# Supplementary Material: SyncNet: Using Causal Convolutions and Correlating Objective for Time Delay Estimation in Audio Signals

*Akshay Raina, Vipul Arora*

SMVDU and IITK

## 1. Data Acquisition

As realised from section 2.1, SyncNet has been evaluated on a diverse range of audio datasets. For the first dataset, one audio file with the sound of 'Tic' uttered by a human at periodic intervals was prepared. It was played on different mobile phones with a variety of background acoustic conditions. Total 170 audio files were recorded with natural environmental factors used as background noise for the generated reference signal. These factors included recorded sounds of birds chirping, wind, people conversating, etc. Furthermore, to increase the number of examples in the MTic dataset, normal noise with SNR chosen at random in the range $[-15dB, 20dB]$ was injected in 450 copies of the reference signal.

Next, 50 randomly chosen audio files from the LibriSpeech Dataset were set as different reference signals. As done with MTic, 25 noisy audio signals were synthesized for each of these 50 signals. Similar procedure was followed for audio files containing music beats selected from the MIREX 2012 dataset. All the noisy audio files are delayed from their corresponding reference signals by randomly chosen duration within $[0s, 0.9s]$.

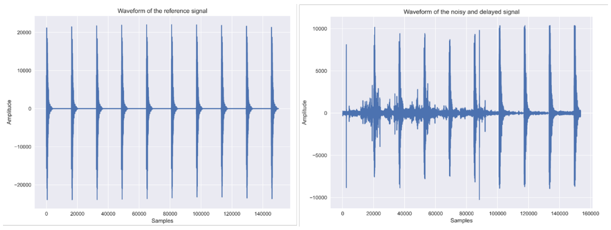

Figure 1: *Waveforms of the reference and one of the noisy and delayed audio files in the MTic Dataset*

## 2. Causal Convolution and Hybrid Loss

It can be well argued that the overlapping of windows for causal convolution leads to an increment in the size of the resulting sequence, relative to the case with consecutive/no-overlap windows. This counters the reduction in the size due to convolution operation by each layer in each tower. For a better picture, this can be extended to a special case, when the increment almost nullifies the reduction in size. Such a situation exists in case the following relation among the layers' hyper-parameters and $\delta$ holds, given the stride and padding are set to 1 and 0 respectively for all layers in each tower.

$$\delta = \sum_{i=1}^{h} d_i(f_i - 1)\delta = -h - 2\sum_{i=1}^{h}(p_i - \frac{f_i}{2}) = -h + \sum_{i=1}^{h} f_i$$

The shift in the sign of the slope of plots in figure 2 also represent this behaviour. In equation 1, $l$ is only *almost* equal to the term on the right side. A more general form of this equation is-

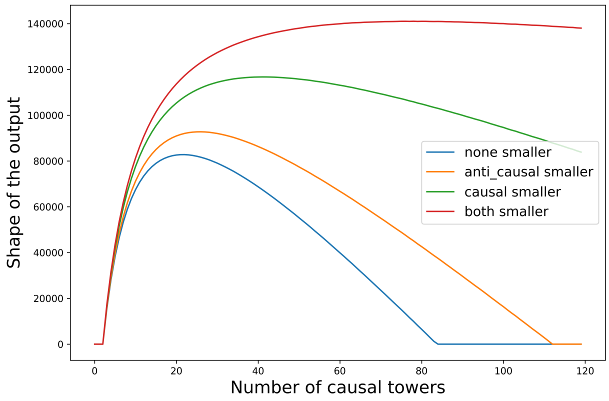

Figure 2: *Shape of the resulting sequence at variable value of p and kernel sizes of Causal (CT) and Anti-Causal (AC) towers*
"

$$l = \frac{L - s}{(1 - \gamma)(p - 1)} + \delta$$

As in equation 6 in section 2.3, the loss function utilized for SyncNet is weighted sum of realizations from three functions, viz. the simple MSE function, $\mathcal{L}_1 = \sum_{i=1}^{N^{'}} (R_i - \hat{R}_i)^2$, the root-mean-log error function, $\mathcal{L}_2 = \sqrt{\sum_{i=1}^{N'} (\log R_i^{\wp} - \log \hat{R}_i^{\wp})^2}$, and the KL-Divergence loss $\mathcal{L}_3 = \sum_{i=1}^{N^{'}} R_i^{\wp}(\log R_i^{\wp} - \hat{R}_i^{\wp})$ with $l_i \forall i \in \{1, 2, 3\}$ being the corresponding weights associated with each term. Being a regression problem, the conventional MSE loss might have worked, but since correlation sequence of signals acquired from real-time sampling may be variable in amplitude, one should also attend minimizing the relative difference between true and predicted sequences. This corresponds to weighted sum with the root-mean-log loss. Moreover, the MSE loss is used over the resulting correlation sequence while other loss functions optimize the pooled sequences. Though a relatively lower weight, $l_1$ was set, it would still help in making the resulting sequence more suitable for the final optimization by other loss functions. Figure 2 shows the influence of $p$ and kernel sizes on the size of the resulting sequence. It can be observed that naturally after some number of causal towers, as $\delta$ gets closer to $l$ with increase in $p$, the size starts reducing.

## 3. Designing deep-networks for comparison

As in section 3, we realized that some of the most common deep-learning based methods for audio-related tasks use mel-spectrogram fed CNNs. We designed a standard CNN taking the spectrograms of the signal-of-interest and corresponding reference signal. One of the simplest deep-learning approaches, i.e., using a 1-dimensional ConvNet was also used for comparison with SyncNet. We ensured the number of layers to be nearly equal. These networks did consist of conv, pooling, batchnorm, etc. layers with dropout.

We noticed that SyncNet takes nearly 20% more time for training than the two networks, and almost no to very little difference in inference time. This is likely because SyncNet consists of more learnable parameters per layer, because of windowing. The performance in terms of precise delay estimates however, is relatively higher.

## 4. Computational cost and Transformed Sequence

We experimented with varying hyperparameters like window-overlap, $\delta$, number of convolutional towers, $p$ and depth of CT or AC, etc. to study the effect of these hyperparameters on the computational complexity of SyncNet. Figure 4 shows that the rate of change in number of learnable parameters is affected by both the size of the causal towers as well as the anti-causal convolution structures, more dominantly by the latter.

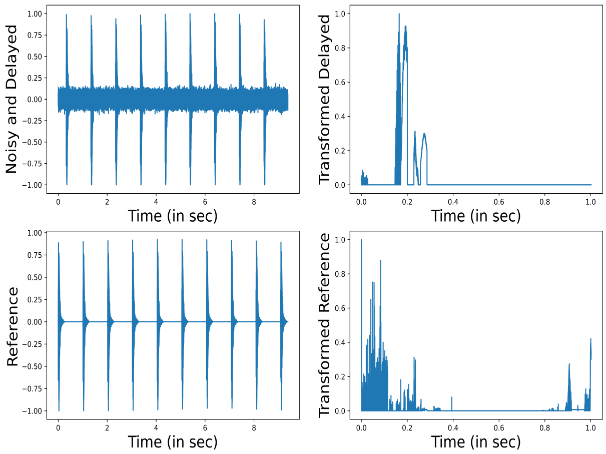

Figure 3: *Input Output for MTic Dataset*

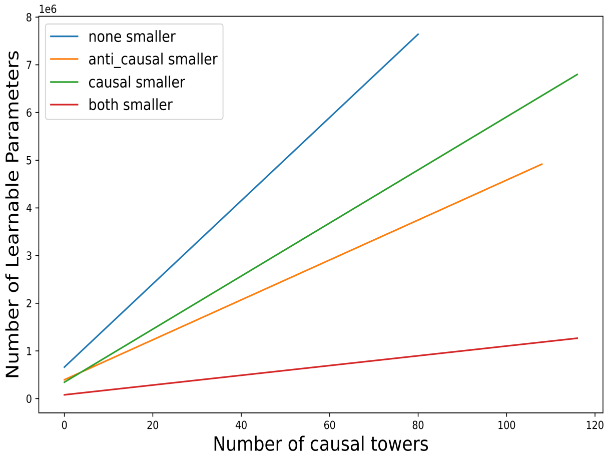

Figure 4: *Influence of p and filter sizes on the number of learnable parameters*

In order to visualize the transformed sequences by SyncNet, we plotted the corresponding sequences for MTic and LibriSpeech dataset in figure 3 and figure 5 respectively. We noticed that the transformation of delayed and noisy signal is cleaner than that of the reference input signal for most of the examples.

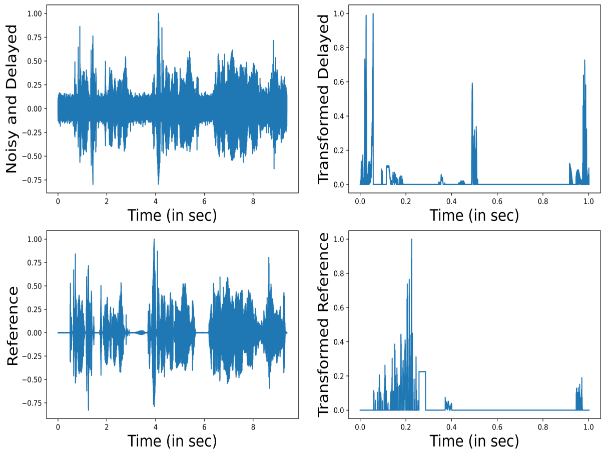

Figure 5: *Input Output for LibriSpeech Dataset*