Taming Visually Guided Sound Generation (WIP)

Vladimir Iashin Esa Rahtu

Tampere University

Goal

To generate visually relevant and high-fidelity sounds

Challenges

- Generation of long sounds
- Many video categories (10s or 100s)
- Generation in real-time
- Lack of human-free evaluation procedures for audio synthesis

Contributions

Model for controlled sound generation based on visual cues
- supports multiple data classes
- generates the sound faster that it takes to play it

Perceptual loss for spectrogram-based sound synthesis
- designed for the open-domain spectrogram generation
- helps VQVAE to reconstruct input from a smaller bottleneck size

Family of metrics for conditional sound generation
- evaluates *relevance* and *fidelity*
- supports evaluation of general-purpose spectrogram generative models

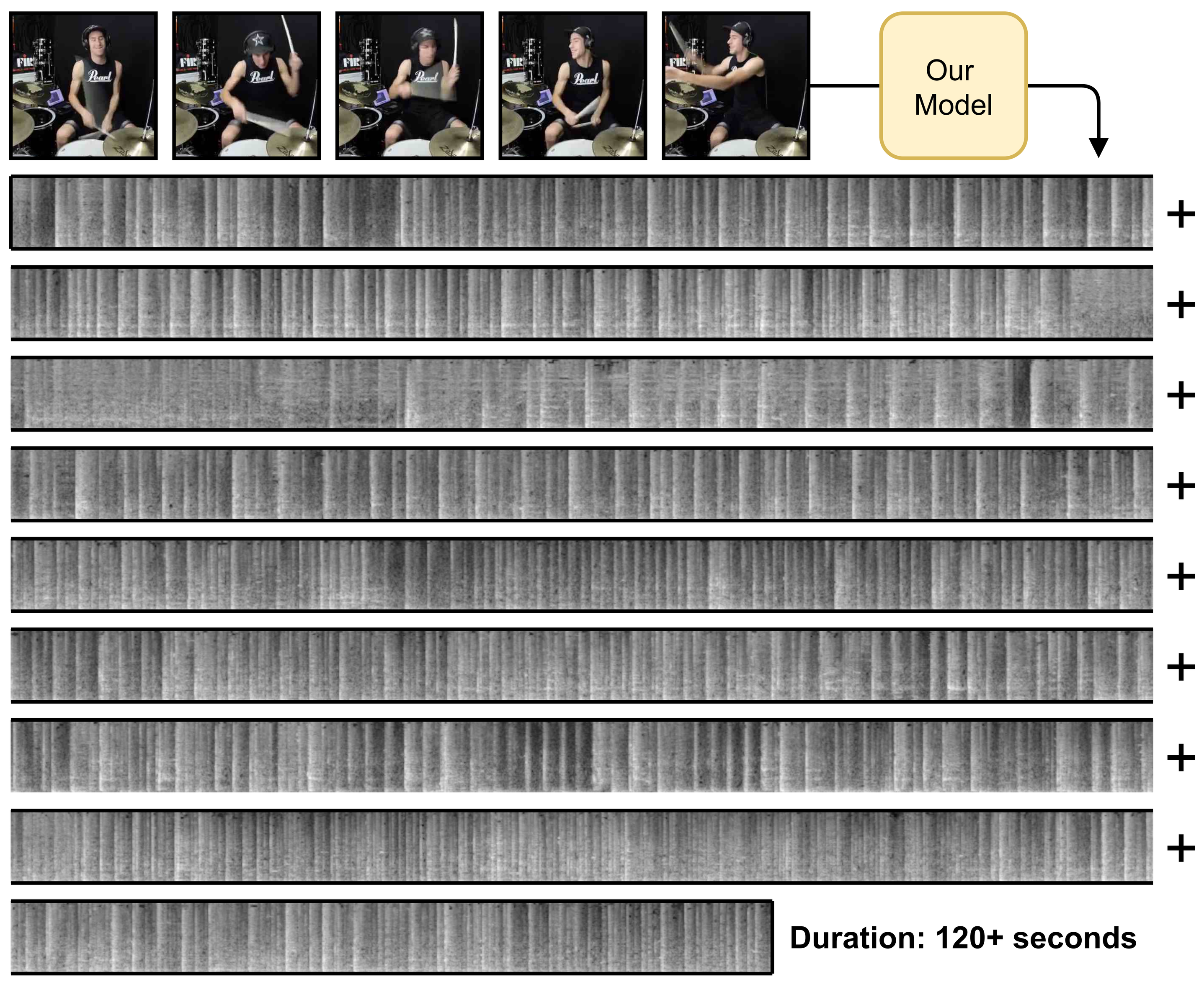
Datasets

Requirement: strong audio-visual correspondence

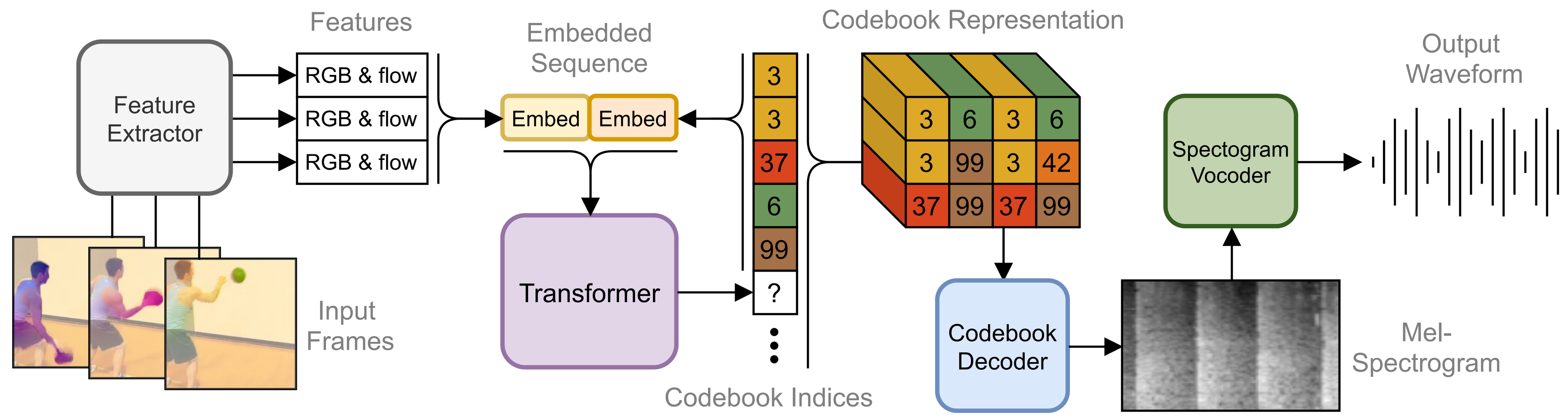
VAS
- Human-curated
- ~12.5k <10-second clips
- 8 classes: *Dog, Fireworks, Drum, Baby, Gun, Sneeze, Cough, Hammer*

VGGSound
- Automatically collected
- ~190k 10-second clips from YouTube
- 300+ classes grouped as: *people, sports, nature, home, tools, vehicles, music*, etc.

The Longest and Greatest Generated Drum Solo You've Seen *(maybe)*

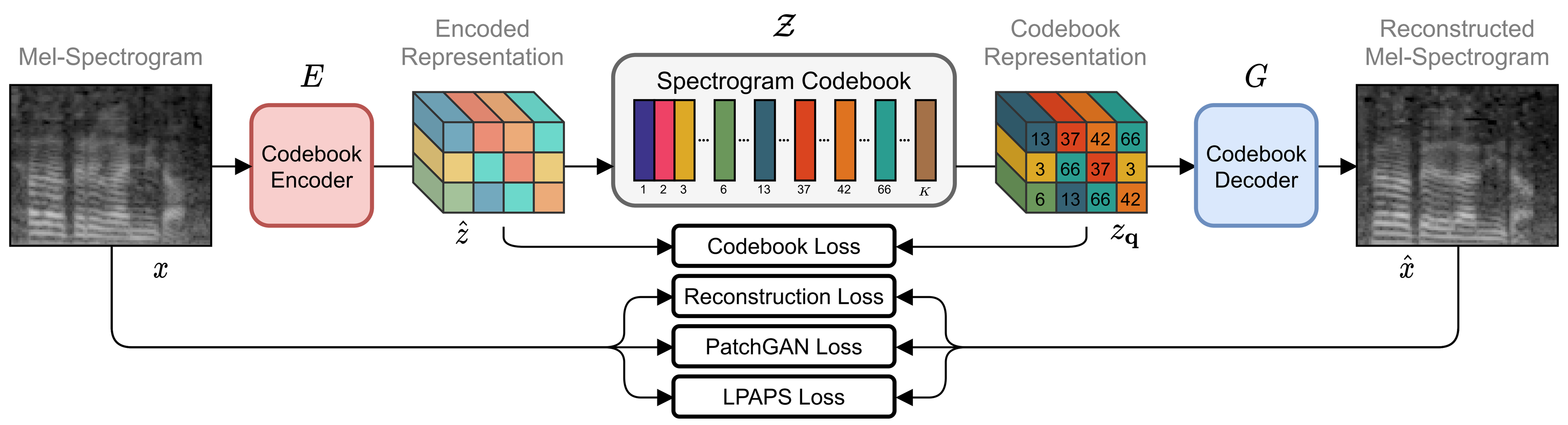


Overview



1. Primed with a set of visual features, the transformer samples indices to a codebook
2. The indices are replaced with the items from the codebook
3. The codebook representation is decoded into the spectrogram
4. The spectrogram is vocoded into a waveform

Spectrogram Codebook Pre-training



Spectrogram Codebook is trained on spectrograms from the VGGSound dataset using the following loss

$$\mathcal{L} = \underbrace{||\text{sg}[E(x)] - z_{\mathbf{q}}||_2^2 + \beta||E(x) - \text{sg}[z_{\mathbf{q}}]||_2^2}_{\text{codebook loss}} + \underbrace{||x - \hat{x}||}_{\text{recons loss}} + \underbrace{\log D(x) + \log(1 - D(\hat{x}))}_{\text{patch-based adversarial loss}} + \underbrace{\sum_s \frac{1}{F^s T^s}||x^s - \hat{x}^s||_2^2}_{\text{LPAPS loss}}$$

Learned Perceptual *Audio* Patch Similarity (LPAPS) with VGGish-*ish*

We train a VGG16 spectrogram classifier on VGGSound (300+ classes), we call it VGGish-ish.
LPAPS is defined a distance in feature space between generated and real spectrograms (see above).

Window-based Spectrogram Vocoder

- **Goal** Vocoder reconstructs a waveform from a spectrogram
- **Solution 1** The Griffin-Lim algorithm that is fast and can handle open-domain samples
- **Problem 1** Low quality of reconstruction from mel-spectrograms due to the intermediate algorithm
- **Solution 2** WaveNet produces high-fidelity samples
- **Problem 2** It is relatively slow (25 mins per 10-second sample on a GPU)
- **Solution 3** To train MelGAN from scratch on VGGSound (1 sec per high-quality 10-second sample on a CPU)

Evaluating Conditional Sound Generation

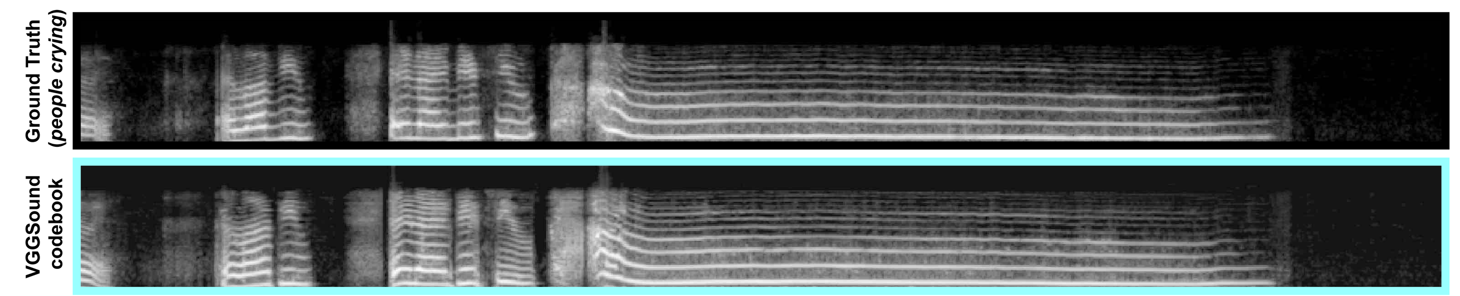
We train a variant of InceptionV3 on VGGSound dataset from scratch and call it Melception.

Melception is used in evaluation of

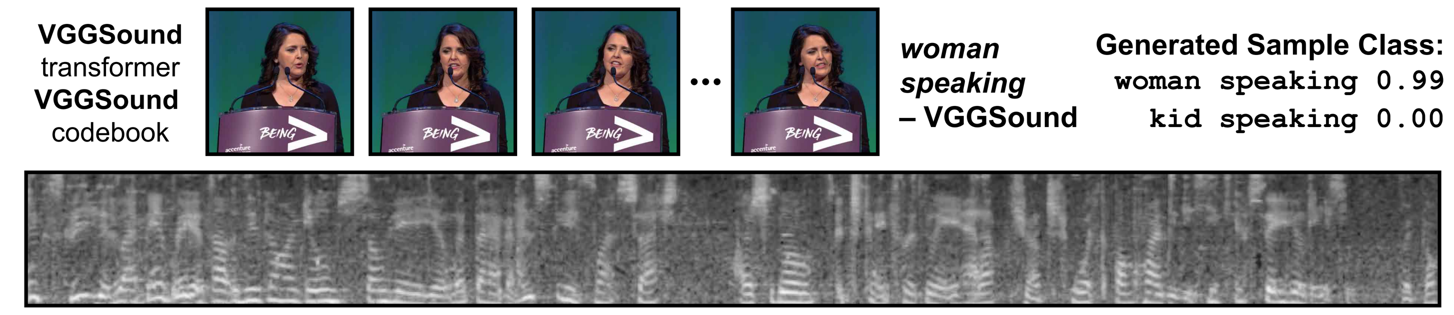
- **Fidelity** in a form of Inception Score, Fréchet- and Kernel Inception Distances
- **Relevance** as an individual distance between class distributions of fake and real audios associated with a condition

Results on VGGSound

Codebook Reconstruction



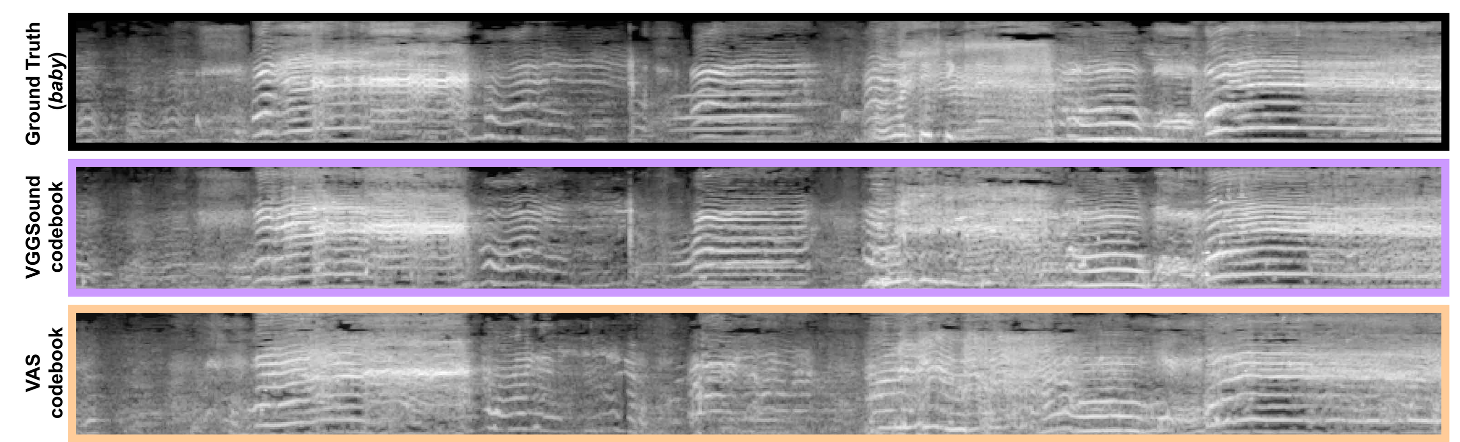
Visually-Guided Sound Generation



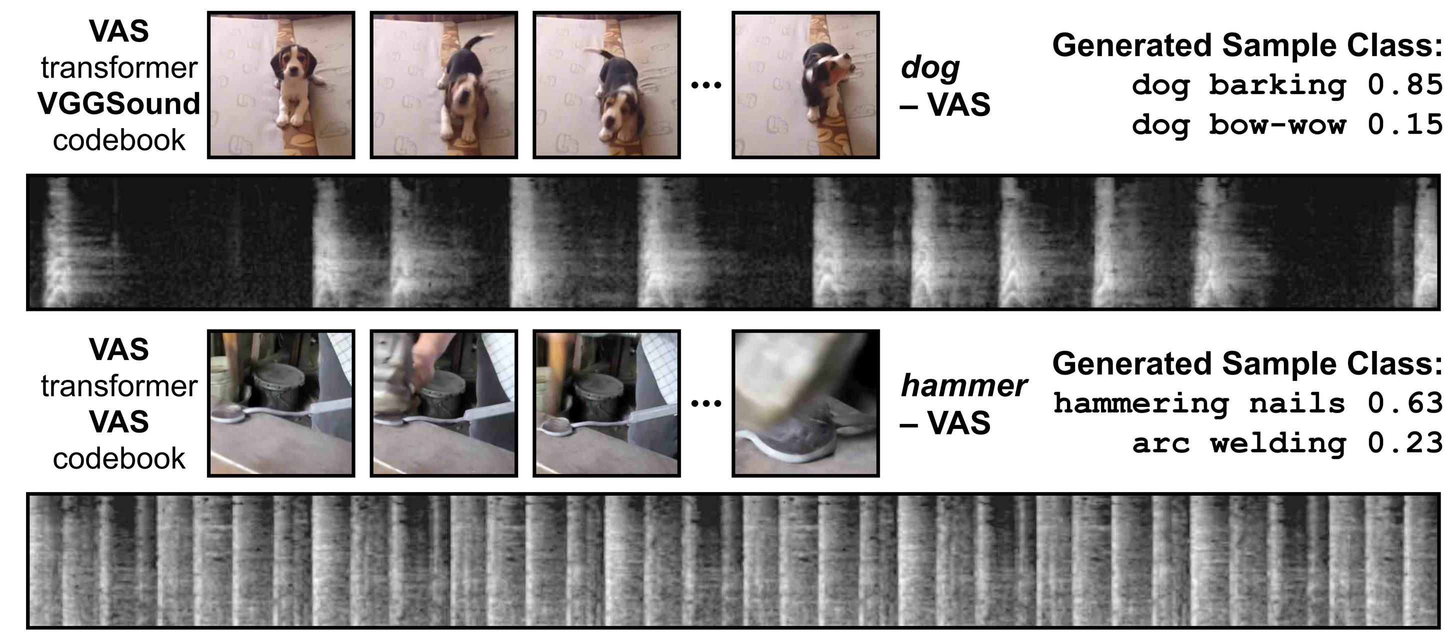
We are the first to apply VGGSound on sound generation, to the best of our knowledge

Results on VAS

Codebook Reconstruction



Visually-Guided Sound Generation



Comparison to State-of-the-art

RegNet supports only one class at once while Ours supports all 8 classes.

	Params	FID↓	$\overline{\text{MKL}}$↓
RegNet [1]	8×105M	78.8	5.7
Ours	377M	25.4	5.9
Ours + cls	377M	24.9	5.5

All models use the same set of visual feats.
[1] Chen *et. al*, in *IEEE TIP*, 2020.