Taming Visually Guided Sound Generation

Vladimir Iashin
vladimir.iashin@tuni.fi
Esa Rahtu
esa.rahtu@tuni.fi
Computing Sciences
Tampere University
Tampere, Finland

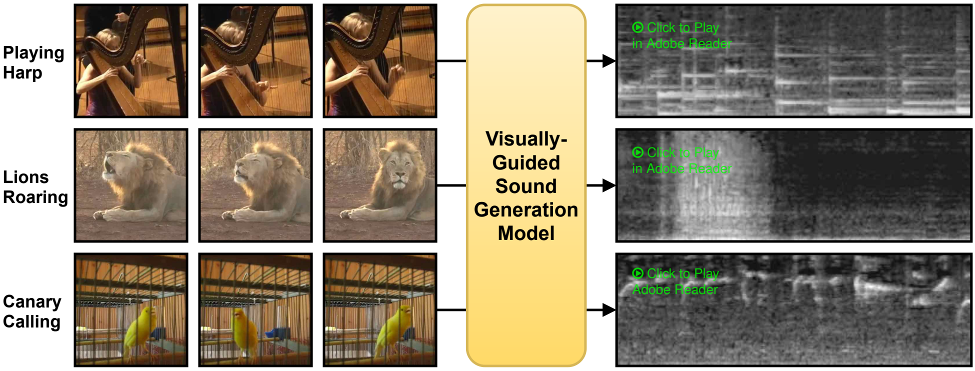


Figure 1: A single model supports the generation of visually guided, high-fidelity sounds for multiple classes from an open-domain dataset faster than the time it will take to play it.

Abstract

Recent advances in visually-induced audio generation are based on sampling short, low-fidelity, and one-class sounds. Moreover, sampling 1 second of audio from the state-of-the-art model takes minutes on a high-end GPU. In this work, we propose a single model capable of generating visually relevant, high-fidelity sounds prompted with a set of frames from open-domain videos in less time than it takes to play it on a single GPU.

We train a transformer to sample a new spectrogram from the pre-trained spectrogram codebook given the set of video features. The codebook is obtained using a variant of VQGAN trained to produce a compact sampling space with a novel spectrogram-based perceptual loss. The generated spectrogram is transformed into a waveform using a window-based GAN that significantly speeds up generation. Considering the lack of metrics for automatic evaluation of generated spectrograms, we also build a family of metrics called FID and MKL. These metrics are based on a novel sound classifier, called Melception, and designed to evaluate the fidelity and relevance of open-domain samples.

Both qualitative and quantitative studies are conducted on small- and large-scale datasets to evaluate the fidelity and relevance of generated samples. We also compare our model to the state-of-the-art and observe a substantial improvement in quality, size, and computation time. Code, demo, and samples: `v-iashin.github.io/SpecVQGAN`

1 Introduction

A user-controlled sound generation has many applications for *e.g.* movie and music production. Currently, foley designers are required to search through large databases of sound effects to find a suitable sound for a scene. A less painstaking approach would be to automatically generate a novel and relevant sound, given a few visual cues. Recent advances in deep learning brought to light many promising models for user-controlled content synthesis.

Previous works have proposed models to controllably generate *e.g.* images [13, 17, 35, 41, 44, 46, 50, 52, 64, 66, 67], videos [6, 12, 25, 34, 38, 42, 59, 60, 60, 63], and audios [1, 9, 15, 22, 24, 43, 57, 58], or separate sounds [18, 19, 69, 70, 74]. However, most of the audio works are music-related, and only a few attempts have been made to generate visually guided audio in an open domain setup [11, 73]. These methods rely on a one-model-per-class approach, which can be prohibitively expensive to scale to hundreds of classes.

Our goal in this paper is to build a single model that is capable of generating sounds conditioned on visual input from multiple classes with a restricted time budget. To address this, we propose to learn a prior in a form of the Vector Quantized Variational Autoencoder (VQVAE) codebook [61] and operate on spectrograms for efficiency. To shrink the sampling space more aggressively, we draw on advances in controlled image generation [17] relying on a variant of VQVAE with adversarial loss and introduce a novel spectrogram perceptual loss.

Such an approach allows us to reliably reconstruct a high-fidelity spectrogram from a smaller representation resolution. We, thus, can train a transformer on a shorter sequence to sample from the codebook and autoregressively construct a high-fidelity spectrogram while being conditioned on the visual cues. Finally, we vocode the spectrogram into a waveform using a variant of MelGAN [32] suitable for open-domain applications.

Human evaluation of content generation models is an expensive and tedious procedure. In the image generation field, this problem is bypassed with the automatic evaluation of fidelity using a family of metrics based on an ImageNet-pretrained [14] Inception model [56] *e.g.* Inception Score [53], Fréchet- [27] and Kernel Inception Distance [4] (FID & KID). The automatic evaluation of a sound generation model, however, remains an open question.

FID was adapted to assess fidelity of the generated audio in [30]. This metric is designed for very short sounds (<1 second) and, therefore, has limited applicability for long audio as it may miss long-term cues. Another challenge in the visually guided sound generation is to reliably estimate the relevance of produced samples. To mitigate both problems, we propose a family of metrics for fidelity and relevance evaluation based on a novel architecture called Melception, trained as a classifier on VGGSound [7], a large-scale open-domain dataset.

The main contributions of this work are: **(1)** a novel efficient approach for multi-class visually guided sound synthesis that relies on a transformer trained to sample from a codebook-based prior; **(2)** a new perceptual loss for spectrogram synthesis, called LPAPS. The loss relies on a novel general-purpose sound classifier, referred to as VGGish-ish, and helps VQ-VAE to learn reconstruction of higher-fidelity spectrograms from small-scale representations; **(3)** a novel set of metrics suitable for automatic evaluation of the fidelity and relevance of spectrogram synthesis, called Melception-based FID and MKL. We show the effectiveness of our approach in comparison with prior work and provide an extensive ablation study on small- and large-scale datasets (VAS and VGGSound) for visually guided sound synthesis.

2 Related Work

Codebook-based Content Generation The use of condensed prior information in a form of a codebook has been shown to effectively reduce the sampling space of generative algorithms. The initial idea was proposed in the seminal work [61] (VQVAE) and further improved in [51] (VQVAE-2). Applications of VQVAE for content generation include images [51, 61], audio [15, 37, 61, 71], and videos [49, 65]. Recently, it was found to be beneficial to train a transformer to sample from the codebook given a rich condition *e.g.*

text [16, 50], low-resolution image, semantic, edge, and depth-maps [17]. Our method, in contrast, is conditioned on a sequence of video frames and generates spectrograms.

Automatic Evaluation of Audio Synthesis While still being an open research question, few promising ideas have been proposed for the automatic evaluation of audio synthesis. Specifically, Kilgour *et al.* [30] adapted FID [27] to evaluate the fidelity of music enhancement algorithms. Unfortunately, the proposed method operates on 1-second windows and, therefore, does not utilize long-term cues. A similar approach was shown on a text-to-speech task in [5]. Alternatively, a model trained on human judgments has been employed as a perceptual loss during training [39]. However, collecting training material for a large-scale dataset poses significant budget requirements. In this paper, we propose a set of metrics designed to measure both the fidelity and relevance of prolonged open-domain spectrograms.

Instrument Music Generation With Visual Cues Generating short music audios became a testbed for many cross-modal generation algorithms. Owens *et al.* [45] pioneered the task by collecting a dataset of short videos containing hitting/scratching drumsticks against objects and used a combination of AlexNet [31] and LSTM [28] as a baseline. Chen *et al.* [9] focused on the generation of an image from the audio and vice-versa for single-instrument performance videos from the URMP dataset [36] using two Generative Adversarial Nets (GAN) [21] while Hao *et al.* [24] improved the performance of the GAN with cross-modal cycle-consistency [72]. Furthermore, Tan *et al.* [57] incorporated self-attention [62] into the GAN architecture and Su *et al.* [55] proposed to generate a piano sound by vocoding Midi predicted from a video. Recently, Kurmi *et al.* [33] brought a generation of short (1s) musical videos into the picture. These methods, however, focus on short ($\sim$1 second) music videos recorded in a controlled setting while our model operates on open-domain 10-second videos.

Open-domain Audio Generation Based on Visual Cues The generation of audio given a set of open-domain visual cues is a novel and challenging task. The first attempt to solve the task was published by Chen *et al.* [8] who proposed to employ a subset of AudioSet [20] to train a model to learn a residual to an average spectrogram for a video class. However, more relevant and higher-fidelity results were obtained by training a separate model for each video class. Namely, Zhou *et al.* [73] trained a separate SampleRNN [40] to generate a waveform for each of the 10 classes in the proposed dataset (VEGAS). Current state-of-the-art results in the generation of relevant and high-fidelity sounds for a video were shown by Chen *et al.* [11] (RegNet). They noticed the negative impact of "unseen" background sound on training dynamics and introduced a ground-truth-based regularizer and an enhanced version of the VEGAS dataset (VAS). While producing the most appealing results, the models are trained for each data class and the sampling speed is slow limiting the applicability of the model. In this paper, we propose a model that is capable of generating visually relevant sounds from videos of multiple classes in a time that is less than it takes to play the sound.

3 Framework

We aim to generate visually relevant and high-fidelity sounds. The main challenge is to design a model that handles videos of multiple categories and operates in real-time. Thus, we train a transformer to autoregressively compose a concise codebook representation of a spectrogram primed with a small set of frame-wise features obtained from a video (Sec. 3.2). The representation is then used in the pretrained codebook decoder to produce a spectrogram as outlined in Sec. 3.1. Finally, a waveform is reconstructed from the spectrogram using a pretrained vocoder as defined in Sec. 3.3. An overview of the architecture is shown in Fig. 2.

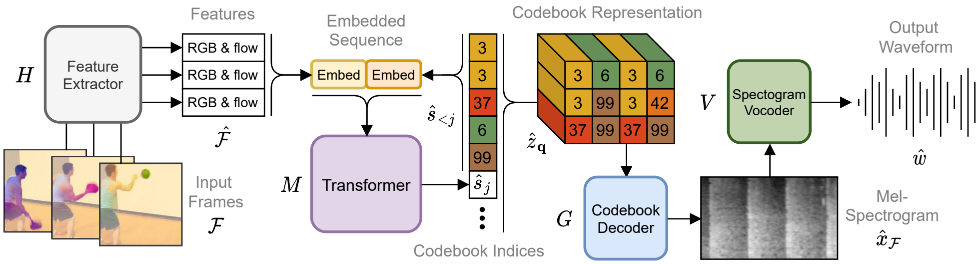


Figure 2: **Vision-based Conditional Cross-modal Autoregressive Sampler**. A transformer autoregressively samples the next codebook index given a sequence of visual features along with previously generated codebook indices. Once sampling is done, a sequence of generated indices is used to look up a pretrained codebook. Next, a pretrained codebook decoder is used to decode a spectrogram from a codebook representation. Finally, the generated spectrogram is turned into a waveform using a pretrained general-purpose spectrogram vocoder.

3.1 Perceptually-rich Spectrogram Codebook

The transformer requires the input to be represented as a sequence. A direct operation on wave samples or raw spectrogram pixels, however, quickly becomes intractable due to the quadratic nature of the dot-product attention. Alternatively, one could apply an encoder such as VQVAE [61] but the quantized bottleneck representation would be still infeasibly large. Our approach draws on VQGAN [17], an efficient autoencoder that allows decoding an image from a smaller-size representation than of VQVAE. To bridge the gap between image and audio signals, we operate on spectrograms and propose a new perceptual loss (LPAPS).

Spectrogram VQVAE Vector-Quantized Variational Autoencoder (VQVAE) [61] is trained to approximate an input using a compressed intermediate representation, retrieved from a discrete codebook. Our adaption of VQVAE, *Spectrogram VQVAE*, inputs a spectrogram $x \in \mathrm{R}^{F \times T}$ and outputs a reconstructed version of it $\hat{x} \in \mathrm{R}^{F \times T}$. First, the input x is encoded into a small-scale representation $\hat{z} = E(x) \in \mathrm{R}^{F' \times T' \times n_z}$ where n_z is the dimension of the codebook entries and $F' \times T'$ is a reduced frequency and time dimension. Next, the elements of the encoded representation $\hat{z}$ are mapped onto the closest items in a codebook $\mathcal{Z} = \{z_k\}_{k=1}^{K} \subset \mathrm{R}^{n_z}$, forming a quantized representation $z_{\mathbf{q}} \in \mathrm{R}^{F' \times T' \times n_z}$:

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\underset{z_k \in \mathcal{Z}}{\arg\min} ||\hat{z}_{ft} - z_k|| \quad \text{for all } (f,t) \text{ in } (F' \times T') \right). \tag{1}$$

Since (1) is non-differentiable, we approximate the gradient by a straight-though estimator [2]. The reconstructed spectrogram $\hat{x}$ is subsequently decoded from the codebook representation as $\hat{x} = G(z_{\mathbf{q}}) = G(\mathbf{q}(E(x)))$. The full VQVAE objective is defined by

$$\mathcal{L}_{\text{VQVAE}} = \underbrace{||x - \hat{x}||}_{\text{recons loss}} + \underbrace{||E(x) - \text{sg}[z_{\mathbf{q}}]||_2^2 + \beta ||\text{sg}[E(x)] - z_{\mathbf{q}}||_2^2}_{\text{codebook loss}} \tag{2}$$

where sg is the stop-gradient operation that acts as an identity during the forward pass but has zero gradient at the backward pass.

The resolution of the intermediate codebook representation $(F' \times T')$ produced by VQ-VAE remains to be too large for a transformer to operate on. However, more suitable down-sampling rates, *e.g.* $1/16$ of the input size, lead to poor reconstructions as shown in [17].

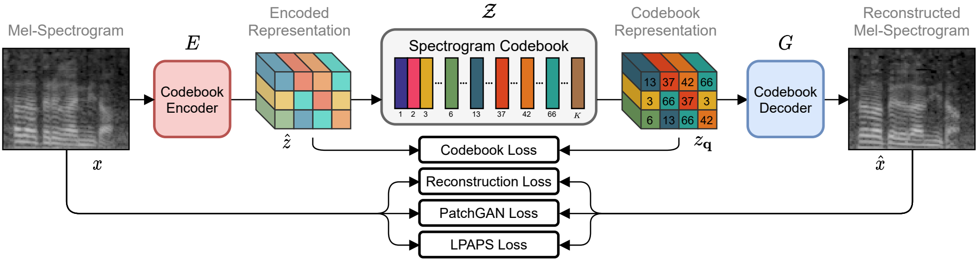


Figure 3: **Training Perceptually-Rich Spectrogram Codebook**. A spectrogram is passed through a 2D codebook encoder that effectively shrinks the spectrogram. Next, each element of a small-scale encoded representation is mapped to its closest neighbor from the codebook. A 2D codebook decoder is then used to reconstruct the input spectrogram. The training of the model is guided by codebook, reconstruction, adversarial, and LPAPS losses.

Spectrogram VQGAN and LPAPS VQGAN [17] is a version of VQVAE, extended with a patch-based adversarial loss [29] and perceptual loss (LPIPS) [68], that help to preserve the reconstruction quality when upsampled from a smaller-scale representation. Since the perceptual loss, used in the original VQGAN, relies on the ImageNet [14] pretrained VGG-16 [54], it is unreasonable to expect decent performance on sound spectrograms. Therefore, we introduce a novel way of guiding spectrogram-based audio synthesis, referred to as Learned Perceptual *Audio* Patch Similarity (LPAPS).

The closest relative of VGG-16 in audio classification is VGGish [26], which has the same capacity as VGG-9. However, we cannot directly build LPAPS on the pretrained VGGish or its architecture, since VGGish digests spectrograms with a rather short time span (<1 second), while our application requires operating on spectrograms spanning up to 10 seconds. Moreover, the lack of depth and, therefore, downsampling operations prevents the model from extracting larger-scale features that could be useful in separating real and fake spectrograms. To address this, we train a variant of the VGG-16 architecture on the VGGSound dataset [7]. We refer to the obtained model as VGGish-ish.

Fig. 3 shows the training procedure for Spectrogram VQGAN with the final loss:

$$\mathcal{L}_{\text{SpecVQGAN}} = \mathcal{L}_{\text{VQVAE}} + \underbrace{\log D(x) + \log(1 - D(\hat{x}))}_{\text{patch-based adversarial loss}} + \underbrace{\sum_s \frac{1}{F^s T^s} ||x^s - \hat{x}^s||_2^2}_{\text{LPAPS loss}}, \tag{3}$$

where D is a patch-based discriminator and x^s, $\hat{x}^s \in \mathrm{R}^{F^s \times T^s \times C^s}$ are features from real and fake spectrograms extracted at the s^{th} scale of VGGish-ish.

3.2 Vision-based Conditional Cross-modal Autoregressive Sampler

The sampler (transformer) is trained to sample a sequence of the codebook indices given a set of visual features. These should match the indices formed by the codebook encoder for the original audio. The conditional prediction of the next token can be formulated as a machine translation task and modeled by the vanilla Encoder-Decoder transformer architecture [62]. Alternatively, the problem can be defined in terms of language modeling, that is often approached with a Decoder-only transformer such as GPT [47]. In this paper, we employ a variant of GPT-2 [48] inspired by its success in autoregressive image synthesis [10, 17].

As outlined in Fig. 2, the sampling starts with the extraction of a sequence of features $\hat{\mathcal{F}} = \{\hat{f}_i\}_{i=1}^N \subset \mathrm{R}^{D_r + D_o}$ formed from a stack of RGB and optical flow frames $\mathcal{F} = \{f_i^r, f_i^o\}_{i=1}^N$.

The sequence of features $\hat{\mathcal{F}}$ is obtained by applying a frame-wise feature extractor H that consists of two pretrained models (for RGB and flow modalities) such that $\hat{\mathcal{F}} = H(\mathcal{F})$. Given a sequence of previously generated codebook indices $\hat{s}_{<j} = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{j-1})$ along with the features $\hat{\mathcal{F}}$, an autoregressive step for the transformer M is defined by

$$p\big(s_j|\hat{s}_{<j}, \hat{\mathcal{F}}\big) = M\big([\hat{\mathcal{F}} : \hat{s}_{<j}]\big), \tag{4}$$

where [:] is a stacking operation and $p\big(s_j|\hat{s}_{<j}, \hat{\mathcal{F}}\big) \in [0,1]^{n_z}$ is a probability distribution over all codebook indices. The next codebook index $\hat{s}_j$ is sampled from the multinomial distribution with weights provided by p. The sampling is initialized at $j = 1$ and primed only with the input features $\hat{\mathcal{F}}$. Once $j = F' \cdot T'$, the sampling stops. The sequence of predicted codebook indices $\hat{\mathcal{S}} = \{\hat{s}_j\}_{j=1}^{F' \cdot T'}$ is used to lookup the codebook $\mathcal{Z}$ so that, after unflattening, the codebook representation $\hat{z}_{\mathbf{q}} \in \mathrm{R}^{F' \times T' \times n_z}$ is formed. The transformer is trained with a typical cross-entropy loss, comparing the predicted codebook indices to those obtained from the ground truth spectrogram. Finally, given the codebook representation $\hat{z}_{\mathbf{q}}$, we decode a spectrogram $\hat{x}_{\mathcal{F}}$ using the decoder G pretrained during the codebook training stage (Sec. 3.1).

We note the importance of unflattening the sequence into a 2D form in a column-major way, precisely as shown in the middle part of Fig. 2, opposed to the row-major approach used for image synthesis [10, 17]. Employing the row-major unflatteting during training restricts model applications as it would correspond to reconstructing the lower frequencies given the higher ones. Specifically, we found that a model trained this way produces poor samples when prompted with a few seconds of real audio.

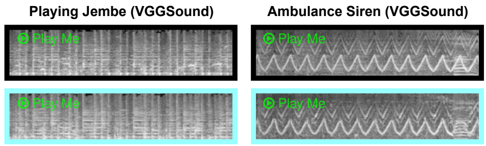
3.3 Spectrogram Vocoder

During the final stage, a waveform $\hat{w}$ is reconstructed from the decoded spectrogram using the pretrained vocoder V. Natural candidates for such vocoding are the Griffin-Lim algorithm [23] and WaveNet (used in prior work [11]). The Griffin-Lim procedure is fast, easy to implement, and it handles the diversity of an open-domain dataset. However, it produces low-fidelity results when operating on mel-spectrograms. In contrast, WaveNet provides high-quality results but remains to be relatively slow on test-time (25 mins per 10-sec sample on a **G**PU). For these reasons, we employ MelGAN [32] that is a non-autoregressive approach to reconstruct a waveform and, therefore, takes only 2 secs per sample on a **C**PU, while still achieving decent quality. Since MelGAN is originally trained for speech or music data, the pretrained models cannot be used in our open-domain scenario. Therefore, we train a MelGAN on the open-domain dataset (VGGSound).

3.4 Automatic Quality Assessment of Spectrogram-based Synthesis

Fidelity Our goal is to automatically evaluate both the fidelity and relevance of the generated samples. In the image generation domain, ImageNet pretrained InceptionV3 [56] is often used to form an opinion on the fidelity of the generated samples. Specifically, Inception Score [53] hypotheses low entropy in conditional label distribution and high entropy on a marginal probability distribution for high-fidelity and diverse samples. More consistent evaluation results were achieved by computing Fréchet Distance between the distributions of pre-classification layer's features of InceptionV3 between fake and real samples (FID) [27]. Considering the domain gap between spectrograms and RGB images, we adapt the Inception architecture for a spectrogram input size and train the model on the VGGSound dataset.

Trained on	Evaluated on	FID↓	$\overline{\text{MKL}}$↓
VGGSound	VGGSound	1.0	0.8
VGGSound	VAS	3.2	0.7
VAS	VAS	6.0	1.0



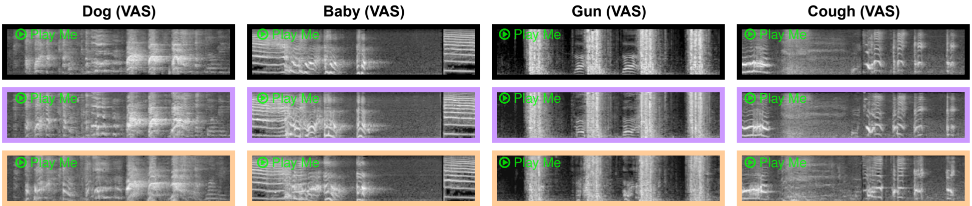


Table 1: **Spectrogram VQGAN shows strong reconstruction ability on hold-out sets of VGGSound and VAS**. Metrics are Melception-based FID and mean MKL. On the top-right: ground truth reconstruction results for two classes are shown for a model trained on VGGSound. The bottom triplets show a comparison of VGGSound-trained and VAS-trained models on four classes from VAS. Adobe Reader can be used to listen for reconstructions.

Relevance Since Inception Score and FID metrics rely on dataset-level distributions, they are not suitable to assess the conditional content synthesis. To this end, we propose a metric, called MKL, that individually compares the distances between output distributions of fake and real audio associated with a condition (*e.g.* frames from a video). As the distance measure, we rely on KL-divergence and use the Melception classifier to build the distributions.

4 Experiments

VGGSound and VAS Datasets VAS dataset [11] consists of 12.5k ~6.73-second clips for 8 classes: *Dog*, *Fireworks*, *Drum*, *Baby*, *Gun*, *Sneeze*, *Cough*, *Hammer*. We follow the same train-test splitting procedure as [11] for a fair comparison. VGGSound dataset [7] consists of ~200k+ 10-second clips from YouTube spanning 309 classes with audio-visual correspondence. The classes can be grouped as *people*, *sports*, *nature*, *home*, *tools*, *vehicles*, *music*, etc. VGGSound is substantially larger, but less curated than VAS due to the automatic collecting procedure. We managed to download ~190k clips from the dataset as some of the videos were removed from YouTube. Our split is similar to the original with the exception that the train part is further split into train and validation. The validation split is formed to match the same number of *videos* per class as in the test set. As a result, we have 156.5k *clips* in the train, 19.1k in the validation, and 14.5k in the test sets. This splitting strategy is used across all training procedures including Melception, MelGAN, and VGGish-ish. To the best of our knowledge, we are the first to use the VGGSound dataset for sound synthesis.

Metrics The proposed model is evaluated in quantitative and qualitative studies. In quantitative evaluation, we rely on Melception-based metrics, namely MKL (averaged across the dataset) and FID for relevance and fidelity evaluation (as defined in Sec. 3.4).

Details We extract log mel-spectrograms of size 80×848 and 212 visual features with dimension $D_r = D_o = 1024$ from ~9.8-second videos before training. The codebook encoder and decoder are generic 2D Conv stacks with two extra attention layers before $\hat{z}$ and after $z_{\mathbf{q}}$. The downsampling and upsampling operations are parametrized. The variant of GPT-2 has 24 layers. Visual features and codebook indices are embedded to match the transformer dimension (1024). Training requires at least one 12GB GPU. See more in the supplementary.

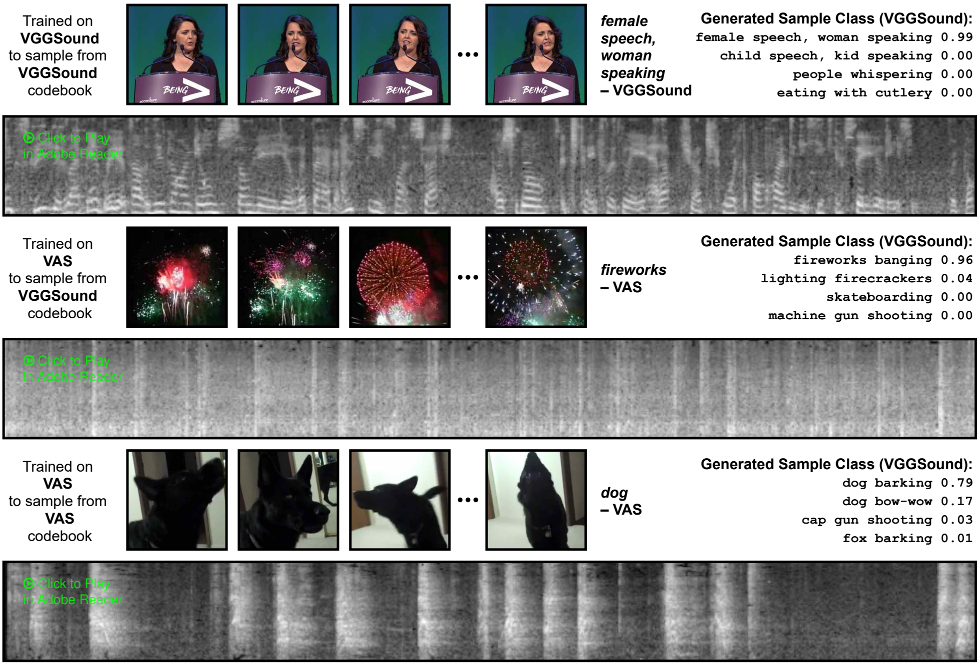


Figure 4: **Samples produced by conditional cross-modal sampler are relevant and have high fidelity.** The top row shows results of a model trained on VGGSound to sample from a VGGSound codebook ("from VGGSound for VGGSound"), the middle is "from VGGSound for VAS", the bottom is: "from VAS to VAS". An "opinion" of Melception is on the right.

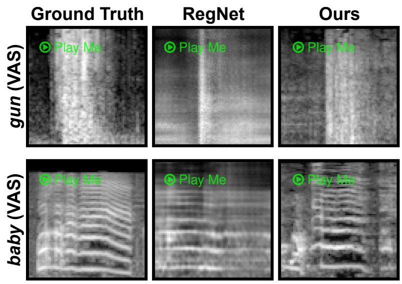
4.1 Results

Reconstruction with Spectrogram VQGAN When compared to ground truth spectrograms, the reconstructions are expected to have high fidelity (low FID) and to be relevant (low mean MKL). Tab. 1 contains quantitative and qualitative results produced by our Spectrogram VQGAN (Sec. 3.1). The results imply high fidelity and relevance on a variety of classes from both VGGSound (test) and VAS (validation) datasets. Notably, the performance of the VGGSound-pretrained codebook is better than of the VAS-pretrained codebook even when applied on the VAS validation set due to larger and more diverse data seen during training. The implementation details and more examples are provided in the Supplementary. Moreover, in Tab. 2 we show the results of the ablation study on the impact of losses on reconstruction quality. In particular, the absence of the adversarial loss results in significant blurriness (which agrees with the findings in [17]) in reconstructed spectrograms and expected substantial downgrade in metrics.

Visually-Guided Sound Generation We benchmark the visually-guided sound generation qualitatively and quantitatively using three different settings: **a)** trained the transformer on *VGGSound* to sample from the *VGGSound* codebook, **b)** trained on *VAS* with the *VGGSound* codebook, and **c)** trained on *VAS* with the *VAS* codebook. Fig. 4 shows a few examples obtained with different settings along with the "opinion" of the Melception classifier on the generated sample label and in Tab. 3, we compare a different number of priming features including sampling without a condition (*No Feats*), which can be seen as the upper-bound on the relevance metric (mean MKL). The quantitative results are provided for two sets of ImageNet-pretrained features: BN-Inception (RGB + flow) and ResNet-50 (RGB).

GAN	LPAPS	FID↓	$\overline{\text{MKL}}$↓
		130.4	9.6
✔		1.4	1.1
✔	✔	1.0	0.8

Table 2: Adversarial and perceptual losses improve reconstruction results on the VGGSound test set.



	Condition	FID↓	$\overline{\text{MKL}}$↓	FID↓	$\overline{\text{MKL}}$↓	FID↓	$\overline{\text{MKL}}$↓	⏱↓
	No Feats	13.5	9.7	33.7	9.6	28.7	9.2	7.7
ResNet	1 Feat	11.5	7.3	26.5	6.7	25.1	6.3	7.7
	5 Feats	11.3	7.0	22.3	6.5	20.9	6.1	7.9
	212 Feats	10.5	6.9	20.8	6.2	22.6	5.8	11.8
Inception	1 Feat	8.6	7.7	38.6	7.3	25.1	6.6	7.7
	5 Feats	9.4	7.0	29.1	6.9	24.8	6.2	7.9
	212 Feats	9.6	6.8	20.5	6.0	25.4	5.9	11.8
	Codebook	VGGSound		VGGSound		VAS		
	Sampling for	VGGSound		VAS		VAS		
	Setting	**(a)**		**(b)**		**(c)**		

Table 3: **The number of features is an important factor for relevance and sampling speed on both datasets**. Fidelity and relevance are measured by FID and mean MKL, speed is in seconds to generate a ∼10-second audio sample.

	Params	FID↓	$\overline{\text{MKL}}$↓	⏱↓
Ours **(b)**	379M	20.5	6.0	12
Ours **(c)**	377M	25.4	5.9	12
RegNet [□]	8 × 105M	78.8	5.7	1500
Ours **(b)** + cls	379M	20.2	5.7	12
Ours **(c)** + cls	377M	24.9	5.5	12

Table 4: **Compared to state-of-the-art, our model generates higher fidelity samples faster and with similar relevance w/ and w/o providing the class label**. RegNet size is multiplied by the num. of classes in VAS.

We observe that: 1) In general, the more features from a corresponding video are used, the better the result in terms of relevance. However, there is a trade-off imposed by the sampling speed which decreases with the size of the conditioning. 2) A large gap (log-scale) in mean MKL between visual and "empty" conditioning suggests the importance of visual conditioning in producing relevant samples. 3) When the sampler and codebook are trained on the same dataset—settings (a) and (c)—the fidelity remains on a similar level if visual conditioning is used. This suggests that it is easier for the model to learn "features-codebook" (visual-audio) correspondence even from just a few features. However, if trained on different datasets (b), the sampler benefits from more visual information. 4) Both BN-Inception and ResNet-50 features achieve comparable performance, with BN-Inception being slightly better on VGGSound and with longer conditioning in each setting. Notably, the ResNet-50 features are RGB-only which significantly eases practical applications. We attribute the small difference between the RGB+flow features and RGB-only features to the fact that ResNet-50 is a stronger architecture than BN-Inception on the ImageNet benchmark [3]. See the technical details, more examples, ablations, and human studies in Supplementary Material.

Comparison with the state-of-the-art In Tab. 4, we compare our model to RegNet [□], which is currently the strongest baseline in generating relevant sounds for a visual sequence. For a fair comparison, we employ the same data preprocessing for audio and visual features as in RegNet [□]. We use the settings (b) & (c) (see Tab. 3) with 212 features in the condition, which is similar to the RegNet input. Since RegNet limits the sampling space explicitly by training a separate model for each class, it is difficult to fairly compare relevance with our model that is trained on all classes. To mitigate this to some extent, we include a class label into the transformer conditioning sequence allowing the model to learn to separate parameter subspaces for all 8 classes. The results suggest that our model produces higher quality spectrograms than RegNet, which is also supported by the lower FID scores. Moreover, RegNet uses two times more parameters. See more examples in the Supplementary Material.

4.2 Qualitative Analysis of the Model Properties

We conduct a human study by single-handedly inspecting over 2k samples for test-set videos of the VGGSound dataset. Despite the biasedness of the study, we believe that the results are worth reporting. The samples are drawn for a random class and using the model trained on the VGGSound dataset with the VGGSound codebook (the setting (a), *5 Feats*, see Sec. 4.1). We divide our observations into three parts: **general properties of the model**, **problems with data preprocessing**, and **dataset-related issues** (see supplementary).

General Properties of the Model The proposed model supports multiple classes and, especially with some patience budget, generates relevant audio for the majority of classes in the VGGSound. The mistakes are not rare, but they are often associated with a poor audio-visual correspondence in the video or because the model generates a sound of another musical instrument instead of the specific one (*e.g.*, *violin* instead of *cello* – both are string instruments). However, the generation of a sample that belongs to a completely different class group is a rare event, *e.g.*, for a bird singing video the model will not generate an audio appropriate for indoor sports activities. We also observed, for classes such as *zebra braying*, *cat purring*, *pig oinking*, *bee, wasp, etc. buzzing*, *cattle mooing*, *alarm clock ringing*, the model struggles to produce a relevant sample possibly due to the unobservable source of the signal (*e.g.*, the flies are flying around the camera pointed to a tree and the flies are never captured but heard).

The model may confuse visually similar sounds, *e.g.*, *people whistling*, *singing*, *talking*, *whispering*, *burping*, etc. Also, if a video shows a close-up of hands, *e.g.*, *machine sewing*, the model may generate a sound of *keyboard typing* or *computer mouse clicking*. We also found that an ASMR setup (Autonomous Sensory Meridian Response) enforces the model to produce clean sounds similar to ASMR but often of a different class. The model struggles to differentiate different types of birds (*e.g.*, *swallow chickadee*, *pheasant*, etc) or hitting instruments (*e.g.*, *bongo*, *timbales*, *timpani*, *steelpan*, etc), yet it tends to produce the sounds of a similar class from, *e.g.*, another bird or instrument. These properties are expected from a model trained on a relatively noisy dataset with a vague separation between classes.

Data Preprocessing Issues After transformation into the mel-scale spectrogram, the audio signal loses the phase and a range of essential frequencies to differentiate sounds from some classes. For instance, by transforming the waveform into mel-scale spectrogram and back, we observed that the sound of *cat caterwauling* became indiscernible from *person sobbing*, *crying*, or *dog howling* classes. Although the speech segments are recognizable, the words are indecipherable. To this end, the model can be trained directly on top of the STFT spectrograms at the cost of efficiency during sampling, however.

5 Conclusion

We introduced a new efficient approach for multi-class visually-guided sound generation, which operates on spectrograms and relies on a prior in a form of a codebook representation. To train the prior, we proposed a new perceptual loss (LPAPS) which is based on a general-purpose classifier (VGGish-ish). This loss allows the model to learn to reconstruct higher-fidelity spectrograms from a small-scale representation. In addition, a novel automatic evaluation procedure is outlined to estimate both fidelity and relevance of generated spectrograms with a new family of metrics based on the Melception classifier. Our experiments on small- and large-scale datasets show the power and efficiency of our model in both quantitative and qualitative studies compared to the state-of-the-art.

Acknowledgments Funding for this research was provided by the Academy of Finland projects 327910 & 324346. We also acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

[1] Théis Bazin, Gaëtan Hadjeres, Philippe Esling, and Mikhail Malt. Spectrogram Inpainting for Interactive Generation of Instrument Sounds. In *Joint Conference on AI Music Creativity*, 2020.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[3] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 2018.

[4] Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

[5] Mikolaj Binkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. High Fidelity Speech Synthesis with Adversarial Networks. In *ICLR*, 2020.

[6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody Dance Now. In *ICCV*, 2019.

[7] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A Large-Scale Audio-Visual Dataset. In *ICASSP*, 2020.

[8] Kan Chen, Chuanxi Zhang, Chen Fang, Zhaowen Wang, Trung Bui, and Ram Nevatia. Visually indicated sound generation by perceptually optimized classification. In *ECCV Workshops*, 2018.

[9] Lele Chen, Sudhanshu Srivastava, Zhiyao Duan, and Chenliang Xu. Deep cross-modal audio-visual generation. In *Thematic Workshops of ACM Multimedia*, 2017.

[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining From Pixels. In *ICML*, 2020.

[11] Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. Generating visually aligned sound from videos. *IEEE Transactions on Image Processing*, 2020.

[12] Yang Chen, Yingwei Pan, Ting Yao, Xinmei Tian, and Tao Mei. Mocycle-GAN: Unpaired Video-to-Video Translation. In *ACM International Conference on Multimedia*, 2019.

[13] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *CVPR*, 2018.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[15] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[16] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. CogView: Mastering Text-to-Image Generation via Transformers. In *NeurIPS*, 2021.

[17] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

[18] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *ECCV*. Springer, 2020.

[19] Chuang Gan, Deng Huang, Hang Zhao, Joshua B. Tenenbaum, and Antonio Torralba. Music Gesture for Visual Sound Separation. In *CVPR*, 2020.

[20] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NeurIPS*, 2014.

[22] Gal Greshler, Tamar Rott Shaham, and Tomer Michaeli. Catch-A-Waveform: Learning to Generate Audio from a Single Short Example. In *NeurIPS*, 2021.

[23] Daniel Griffin and Jae Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 1984.

[24] Wang-Li Hao, Zhaoxiang Zhang, and He Guan. CMCGAN: A uniform framework for cross-modal visual-audio mutual generation. In *AAAI*, 2018.

[25] Zekun Hao, Xun Huang, and Serge J. Belongie. Controllable Video Generation With Sparse Trajectories. In *CVPR*, 2018.

[26] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN architectures for large-scale audio classification. In *ICASSP*, 2017.

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[29] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*, 2017.

[30] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*, 2012.

[32] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C. Courville. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. In *NeurIPS*, 2019.

[33] Vinod K Kurmi, Vipul Bajaj, Badri N Patro, KS Venkatesh, Vinay P Namboodiri, and Preethi Jyothi. Collaborative Learning to Generate Audio-Video Jointly. In *ICASSP*. IEEE, 2021.

[34] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[35] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *CVPR*, 2020.

[36] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia*, 2018.

[37] Xubo Liu, Turab Iqbal, Jinzheng Zhao, Qiushi Huang, Mark D Plumbley, and Wenwu Wang. Conditional Sound Generation Using Neural Discrete Time-Frequency Representation Learning. In *MLSP Workshop*, 2021.

[38] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In *ECCV*, 2020.

[39] Pranay Manocha, Adam Finkelstein, Richard Zhang, Nicholas J. Bryan, Gautham J. Mysore, and Zeyu Jin. A differentiable perceptual audio metric learned from just noticeable differences. In *Interspeech*, October 2020.

[40] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio. SampleRNN: An Unconditional End-to-End Neural Audio Generation Model. In *ICLR*, 2017.

[41] Takeru Miyato and Masanori Koyama. cGANs with Projection Discriminator. In *ICLR*, 2018.

[42] Medhini Narasimhan, Shiry Ginosar, Andrew Owens, Alexei A Efros, and Trevor Darrell. Strumming to the Beat: Audio-Conditioned Contrastive Video Textures. In *CVPR Workshops*, 2021.

[43] Javier Nistal, Stefan Lattner, and Gael Richard. Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks. In *ISMIR*, 2020.

[44] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *ICML*, 2017.

[45] Andrew Owens, Phillip Isola, Josh H. McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. Visually Indicated Sounds. In *CVPR*, 2016.

[46] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven Manipulation of Stylegan Imagery. In *ICCV*, 2021.

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[48] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

[49] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *VISIGRAPP*, 2021.

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *ICML*, 2021.

[51] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. In *NeurIPS*, 2019.

[52] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative Adversarial Text to Image Synthesis. In *ICML*, JMLR Workshop and Conference Proceedings, 2016.

[53] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs. In *NeurIPS*, 2016.

[54] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[55] Kun Su, Xiulong Liu, and Eli Shlizerman. Audeo: Audio Generation for a Silent Performance Video. In *NeurIPS*, 2020.

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, 2016.

[57] Huadong Tan, Guang Wu, Pengcheng Zhao, and Yanxiang Chen. Spectrogram Analysis Via Self-Attention for Realizing Cross-Model Visual-Audio Generation. In *ICASSP*, 2020.

[58] Maciej Tomczak, Masataka Goto, and Jason Hockman. Drum Synthesis and Rhythmic Transformation with Adversarial Autoencoders. In *ACM International Conference on Multimedia*, 2020.

[59] Soumya Tripathy, Juho Kannala, and Esa Rahtu. FACEGAN: Facial Attribute Controllable rEenactment GAN. In *WACV*, 2021.

[60] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing Motion and Content for Video Generation. In *CVPR*, 2018.

[61] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *NeurIPS*, 2017.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017.

[63] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-Video Synthesis. In *NeurIPS*, 2018.

[64] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks. In *CVPR*, 2018.

[65] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video Generation using VQ-VAE and Transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[66] Han Zhang, Tao Xu, and Hongsheng Li. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *ICCV*, 2017.

[67] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. In *ICML*, 2019.

[68] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018.

[69] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018.

[70] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The Sound of Motions. In *ICCV*, 2019.

[71] Yi Zhao, Haoyu Li, Cheng-I Lai, Jennifer Williams, Erica Cooper, and Junichi Yamagishi. Improved Prosody from Learned F0 Codebook Representations for VQ-VAE Speech Waveform Reconstruction. In *Interspeech*, 2020.

[72] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qi-Xing Huang, and Alexei A. Efros. Learning Dense Correspondence via 3D-Guided Cycle Consistency. In *CVPR*, 2016.

[73] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L. Berg. Visual to Sound: Generating Natural Sound for Videos in the Wild. In *CVPR*, 2018.

[74] Lingyu Zhu and Esa Rahtu. Visually guided sound source separation using cascaded opponent filter network. In *ACCV*, 2020.