S2I) and Image-to-Sound (I2S) generation. Each of them consists of three parts: an encoder network, a generator network, and a discriminator network. We describe the generator and discriminator networks in Sec. 3.1, and their training strategies in Sec. 3.2. We present the encoder networks for sound and image in Sec. 3.3 and Sec. 3.4, respectively.

3.1 Generator and Discriminator Networks

S2I Generator The S2I generator network is denoted as: $G_{S\mapsto I} : \mathbb{R}^{|\varphi(A)|} \times \mathbb{R}^Z \mapsto \mathbb{R}^I$. The sound encoding vector of size 128 is first compressed to a vector of size 64 via a fully connected layer followed by a leaky ReLU, which is denoted as $\varphi(A)$. Then it is concatenated with a random noise vector $z \in \mathbb{R}^Z$. The generator takes this concatenated vector and produces a synthetic image $\hat{x}_I \leftarrow G_{S\mapsto I}(z, \varphi(A))$ of size 64x64x3.

S2I Discriminator The S2I discriminator network is denoted as: $D_{S\mapsto I} : \mathbb{R}^I \times \mathbb{R}^{|\varphi(A)|} \mapsto [0,1]$. It takes an image and a compressed sound encoding vector and produces a score for this pair being a genuine pair of image and sound.

I2S Generator Similarly, the I2S generator network is denoted as: $G_{I\mapsto S} : \mathbb{R}^{|\phi(I)|} \times \mathbb{R}^Z \mapsto \mathbb{R}^A$. The image encoding vector of size 128 is compressed to size 64 via a fully connected layer followed by a leaky ReLU, denoted as $\phi(I)$, and concatenated with a noise z. The generator takes it and do a forward pass to produce a synthetic sound spectrogram $\hat{x}_A \leftarrow G_{I\mapsto S}(z, \phi(I))$ of size 128x34.

I2S Discriminator The I2S discriminator network is denoted as: $D_{I\mapsto S} : \mathbb{R}^A \times \mathbb{R}^{|\phi(I)|} \mapsto [0,1]$. It takes a sound spectrogram and a compressed image encoding vector and produces a score for this pair being a genuine pair of sound and image.

Our implementation is based on the GAN-CLS by Reed et al. [23]. We extend it to handle the challenges in operating sound spectrograms which have a rectangular size. For the I2S generator network, after getting a 32x32x128 feature map, we apply two successive deconvolution layers, where each has a kernel of size 4x4 with

stride 2x1 and 1x1 zero-padding, and obtain a matrix of size 128x34. The I2S discriminator network takes sound spectrograms of size 128x34. To handle ground-truth spectrograms, we use the numpy resize function to resize them from 128x44 to 128x34. We apply two successive convolution layers, where each has a kernel of size 4x4 with stride 2x1 and 1x1 zero-padding. This results in a 32x32 square feature map. In practice, we have observed that adding more convolution layers in the I2S networks helps get better output in fewer epochs. We add two layers to the generator network and 12 layers to the discriminator network. During evaluation, we use the numpy resize function to get a matrix of size 128x44 for comparing with ground-truth spectrograms.

3.2 Adversarial Training Strategies

Without loss of generality, we assume that the training set contains pairs of images and sounds $\{(I_i^j, A_i^j)\}$, where I_i^j represents the jth image of the ith instrument category in our dataset and A_i^j represents the corresponding sound. Here, $i \in \{1, 2, 3 \ldots, 13\}$ represents the index to one of the musical instruments in our dataset, e.g., *cello* or *violin*. Note that even images and sounds within the same musical instrument category differ in terms of the player, pose, and musical note. We use $\mathbf{I}_{-i}$ to represent the set of all images of instruments of all the categories except the ith category, and use $\mathbf{I}_i^{-j}$ to represent the set of all images in the ith instrument category except the jth image. The sound counterparts, $\mathbf{A}_{-i}$ and $\mathbf{A}_i^{-j}$, are defined likewise.

Based on the input, we define three kinds of discriminator outputs: S_r, S_f and S_w. Here, S_r is the score for a *true* pair of image and sound that is contained in our training set, and S_f is the score for the pair where one modality is *generated* based on the other modality, and S_w is the score for the wrong pair of image and sound. Wrong pairs are sampled from the training dataset. The generator network is trained to maximize

$$\log(S_f) \ , \tag{3}$$

and the discriminator is trained to maximize

$$\log(S_r) + (\log(1 - S_w) + \log(1 - S_f))/2 \ . \tag{4}$$

Note that by using different types of wrong pairs, we can eventually guide the generator to solve various tasks.

S2I Generation (Instrument-Oriented) We train a single S2I model over the entire dataset so that it can generate musical performance images of different instruments from different input sounds. In other words, the same model can generate an image of person-playing-violin from an unheard sound of violin, and can generate an image of person-playing-saxophone from an unheard sound of saxophone.We apply the following training settings:

$$\begin{aligned} \hat{x}_I &\leftarrow G_{S\mapsto I}(\varphi(A_i^j), z) \\ S_f &= D_{S\mapsto I}(\hat{x}_I, \varphi(A_i^j)) \\ S_r &= D_{S\mapsto I}(I_i^j, \varphi(A_i^j)) \\ S_w &= D_{S\mapsto I}(\omega(\mathbf{I}_{-i}), \varphi(A_i^j)) \ , \end{aligned} \tag{5}$$

where $\hat{x}_I$ is the synthetic image of size 64x64x3, z is the random noise vector and $\varphi(A_i^j)$ is the compressed sound encoding. $\omega(\cdot)$ is a random sampler with a uniform distribution, and it samples images from the wrong instrument category to construct wrong pairs for calculating S_w. We use the sound-to-image network structure as in Fig. 2 (a).

S2I Generation (Pose-Oriented) We train a set of S2I models with one for each musical instrument category. Each model captures the relations between different human poses and input sounds of one instrument. For example, the model trained on violin image-sound pairs can generate a series of images of person-playing-violin with different hand movements according to different violin sounds. This is a fine-grained generation task compared to the previous instrument-oriented task. We apply the following training settings:

$$\begin{aligned} \hat{x}_I &\leftarrow G_{S\mapsto I}(\varphi(A_i^j), z) \\ S_f &= D_{S\mapsto I}(\hat{x}_I, \varphi(A_i^j)) \\ S_r &= D_{S\mapsto I}(I_i^j, \varphi(A_i^j)) \\ S_w &= D_{S\mapsto I}(\omega(\mathbf{I}_i^{-j}), \varphi(A_i^j)) \ , \end{aligned} \tag{6}$$

where the main difference from Eq. (5) is that here in constructing the wrong pairs we sample images from wrong images in the correct instrument category, $\mathbf{I}_i^{-j}$, instead of images in wrong instrument categories, $\mathbf{I}_{-i}$. Again, we use the network structure as in Fig. 2 (a).

I2S Generation We train a single I2S model over the entire dataset so that it can generate sound magnitude spectrograms of different instruments from different musical performance images. In other words, the same model can generate For example, the model generates a sound spectrogram of drum given an image that has a drum. The generator should not make mistakes on the type of instruments while generating spectrograms. In this case, we set the training as the following:

$$\begin{aligned} \hat{x}_A &\leftarrow G_{I\mapsto S}(\phi(I_i^j), z) \\ S_f &= D_{I\mapsto S}(\hat{x}_A, \phi(I_i^j)) \\ S_r &= D_{I\mapsto S}(A_i^j, \phi(I_i^j)) \\ S_w &= D_{I\mapsto S}(\omega(\mathbf{A}_{-i}), \phi(I_i^j)) \ . \end{aligned} \tag{7}$$

Recall that $\hat{x}_A$ is the generated sound spectrogram with size 128x34, and $\phi(I_i^j)$ is the compressed image encoding. We use the image-to-sound network as in Fig. 2 (b).

3.3 Sound Encoder Network

The sound files are sampled at 44,100 Hz. To encode sound, we first transform the raw audio waveform into the time-frequency or time-quefrency domain. We explore a set of representations including the Short-Time Fourier Transform (*STFT*), Constant-Q Transform (*CQT*), Mel-Frequency Cepstral Coefficients (*MFCC*), Mel-Spectrum (*MS*) and Log-amplitude of Mel-Spectrum (LMS). Figure 3 shows images of the above-mentioned representations for the same sound. We can see that *LMS* shows clearer patterns than other representations.

We further run a CNN-based classifier on these different representations. We use four convolutional layers and three fully connected layers (see Fig. 4). In order to prevent overfitting, we add penalties (l2 = 0.015) on layer parameters in fully connected layers, and we apply dropout (0.7 and 0.8 respectively) to the last two

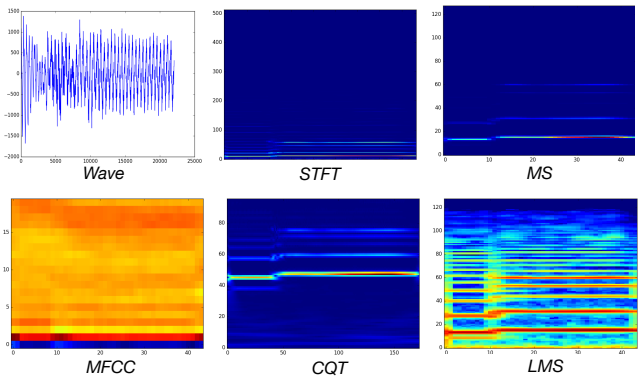

Figure 3: Different representations of audio that are fed to the sound encoder network. The horizontal axis is time and the vertical axis is amplitude (for Wave), frequency (for STFT, MS, CQT, and LMS) or quefrency (for MFCC).

Accuracy	*MS*	*LMS*	CQT	MFCC	STFT
3 layers	62.01%	***84.12 %***	73.00%	80.06%	74.05%
4 layers	66.09%	***87.44 %***	77.78%	81.05%	75.73%

Table 1: Accuracy of audio classifier. We apply three Conv layers and four Conv layers respectively and it shows that the best performance is achieved using four Conv layers.

layers. The classification accuracies obtained by different representations are shown in Table 1. We can see that *LMS* shows the highest accuracy. Therefore, we chose *LMS* over other representations as the input to the audio encoder network. Furthermore, *LMS* is smaller in size as compared to *STFT*, which saves the running time. Finally, we feed the output of the FC layer (size: 1x128) of CNNs classifier to GAN network as the audio feature.

Further merit of *LMS* is detailed in the experiment section. We thus choose *LMS* to represent the audio. To calculate *LMS*, a Short-Time Fourier Transform (STFT) with a 2048-point FFT window with a 512-point hop size is first applied to the waveform to get the linear-amplitude linear-frequency spectrogram. Then a mel-filter bank is applied to warp the frequency scale into the mel-scale, and the linear amplitude is converted to the logarithmic scale as well.

3.4 Image Encoder Network

For encoding images, we train a CNN with six convolutional layers and three fully connected layers (see Fig. 5). All the convolution kernels are of size 3x3. The last layer is used for classification with a softmax loss. This CNN image classifier achieves a high accuracy of more than 95 percent on the testing set. After the network is trained, its last layer is removed, and the feature vector of the second to the last layer having size 128 is used as the image encoding in our GAN network.

4 DATASETS

To the best of our knowledge, there is no existing dataset that we can directly work on. Therefore, we compose two novel datasets

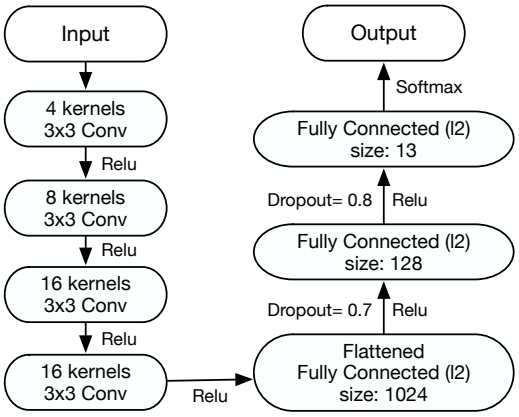

Figure 4: Audio classifier trained with instrument category loss.

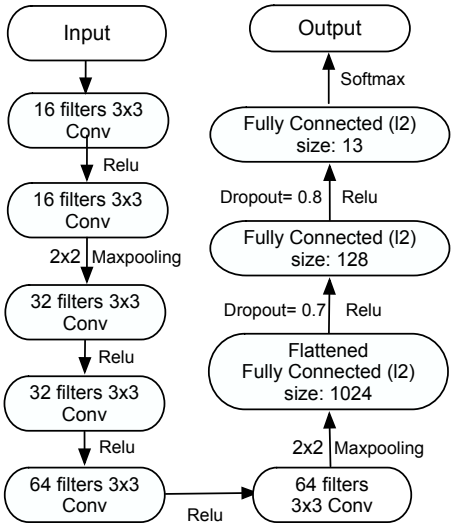

Figure 5: Image classifier trained with instrument category loss.

to train and evaluate our models, and they are a Subset of URMP (Sub-URMP) dataset and a ImageNet Image-Sound (INIS) dataset.

Sub-URMP dataset is assembled from the original URMP dataset [11]. It contains 13 musical instrument categories. In each category, there are recorded videos of 1 to 4 persons playing different music pieces (see Fig. 6). We separate about 80% videos for training and about 20% for testing and ensure that a video will not appear in both training and testing sets. We use a sliding window method to obtain the samples. The size of the sliding window is 0.5 seconds and the stride is 0.1 seconds. We use the first frame of each video chunk to represent the visual content of the sliding window. The audio files are in WAV format with a sampling rate of 44.1 kHz and a bit depth of 16. The image files are 1080P (1080x1920). There are a total of 80, 805 sound-image pairs in our composed Sub-URMP dataset. The basic information is shown as Table 2. We use this dataset as our main dataset to evaluate models in Sec. 5.

Figure 6: Examples from the Sub-URMP dataset. Each category contains roughly 6 different complete solo pieces.

Category	Cello	Double Bass	Oboe	Sax	Trumpet	Viola	Bassoon	Clarinet	Horn	Flute	Trombone	Tuba	Violin
Training set	9800	1270	4505	7615	1015	6530	1735	8125	5540	5690	8690	3285	7430
Testing Set	1030	1180	390	910	520	485	390	945	145	525	925	525	945

Table 2: Distribution of image-sound pairs in the Sub-URMP dataset.

Figure 7: Examples from the INIS dataset. Bottom row contains generated images by our S2I-A model . Due to large variation, images are not as good as those generated in the Sub-URMP dataset.

Category	Piano	Saxphone	Violin	Drum	Guitar
Complete songs	23	7	21	7	19
Training set	766	1171	631	1075	818
Testing set	327	500	269	460	349

Table 3: Distribution of image-sound pairs in INIS dataset.

All images in the INIS dataset are collected from ImageNet, shown in Fig. 7. There are five categories, and each contains roughly 1200 images. In order to eliminate noise, all images are screened manually. Audio files of this dataset come form a total of 77 solo performances downloaded from the Internet, such as a piano performance of Beethoven's *Moonlight Sonata* and a violin performance of Led Zeppelin's *Bonzo's Montreux*. We sample 7200 small audio chunks from all pieces with each having 0.5 second duration. We match the audio chunks to the instrument images to manually create sound-image pairs. Table 3 shows the statistics of this dataset.

5 EXPERIMENTS

We first introduce our model variations in Sec. 5.1, and then present our evaluation on instrument-oriented Sound-to-Image (S2I) generation in Sec. 5.2, pose-oriented S2I generation in Sec. 5.3 and Image-to-Sound (I2S) generation in Sec. 5.4.

5.1 Model Variations

We have three variations for our sound-to-image network.

S2I-C network This is our main sound-to-image network that uses classification-based sound encoding. The model is described in Sec. 3.

S2I-N network This model is a variation of the S2I-C network. It uses the same sound encoding but it is trained without the mismatch S_w information (see Eq. 5).

S2I-A network This model is a variation of the S2I-C network and differs in that it uses autoencoder-based sound encoding. Here, we use a stacked convolution-deconvolution autoencoder to encode sound. We use four stacks. For the first three stacks, we apply convolution and deconvolution, where the output of convolution is given as input to the next layer in stacks. In the last stack, the input (a 2D array of shape 120x36) is flattened and projected to a vector of size 128 via a fully connected layer. The network is trained to minimize MSE for all stacks in order.

5.2 Evaluating Instrument-Oriented S2I Generation

We show qualitative examples in Fig. 1 for S2I generation. It can be seen that the quality of the images generated by S2I-C is better than its variations. This is because the classifier is explicitly trained to classify the instruments from sound. Therefore, when this encoding is given as a condition to the generator network, it faces less ambiguity in deciding what to generate. Furthermore, while training the classifier, we observe the classification accuracy, which is a direct measurement of how discriminative the encoding is. This is not true in the case of autoencoder, where we know the loss function value, but we do not know if it is a good condition feature in our conditional GANs.

5.2.1 ***Human Evaluation.*** We have human subjects evaluate our sound-to-image generation. They are given 10 sets of images for each instrument. Each set contains four images; they are generated by S2I-C, S2I-N and S2I-A and a ground-truth image to calibrate the scores. Human subjects are well-informed about the music instrument category of the image sets. However they are not aware of the mapping between images to methods. They are asked to score the images on a scale of 0 to 3, where the meaning of each score is given in Table. 4.

Score	Meaning
3	Realistic image & match instrument
2	Realistic image & mismatch instrument
1	Fair image (player visible, instrument not visible)
0	Unrealistic image

Table 4: Scoring guideline of human evaluation.

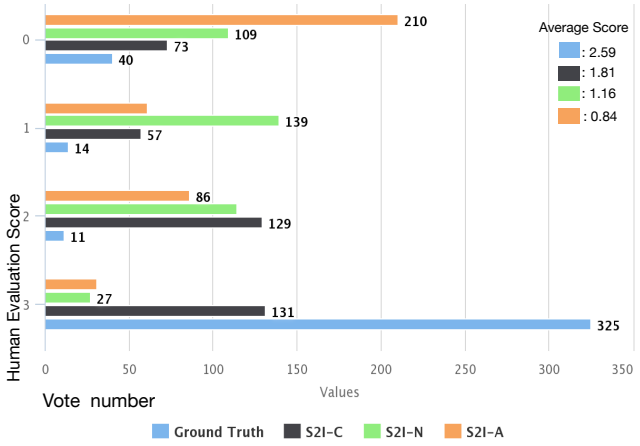

Figure 8: Result of human evaluation on generated images. The upper right shows average scores of S2I GANs on human evaluation.

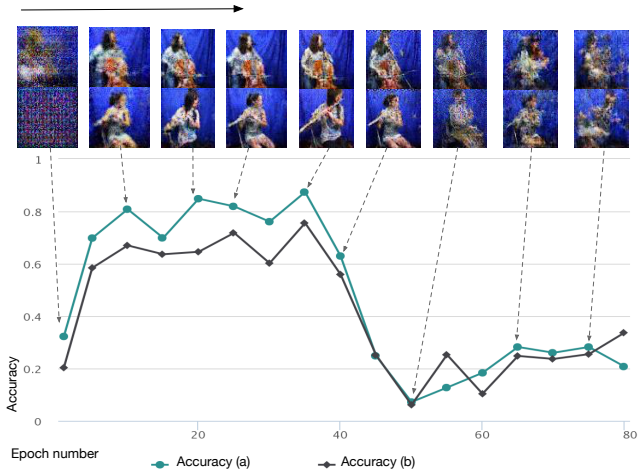

Figure 9: Evolution of image quality and classification accuracy on generated images versus the number of epochs. Accuracy (a) is the percentage of fake images generated in the training set of S2I-C that are classified into the right category by using classifier Γ. Accuracy (b) is the percentage of fake images generated in the test set of S2I-C that are classified into the right category by using classifier Γ.

Figure 8 shows the results of human evaluation. More than half of all images generated by S2I-C are considered as realistic by our human subjects, i.e. getting score 2 or 3. One third of them get score 3. This is much higher than S2I-N and S2I-A. In terms of mean score, S2I-C gets 1.81 where the ground-truth gets 2.59 due to small size; all images are evaluated at size 64x64.

Images from three instruments in particular were rated with a very high score among all images generated by S2I-C. Out of 30 Cello images, 18 received highest score of 3, while 25 received scores of 2 or above. Cello images received an average score of 1.9. Out of 30 Flute images, 15 received the highest possible score of 3, while 24 received a score of 2 or above. Flute images also received an average score of 2.1. Out of 30 Double-Bass images, 18 received a score of 3, while 21 received a score of 2 or more. The average score that Double-Bass images received was 2.02.

5.2.2 ***Classification Evaluation.*** We use the classifier used for encoding images (see Fig. 5) for evaluating our generated images. When classifying real images, the accuracy of the classifier is above 95%, thus we decide to use this classifier (Γ) to verify whether the generated (fake) images are belong to the expected instrument categories. We calculate the accuracies on images generated by S2I-C, S2I-A and S2I-N. Table. 5 shows the results. It shows that the accuracy of S2I-A and S2I-N is far worse than the accuracy of S2I-C.

5.2.3 ***Evolution of Classification Accuracy.*** Figure 9 shows the classification accuracy on images generated in both the training set and the test set. It is plotted for every fifth epoch. The model used for plotting this figure is our main S2I-C network. We visualize generated images for a few key moments in the figure. It shows that the accuracy increases rapidly up till the 35th epoch, and then begins to fall sharply till the 50th epoch, after which it again picks up a little, although the accuracy is still much lower than the peak accuracy. The training and testing accuracies follow nearly the same trend.

Epoch 50 has lower accuracy than epoch 60 onwards, despite the images in epoch 50 looking slightly better than later epochs. One potential reason is, at epoch 50, the discriminator has not lost its ability to discriminate real images against fake images, but has lost its classification ability. Thus the classifier can look at the generated image and predict a category, but the classes are random. In case of epochs 60 onwards, the generated images are random, so when they are fed into the classifier, the classifier just outputs the class with the most images.

In other words, images in epoch 50 look like images from the dataset, but they rarely correspond to the right instrument;the images after epoch 60, however, do not look like images from the dataset at all, and thus the classifier makes a guess according to the most populated class.

It is interesting to note that even the fifth epoch has much higher training and testing accuracies than any epoch after 40. What this means is that, even after as few as 5 epochs, not only are the images getting aligned with the expected category, the generated images also have enough quality so that a classifier can extract distinguishing features from them. This is not true in the case of a random image like the ones after epoch 50.

Mode	S2I-C	S2I-A	S2I-N
Training Set	87.37%	10.63%	12.62%
Testing Set	75.56%	10.95%	12.32%

Table 5: Classifier-based evaluation accuracy for images.

5.3 Evaluating Pose-Oriented S2I Generation

The model and the training strategy for our pose-oriented S2I generation is described in Sec. 3.2. The results we got were encouraging: various poses can be observed in the generated images (see Fig. 10). Note that for sound encoding, we used the same image classifier as S2I-C. It is trained to classify various instruments, not various poses. With a classifier that is trained to classify music notes, we expect the results to better match the expected poses.

Figure 10: Generated pose image. The first row shows viola images. The second and third rows are both violin images, which show that one single model can generate multiple persons with different poses. The fourth row is cello, where the variation of poses is more significant.For one instrument category, different videos were used in the training and test sets.

5.4 Evaluating I2S Generation

Because of the loss of phase information and the non-even frequency resolutions, the transformation from waveform to the *LMS* representation is not invertible. Therefore, we conduct evaluation on generated sound spectrograms instead of the waveforms. We use the sound classifier (see Fig. 4), which is trained to encode sound for image generation, to evaluate how discriminative the generated sound spectrograms are. The reason we use this model is because the model is trained on real *LMS*, and achieves a high accuracy of 80% on the test set of real *LMS*. We achieve 11.17% classification accuracy on the generated *LMS*.Furthermore, Figure 11 shows the generated *LMS* compared to the real *LMS*. We can see that, in generated *LMS*, there is less energy in the high frequency range and more energy in the low frequency range, which is the same as the real *LMS*.

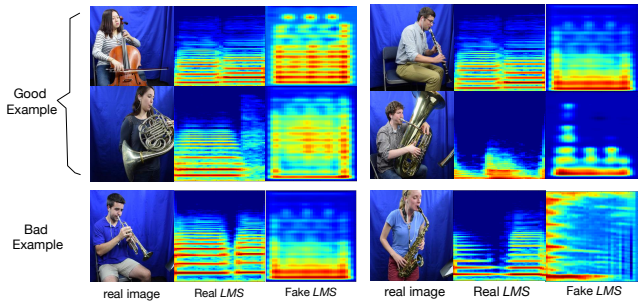

Figure 11: Generated sound spectrogram and ground-truth.

6 CONCLUSION

In this paper, we introduced the problem of cross-modal audio-visual generation and made the first attempt to use conditional GANs on intersensory generation. In order to evaluate our models, we composed two novel datasets, i.e., Sub-URMP and INIS. Our experiments demonstrated that our model can, indeed, generate one modality (visual/audio) from the other modality (audio/visual) to a good extent at both the instrument-level and the pose-level. For example, our model is able to generate poses of a cello player given the note that is being played.

Limitation and Future Work. While our I2S model generates the *LMS*, the accuracy is low. On the other hand, we are able to generate various poses using our S2I network, but it is hard to quantify how good the generation is. Strengthening the Autoencoder would enable accurate unsupervised generation. The present autoencoder appears to be limited in terms of extracting good representations. It is our future work to explore all these directions.

7 ACKNOWLEDGEMENT

We would like to thank Bochen Li and Yichi Zhang, Department of ECE, University of Rochester, for helpful suggestions and help with the URMP dataset.

REFERENCES

[1] Sima Behpour and Brian D Ziebart. 2016. Adversarial methods improve object localization. In *Advances in Neural Information Processing Systems Workshop.*
[2] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. 2016. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European Conference on Computer Vision.*
[3] Richard K Davenport, Charles M Rogers, and I Steele Russell. 1973. Cross-modal perception in apes. *Neuropsychologia* 11, 1 (1973), 21–28.
[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition.*
[5] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems.*
[6] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. 2014. Cross-modal retrieval with correspondence autoencoder. In *ACM International Conference on Multimedia.*
[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems.*
[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition.*
[9] S. Kumar, V. Dhiman, and J. J. Corso. 2014. Learning compositional sparse models of bimodal percepts. In *AAAI Conference on Artificial Intelligence.*
[10] Bochen Li, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. 2017. See and listen: score-informed association of sound tracks to players in chamber music

performance videos. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

[11] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. 2016. Creating a classical musical performance dataset for multimodal music analysis: Challenges, Insights, and Applications. In *arXiv:1612.08727*.

[12] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. 2016. Semantic segmentation using adversarial networks. In *arXiv:1611.08408*.

[13] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2016. Adversarial autoencoders. In *International Conference on Learning Representations*.

[14] Christophe Mignot, Claude Valot, and Noelle Carbonell. 1993. An experimental study of future "natural" multimodal human-computer interaction. In *INTERACT'93 and CHI'93 Conference Companion on Human Factors in Computing Systems*.

[15] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. In *arXiv:1411.1784*.

[16] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *International Conference on Machine Learning*.

[17] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. 2016. Visually indicated sounds. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[18] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. 2017. SEGAN: Speech Enhancement Generative Adversarial Network. In *arXiv:1703.09452*.

[19] Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Nikhil Rasiwasia, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. 2014. On the role of correlation and abstraction in cross-modal multimedia retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (2014), 521–535.

[20] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*.

[21] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A new approach to cross-modal multimedia retrieval. In *ACM International Conference on Multimedia*.

[22] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[23] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text-to-image synthesis. In *International Conference on Machine Learning*.

[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*.

[25] Nasim Souly, Concetto Spampinato, and Mubarak Shah. 2017. Semi and Weakly Supervised Semantic Segmentation Using Generative Adversarial Networks. In *arXiv:1703.09695*.

[26] Nitish Srivastava and Ruslan R Salakhutdinov. 2012. Multimodal learning with deep Boltzmann machines. In *Advances in Neural Information Processing Systems*.

[27] Barry E Stein and M Alex Meredith. 1993. *The merging of the senses*. The MIT Press.

[28] Russell L Storms. 1998. *Auditory-visual cross-modal perception phenomena*. Ph.D. Dissertation. Naval Postgraduate School.

[29] M Iftekhar Tanveer, Ji Liu, and M Ehsan Hoque. 2015. Unsupervised extraction of human-interpretable nonverbal behavioral cues in a public speaking scenario. In *ACM International Conference on Multimedia*.

[30] Bradley W Vines, Carol L Krumhansl, Marcelo M Wanderley, and Daniel J Levitin. 2006. Cross-modal interactions in the perception of musical performance. *Cognition* 101, 1 (2006), 80–113.

[31] Jean Vroomen and Beatrice de Gelder. 2000. Sound enhances visual perception: cross-modal effects of auditory organization on vision. *Journal of experimental psychology: Human perception and performance* 26, 5 (2000), 1583.

[32] Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. 2016. A Comprehensive Survey on Cross-modal Retrieval. In *arXiv:1607.06215*.

[33] Hang Zhang and Kristin Dana. 2017. Multi-style generative network for real-time transfer. In *arXiv:1703.06953*.