

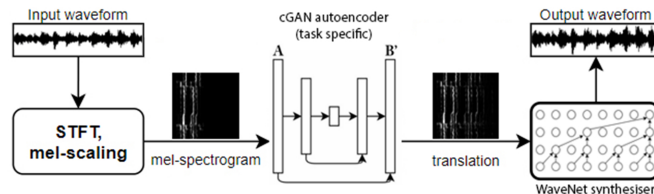
A UNIFIED NEURAL ARCHITECTURE FOR INSTRUMENTAL AUDIO TASKS

Steven Spratley, Daniel Beck, and Trevor Cohn
School of Computing and Information Systems, The University of Melbourne

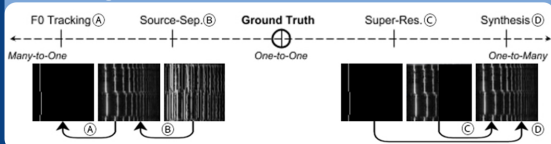
Abstract

► Within Music Information Retrieval (MIR), prominent tasks — including **pitch-tracking, source-separation, super-resolution, and synthesis** — typically call for specialised methods, despite their similarities. Conditional Generative Adversarial Networks (cGANs) have been shown to be highly versatile in learning general image-to-image translations, but have not yet been adapted across MIR. In this work, we present an end-to-end supervisable architecture to perform all aforementioned audio tasks, consisting of a **WaveNet synthesiser** conditioned on the output of a **jointly-trained cGAN spectrogram translator**. In doing so, we demonstrate the potential of such flexible techniques to unify MIR tasks, promote efficient transfer learning, and converge research to the improvement of powerful, general methods. Finally, to the best of our knowledge, we present the first application of GANs to guided instrument synthesis.

Overview of the GAN-WN Architecture



Framing MIR Tasks as Harmonic Distillation / Addition



Data and Representation

► We modelled the solo violin for three reasons. One, forming ‘deep’ models for each task was infeasible with our resources. Two, the violin is notoriously hard to model due to its expressive range, serving as a proof-of-concept for other instruments. Three, we expected it to facilitate joint modelling of tasks.

► For F_0 -tracking and synthesis tasks, we created **paired sine-wave and violin audio tracks** using software instruments and data from the Bach10 dataset. For super-resolution, we **quarter-sampled over 12 hours of live violin recordings** to generate the lofi track. For source-separation, we used **multi-track recordings** from the Bach10, Freischutz, and Phenix-anechoic datasets, and created our own synthetic data using MIDI files of Bach’s *Four Orchestral Suites* played through software instruments.

► Tracks were 16 kHz as standard, as the spectrum becomes increasingly sparse in frequencies captured by higher rates (i.e. expensive diminishing returns). Once finalised, tracks were compressed and normalised, and processed via the short-time Fourier transform (STFT). We **mel-scaled** all data to model for human frequency perception, and reclaim memory for larger kernels and layers.

Translation and Reconstruction

► Our method extends *pix2pix* in order to fit a translation model to each of our datasets of paired spectrograms. In each case, once generator G is properly trained, testing becomes a matter of converting audio of arbitrary length to its mel-spectrogram representation and applying G convolutionally.

► We also trained a **joint model** by increasing the input channels of our cGAN, and noticed **increased performance in fewer iterations**, suggesting that kernels were shared efficiently between tasks.

► We reconstructed spectrograms in the following 3 ways:

- ◆ **GAN-V**: Griffin-Lim. Displayed noticeable artifacts due to mel-scaling compression.
- ◆ **GAN-S** and **S2**: Secondary cGAN/s & Griffin Lim, improving rescaling from mel to linear.
- ◆ **GAN-WN**: WaveNet locally-conditioned on spectrograms, circumventing lossy rescaling.

► We trained one WaveNet model on the task that had the most available data (super-resolution). By partitioning into equal test and train (~6 hours of audio each), our cGAN produced 6 hours of spectrograms from unseen data, which were then paired with their ground truth audio in order to train WaveNet. In doing so, we train GAN-WN as a cascade architecture, making the reconstruction stage more robust to translation errors. Memory resources for WaveNet limited training instances to ~30k steps, or 1.9s in duration, which suited the 1.6s chunk size represented by our datasets.

Results

► We evaluated on the bases of **spectrogram distance as well as human audition**. We report both % error (normalised L1) as well as structural similarity (SSIM). We also hosted an **APE-style audition test**, tasking participants to audition sets of clips and assess them against both a Likert scale, and each other.

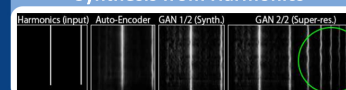
► Our baselines were as follows: Pitch-tracking was compared with *Ableton Live 9’s* audio-to-MIDI function, source-separation with non-negative matrix factorisation (NMF) and pre-trained CNN, super-resolution with linear and cubic interpolation, and synthesis purely by ablation. There are more recent baselines for many of these tasks; note that we don’t aim to overtake states-of-the-art, rather, our goal is to sufficiently demonstrate our approach’s *overall* capability.

► We report competitive performance with our chosen baselines, as shown below with spectrograms as a visual guide to procedures and outcomes. **We believe our findings advocate for further research towards generalisable methods such as ours, given their flexibility**; a breakthrough to such an architecture could mean wide-reaching effects to multiple tasks in MIR.

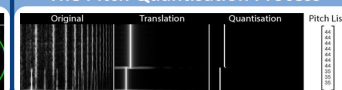
Norm. L1 distance and SSIM Results

% error	Source-Separation			Super-Resolution			F_0 Tracking Results				
	cGAN	NMF	CNN	cGAN	Cubic	Linear	Method	Correct	Total	Precision	Mean error
SSIM	0.52	0.50	0.47	0.68	0.53	0.51	Ableton	109 / 130	157	69.43%	32.69
							cGAN	119 / 130	122	97.54%	17.35

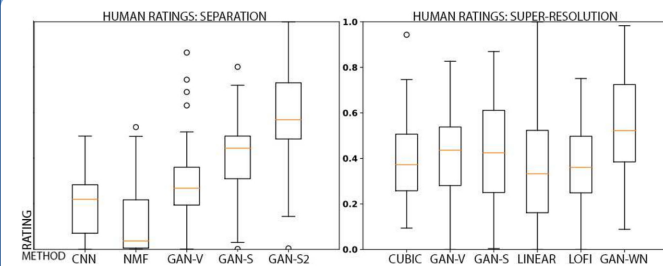
Synthesis from Harmonics



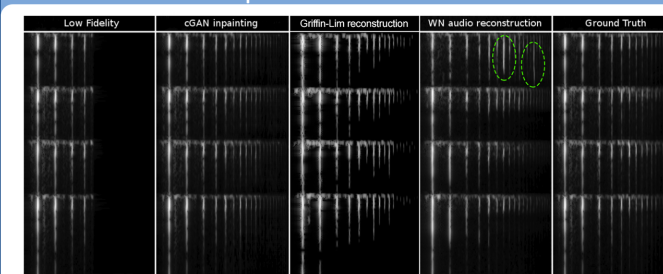
The Pitch Quantisation Process



Outcomes of Subjective Evaluation

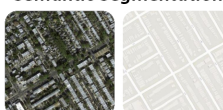


Visual Comparison of Reconstruction Methods



A **B'** :: **X** **Y'**

Semantic Segmentation



Denoising






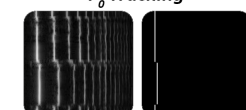
Image Inpainting



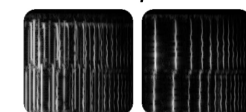
Style Transfer



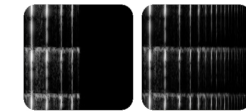
F_0 Tracking



Source Separation



Super-resolution



Synthesis

