Generative Audio Inpainting in the Frequency Domain

G60 (s1874170, s1837050, s1886315)

Abstract

This work tackles the task of audio inpainting (reconstructing missing samples) for speech data. First and foremost, this work will motivate why, and experimentally show that, the field of audio inpainting can benefit from deep generative models (conditional GANs in particular) applied to the frequency domain of audio data. Overall, the results show that an adversarial approach does not only provide competitive results with more traditional approaches (such as LPC) but also generates sharper reconstructions that closely approximate the target. Finally the effect of varying the amount of missing audio data and size of the available context on the quality of the reconstruction is investigated.

1. Introduction

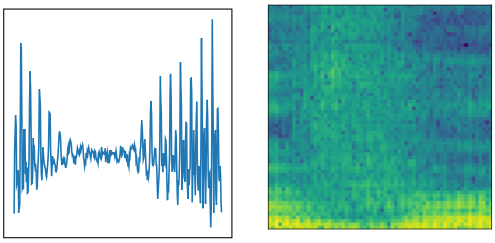
Communication is an integral part of human society, it sets us apart, and throughout the history, the means by which we communicate have evolved continuously. Nowadays, it is even possible to talk to people across the globe in real-time. This process, however, is not yet problem-free. During a conversation, the audio can sometimes be noisy, of low quality, or even falter. The field of audio improvement has a rich history and is successfully being applied to improve such audio issues. Especially in the last few years methods have consistently been improving due to the rise in popularity of deep learning methods, which are being applied with increasing success in the audio domain, achieving state of the art results on some of the aforementioned issues (Pascual et al., 2017; Kuleshov et al., 2017; Michelsanti & Tan, 2017).

In this work, deep generative models will be applied to solve the issue of restoring missing gaps in audio data (faltering audio), a task referred to as audio extrapolation, reconstruction or inpainting. This issue could be, for example, the result of packet loss in a VoIP (Voice over IP) call, or due to the corruption of parts of any audio data. More specifically, we will evaluate how a conditional Generative Adversarial Network (cGAN) may be applied to restoring gaps in the (2-dimensional) frequency domain. This transforms the problem to one closely resembling image inpainting, on which such models have long since proven to produce state-of-the-art results (Mirza & Osindero, 2014) (Isola et al., 2017).

In the next section, previous work on both audio and image inpainting will be discussed, as well as works that previously applied generative models for audio improvement in general. Afterwards, in section 3, the specific contributions of this work will be set out, followed in section 4 by a detailed description and justification of the exact approach taken. Then, in section 5, the experiments and achieved results will be discussed, and finally, conclusions will be drawn in section 6.

2. Background and Related Work

Audio signals can be analysed in the time and frequency domain. In the time domain we visualise how the amplitude x_t of the signal changes as a function of time t (one dimensional data; often called the waveform of the audio, an example is shown in Figure 1(a)). In the frequency domain, we use the fact that a signal can be expressed as a combination of basic sine waves of different frequencies f_n and represent how much of each frequency is present in the signal at a given time $\bar{t}$ (two dimensional data; often called the frequency spectrum of the signal, an example can be seen in Figure 1(b)). For each given frequency, the sine wave is usually represented as a complex number; in polar coordinates, the magnitude A indicates the amplitude of the sine waves and the phase ϕ denotes how much the sine wave is shifted (i.e. $A \sin(f_n t + \phi)$). An example of a function that converts a signal from the time to the frequency domain is the Short Time Fourier Transform (STFT). Conversely, the inverse (iSTFT) combines the frequency information to retrieve the time-domain representation. These conversions are not lossless, but can be performed with inaudible error.



(a) Audio Waveform (b) Frequency Spectrum

Figure 1. Time and frequency domain representations of the same audio snippet. Both figures have time as the x-axis. The waveform and spectrum have as y-axis the amplitude and frequency, respectively. The amplitude is conveyed by the pixel intensity in the spectrum, where yellow corresponds to higher amplitudes, and blue to lower amplitudes.

2.1. Audio Inpainting

The reconstruction of missing audio samples, also known as audio inpainting, has been an active field of research for decades due to its importance in communication, audio analysis and compression. Autoregressive (or AR) methods are a set of interpolation teachniques that have been widely studied for many years (Janssen et al., 1986; Etter, 1996; Esquef et al., 2003) and aim to solve this task in the time domain. In AR modelling, x_t (the amplitude of the signal at a given time) is predicted as a linear combination of the preceding p values:

$$x_t = \sum_{i=t-p}^{t-1} c_i x_i$$

To reconstruct a gap of missing audio in a given audio sample, the AR coefficients c_i are then first fitted based on the audio samples that are present (the context) by minimising, for example, the mean squared error. Afterwards, the missing values are predicted in an iterative manner (i.e. the predicted value x_t will be re-used to predict x_{t+1}). The amount of values p in the context used to predict a sample is referred to as the order. In this sense, the reconstruction can be seen as the convolution of a filter of size p along the audio signal.

Basic AR methods are well-suited for small gaps. When the number of missing samples increases, however, an increasing number of previously predicted values are used to make new predictions, which can significantly harm performance (Esquef et al., 2003). To alleviate this issue, the order p can be increased. This is only effective, however, if there is a certain stationarity in the signal, and also results in a more computationally expensive process. Alternatively, if samples before and after the missing gap are available, both a forward and a backward prediction can be combined to increase performance (Etter, 1996). An example of such a method is Linear Predictive Coding (LPC) (Kauppinen & Roth, 2002). In spite of its simplicity, LPC has been shown to be a competitive approach at recovering missing samples from pure tones, instrument sounds and music (Marafioti et al., 2018). LPC has also been widely used to analyse speech data providing good approximations (although presenting problems with nasal and high-pitched speech) (Rajman & Pallota, 2007, p.106-107).

AR methods are, however, restricted to exploiting the local linearity of the data. Making them, by design, prone to miss long-term patterns and incapable to model non-linear relationships. Secondly, as they treat every recording separately, AR methods do not exploit possible extrapolations that can happen inside of a domain. Besides linear methods, non-linear approaches (Cocchi & Uncini, 2002), as well as Bayesian methods (Godsill & Rayner, 1998) have also been investigated, each providing different benefits when mitigating these issues.

The more recent deep learning methods, however, provide an excellent candidate to solve both of these issues. Deep learning based approaches are known to excel at this type of pattern recognition and can be trained on large data sets. Arguably the most successful deep learning models are convolutional neural networks (CNNs) having many applications in the image domain (see (Bhandare et al., 2016) for a review). Although also available in 1D, the most well-known CNN architectures take as input 2D data, making these methods more naturally applicable for audio inpainting in the frequency domain. This transforms the problem to one closely resembling image inpainting, on which such models have long since proven to produce state of the art results (which will be discussed in the following subsection).

This approach (using a convolutional encoder/decoder setup) was investigated in a recent work (Marafioti et al., 2018), which showed to be successful at reconstructing the frequency data for gaps of 64ms in size on instrumental data sets. However, due to the nature of the data-sets (for which a certain degree of stationarity is common) the proposed method does not substantially improve the reconstruction quality obtained by LPC. As noted by (Marafioti et al., 2018), when reconstructing audio in the frequency domain, only magnitude information has to be reconstructed, as phase-less reconstruction methods (Griffin & Lim, 1984a; Prusa et al., 2017) can restore the time-domain audio signal from just magnitude information alone without any audible error.

2.2. Image Inpainting

Image inpainting is the task of filling in missing pixel values in an image region in a visually plausible manner. This makes it a very similar task to audio inpainting in the frequency domain (where missing parts of the spectra (1(b)) would be painted in). Image inpainting has received notable commercial and research interest since its introduction (Bertalmio et al., 2000), and has many potential application areas. Examples include restoring corrupted images, decensoring images, and replacing entire image regions, which is useful for many common image processing tasks, such as red-eye, logo or text removal. Similar applications could also translate over to audio inpainting.

Early work on image inpainting is divided into two main approaches, textural inpainting (Efros & Leung, 1999; Jia & Tang, 2003), and structural inpainting (Chan & Shen, 2001; Telea, 2004; Barnes et al., 2009), with later works also proposing hybrid approaches (Bertalmío et al., 2003; Criminisi et al., 2004). These methods perform well on regions with continuous structures and patterns, but struggle to synthesise unseen content and to handle large missing regions.

More recently, deep learning approaches have demonstrated state-of-the-art results in image inpainting, synthesising realistic and original content that blends into complex contexts. One of the first deep learning approaches to image inpainting, (Pathak et al., 2016), proposed encoding image context using a CNN architecture to generate ‘natural’ image regions. Furthermore, they demonstrated the effectiveness of combining a reconstruction loss with an adversarial loss, to produce sharper and higher quality outputs. To compute the adversarial loss a Conditional Generative Adversarial Network (cGAN) was used, as proposed by (Mirza & Osindero, 2014). In later work (Isola et al., 2017), a general image-to-image translation framework was proposed, known as PIX2PIX, which achieved state of the art results for a variety of image-to-image tasks, including image inpainting.

The PIX2PIX framework also uses a cGAN model trained on paired training examples, but introduce two important architectural changes. Firstly, a U-Net (Ronneberger et al., 2015) architecture was used as the generator, instead of a more straightforward convolutional encoder/decoder model. The difference between the two, is that U-Nets include skip connections between the encoder and decoder (as illustrated by Figure 3), which prevents the issue that all information has to be passed through low-resolution 'bottleneck' layers; information on all levels of detail can be preserved. Secondly, a PatchGAN classifier was used for the discriminator. Instead of directly outputting a scalar prediction for whether the image is an original or a fake, a PatchGAN produces such predictions at the level of patches (hence outputting, for example, a 4×4 grid of values, each corresponding to a different patch in the original image). It was shown that using a PatchGAN as opposed to a standard GAN, results in sharper inpaintings (Isola et al., 2017).

2.3. Conditional GANs for Audio Improvement

Although yet to be applied to audio inpainting specifically, the PIX2PIX framework (and similar approaches) have already been adapted to solve other audio improvement tasks in the frequency domain.

In (Michelsanti & Tan, 2017), the effectiveness of the PIX2PIX approach is investigated for noise removal in the frequency domain, achieving promising results. (Donahue et al., 2017) further explore the use case of speech enhancement in the context of Automatic Speech Recognition by including reverberant noise in the audio samples. Their presented network (FSEGAN) shows an improved performance over a prior implementation of the same task on the time-domain (Pascual et al., 2017). This further demonstrates the effectiveness of cGAN-based approaches for audio improvement in the frequency domain.

3. Contributions

First and foremost, this work will motivate why, and experimentally show that, the field of audio inpainting can benefit from deep generative models (conditional GANs in particular) applied to the frequency domain of audio data. Specifically, cGANs will be applied to predict missing gaps in speech data based on context to show that this approach can be used to generate realistic audio reconstructions.

Afterwards, the effect of modifying various settings of this problem will be explored on the same data set:

- How does the duration of the missing gap effect reconstruction performance? How much missing data can be reconstructed with high accuracy?
- How much context (i.e. the amount of data, before and/or after the missing data) is necessary for a good reconstruction?

Speech data has been selected in particular as it includes uncommon audio patterns and may show the potential of deep learning techniques on more than just the commonly used instrumental data sets. In addition, audio inpainting is very relevant for speech data, for applications such as VoIP, as discussed in the introduction.

4. Methodology

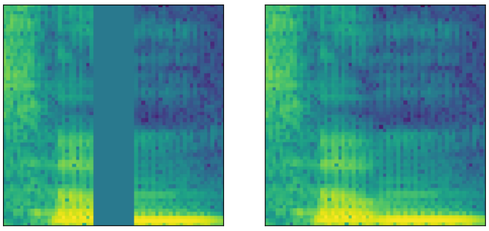
4.1. Task, Dataset and Setup

As discussed in the introduction, this work focuses on the inpainting of audio data in the frequency domain. More precisely, the setup used in this work, and the tasks solved, are as follows:

1. First the original audio data is transformed to the frequency domain using the Short Time Fourier Transform (STFT).
2. The resulting complex-valued data is transformed into its polar coordinates as it can be more easily normalized to a well-behaving distribution than in the Cartesian coordinates[1] (See Figure 4). Note, however, that phase information is discarded. This is possible as phase-less reconstruction methods are capable of reconstructing the audio signal without phase information with in-audible error as discussed in subsection 2.1.
3. Finally, random patches across the frequency data are extracted, and training pairs are created, consisting of the original patch (target image), and the original patch with a middle stroke set to zero (source image). This creates pairs as seen in Figure 2. The patches used in the work have a height of 64 pixels (this is discussed in subsection 4.3).
4. The objective is then to fill in the missing magnitude data of the source image such that the inpainting is natural and blends seamlessly into the surrounding context.

The speech data set used for training and evaluation is the the CSTR VCTK Corpus (Veaux et al., 2017) which consists of sentences spoken by 109 native English speakers with various accents. All speech data was recorded in a hemi-anechoic chamber using an omni-directional head-mounted microphone. The recordings have been downsampled from 96kHz to 48kHz, and compressed to 16 bits.

[1] (Marafioti et al., 2018) also showed experimentally that predicting the magnitude information instead of the complex numbered information improves results.



(a) Source (b) Target

Figure 2. One example pair from the validation set (patch-width 170ms, gap-width 32ms).

After processing, the full data set contains about 20 hours worth of continuous speech data, which was then split into 16 hours for the training set, and 2 hours for both testing and evaluation sets.

Specifics about the model used for the inpainting, as well as the loss used for training are discussed in subsection 4.2. Then, in subsection 4.3, the STFT and data processing steps are further detailed. Finally in subsection 4.4 two methods used to measure the quality of a reconstruction are presented, as well as baseline methods that were used for comparison.

4.2. Model Architecture and Loss Function

The most straightforward approach for solving the task outlined above would be to aim to reconstruct the target images ($\mathcal{T} = [t_1, t_2, ..., t_n]$) from the source images ($\mathcal{S} = [s_1, s_2, ..., s_n]$). A similar approach is taken by (Marafioti et al., 2018), in which a DNN taking the *source image* as input, is tasked to output the *target image*, and the overall model is optimized using (a slightly tweaked version of) the mean squared error (MSE) between the two images. It might not necessarily be the case, however, that achieving the lowest possible MSE will result in the most natural, realistic inpainting for the audio.

To see why, imagine, for example, that the original stroke contains values randomly sampled from some Gaussian distribution. Minimizing the MSE would then correspond to filling in a constant value – equal to the mean of the Gaussian – for each of the values in the gap, which would certainly not result in the most natural sounding inpainting. Although such an extreme case is not likely to happen for real-world audio, any audio track is prone to have some form of noise behind it, or other elements that can not be predicted exactly from the context (and could hence suffer from excessive smoothing). For this reason, it seems probable that using an adversarial loss, in addition to a loss designed to reconstruct the target, can help to produce more realistic audio reconstructions.

Our hypothesis is that by introducing an adversarial loss, the network will be able to fill in 'noise' with actual noise (instead of constant values) and learn to create sharper and more 'natural' reconstructions overall, as was the benefit seen for the task of image inpainting. To investigate this hypothesis both the general approach followed by (Marafioti et al., 2018) (as discussed in subsection 2.1) and the PIX2PIX framework (as discussed in subsection 2.2) will be implemented and compared. To allow for fair comparisons the same U-Net 'generator' was used for both approaches, making the only main difference the inclusion of an adversarial loss. From now on, both approaches will be referred to as 'U-Net' and 'U-Net Adv', respectively. The implemented U-Net model is a fairly standard 5-layer U-Net, a high level description can be seen in Figure 3.

Following the PIX2PIX framework, The U-Net approach was trained using just the L1 pixel-wise loss ($\mathcal{L}_{L1}$) at the location of the missing pixels. The adversarial loss ($\mathcal{L}_{adv}$) which is also needed to train the U-Net Adv. approach comes from the discriminator (D). Following PIX2PIX, a PatchGAN discriminator (as described in subsection 2.2) was used. The specific network used can be seen in Figure 8. To get a scalar adversarial loss from the grid of outputs, the MSE between the grid of adversarial losses and an equal-sized grid of target values y (which denote whether the input is real, $y = 0$, or generated, $y = 1$) is computed. For a given sample we get:

$$\mathcal{L}_{adv}\left(D\left(s_n, x\right), y\right) = \mathrm{MSE}\left(D\left(s_n, x\right), y\right) \tag{1}$$

where x is the input of the discriminator corresponding either to t_n (in which case $y = 0$) or $G(s_n)$ (for which $y = 1$).

The objective of the U-Net generator (G) is to deceive the discriminator by generating as realistic output images as possible (minimising $\mathcal{L}_{adv}$), as well as generating outputs close to the original target (minimising $\mathcal{L}_{L1}$) while being conditioned on the input source image. Often noise is added to the input of the generator, this was not done, however, as deterministic results are fine for the given application.

More formally, the cGAN approach is a min-max optimisation problem in which the two networks (G and D) are competing against each other and iteratively optimise opposing objective functions. The generator is trained to optimise (fixing D):

$$G^* = \underset{G}{\mathrm{argmin}}\, \mathcal{L}_{adv}\left(D\left(s_n, G\left(s_n\right)\right), 0\right) + \lambda \mathcal{L}_{L1}, \tag{2}$$

$\lambda\mathcal{L}_{L1}$ is the weighted L1 pixel-wise loss between the generated image, $G(s_n)$, and the original target image, t_n. The weighting factor λ was set to ten, which balances the magnitudes of the adversarial and pixel wise losses (more specifically, this causes the pixel wise loss to be about twice as high as the adversarial loss on average, putting slightly more weight on preferring one-to-one reconstructions over realistic ones). The discriminator optimises (fixing G):

$$D^* = \underset{D}{\mathrm{argmax}}\, \frac{1}{2}\left(\mathcal{L}_{adv}(D(s,t), 0) + \mathcal{L}_{adv}(D(s, G(s)), 1)\right), \tag{3}$$

which can be seen as the average of the adversarial loss for predicting original target images, t_n, and the adversarial loss for predicting generated images, $G(s_n)$.

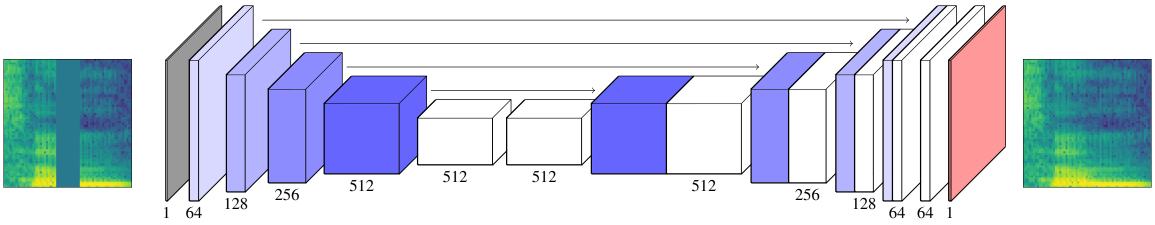


Figure 3. Specific U-Net architecture used. Same coloured blocks and arrows denote skip connections.

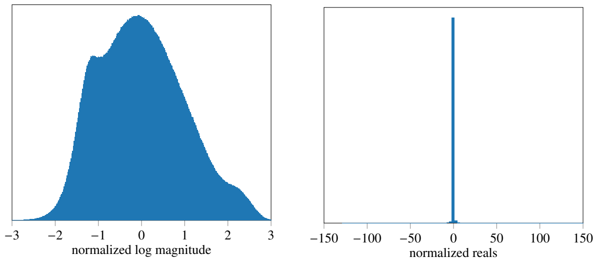
The remainder of this section outlines implementational details on both the generator and discriminator models. The used generator (as shown in Figure 3) was fully implemented according to the original PIX2PIX implementation, which includes LeakyReLU activation units (with slope coefficient 0.2) and uses dropout (with a probability of 0.5) after each of the last two down-sampling layers. These changes were originally made to facilitate training, by enabling better gradient flow and to prevent over-fitting. One final detail to note, however, is that the used U-Net model consists of less layers overall, as the used inputs are of a lower resolution than in PIX2PIX.

The discriminator model, depicted in Figure 8, is again similar to the PIX2PIX model. It is a four-layer deep convolutional network with LeakyReLU activation layers and instance normalisation (Ulyanov et al., 2016). It takes as input the source image, s, which it conditions its prediction on, and the target image, t, and concatenates the two images channel-wise. The only major deviation is that the convolutional layers were setup as such that the final output has a receptive field size of 38 × 38 (i.e. the size of the patches on which PatchGAN basis its predictions). In the original work a patch size of 70 × 70 was used, this was too larger, however, as the input images of this work are just 64 × 64. Using patches that are too small (e.g. 16 × 16), though, can cause tiling artefacts, as found by (Isola et al., 2017), which motivates the choice of 38 × 38.

4.3. Data Processing

The implementation of the (i)SFTF used in this work minimizes the MSE between the original signal, and a signal transformed using the SFTF followed by the iSFTF (Griffin & Lim, 1984b). The SFTF transformation was setup as such that it results in a total of 64 different frequency values from 0 to 12KHz which causes each 'pixel' (when using default settings) to correspond to 2.67ms across (or 128 audio samples at the given bit-rate of the data-set). Capping the frequencies at 12KHz is justified as most speech information is present below 6KHz. Limiting the resolution to just 64, however, frequencies was necessary as computational resources were limited and it was not possible to train on images with a larger size. It is not possible, however, to use phase-less reconstruction methods at the given resolution, preferably the resolution would have been set to 256 different frequencies, which does allow for phase-less reconstructions. Due to this, it is not possible to generate audio samples from the given reconstructions, and this work can only investigate in the potential given the results in the frequency domain.

After the transformation to the frequency domain, the data is normalised by taking the normalised log magnitudes, this results in a significantly better behaving data distribution, as illustrated by Figure 4. As mentioned above and in section 2, the phase information is discarded as the problem can setup such that phaseless reconstruction methods can reconstruct the audio signal with just magnitude information alone. Finally, during training, random patches of the entire data-set are taken with gaps set at the center.



(a) Normalised Magnitudes (b) Complex Valued

Figure 4. Normalised data distributions of the frequency spectra. Left shows the data expressed as the normalised log magnitude, right shows the real values of their complex values directly (the imaginary values followed a similar distribution).

.

4.4. Baselines and Performance Measures

Besides the DNN approaches, the LPC method as discussed in section 2 (one of the go-to methods for audio inpainting) is also included as a baseline approach for in-painting in the time domain.

To gain insight into the quality of the reconstruction, two quantitative performance measures are used, one for the time and another for the frequency domain. To evaluate the reconstruction in the frequency domain, the MSE between reconstructed frequency data and the target data (at the location of the patch) is used. Although this does not necessary show how close the audio comes to a *realistic* inpainting, as discussed in the previous subsections, it can still serve as a good comparison of how close the reconstruction gets to the target.

For the time domain, the signal-to-noise ratio (SNR) measure is used for evaluation:

$$SNR(y, \hat{y}) = 10 \log_{10} \frac{P_{signal}}{P_{noise}} = 10 \log_{10} \frac{||y||_2}{||y - \hat{y}||_2}$$

where y is the target and $\hat{y}$ the inpainted signal, both measured at the gap. This metric captures the level of noise that is present in a signal with a positive value indicating that there is more signal than noise. In other words, the larger the SNR the cleaner the audio and the more accurate the prediction.

To measure the SNR of the audio inpaintings, they would have to be transformed back to the time domain, and as discussed in the previous subsection, using phase-less reconstruction methods will not be possible for the setup used in this work. Instead the original phase information of the target signal will be used, after which the signal can be transformed back using just iSTFT. To make comparisons fair, this is done for all approaches including LPC which is performed in the time domain. Doing this, however, has obvious drawbacks, as the Adv U-net approach does not try to exactly reconstruct the original target, in which case using the original phase information of the target, can degrade the SNR.

5. Experiments

The main experiment of this work (i.e. comparing the result of the Adv. U-net method to the various baselines) is discussed here. Then later in subsection 5.1, the effects of variable gap and context sizes are discussed. For the main experiment, the various models are tasked to paint-in 32ms of missing audio data (which corresponds to a width of 12px in the frequency domain) using a context of 69ms on either side (a width of 26px in the frequency domain). Overall this results in 'images' with a width 64×64 'pixels'.

The U-Net model was trained for 16 full epochs using the Adam optimiser with default settings (Kingma & Ba, 2015). Although the model did not yet fully converge, improvements were minimal. The Adv. U-net model was similarity trained with Adam (for both the generator and the discriminator) and training was capped at 30 epochs (training plots can be found in Figure 9), due to computational constraints (although improvements were still clearly noticeable). Training parameters for the Adv. Unet model were set to follow the original pix2pix implementation ($\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a batchsize of four). The weights of the generator were initialised to the weights of the trained U-Net model and the cGAN was trained in accordance with useful tips have been compiled from a NIPS talk given in 2016, 'How to Train a GAN?'[2]. This was done to facilitate quicker convergence, motivated by the limitations on the available computational resources.

Results on the performance measures on the validation set are noted in Table 1 and resulting in-paintings on the validation set are visualized in Figure 5 (for two validation examples) and in Figure 10 (for additional validation examples).

	Magnitude MSE	**SNR**
LPC	0.3726 ± 0.2275	5.047 ± 5.703
U-Net	0.1180 ± 0.0903	8.014 ± 3.225
Adv. U-Net	0.2127 ± 0.1300	4.778 ± 2.317

Table 1. Performance measures (MSE and SNR) on the validation set for the various methods.

Table 1 suggests that the DNN approach is indeed suitable to solve the problem of audio-inpainting for speech data, as was expected from the results of (Marafioti et al., 2018). In this case, the neural network model obtains a significantly better SNR value than LPC. This can be in part due to the nature of the dataset, as discussed in subsection 2.1, on the other hand this can be explained by the fact that the U-Net benefits the most from adding the original phase information, as the U-Net tries to minimise the differences between the original magnitudes. Additionally, the U-Net model obtains a much better value for the MSE then all other methods, which is what should be expected as the network was optimised to minimise the difference between the the inpainting and the target.

Although obtaining the lowest MSE value, the results obtained for the U-Net (see Figure 5(d)), are rather 'washed out', which can be explained by the fact that the network needs to average over many training examples. In addition, we can see that the U-Net does indeed fill in constant values for noise (especially notable in the second example), as previously hypothesised. Looking at 5(d) we can see that the Adv. Unet model indeed resolves these issues as expected. Inpaintings look sharper, more natural and closer to the original. Looking at the second example, we can see that the Adv. U-Net is not capable of painting in the vertical stroke which was fully contained within the gap. If such elements would be part of speech, it may be possible that when trained on larger data-sets, or when using larger contexts, the network can learn to paint such aspects in. Also possible is that such parts were audio artefacts or noise, in which case the Adv. Unet even improves upon the original audio by filtering this out. Looking at Table 1 we can see that this does affect the MSE, however, which makes sense as the network is no longer only optimising for similarity with the original.

Finally we note that the SNR metric of the Adv. U-Net is also lower then the other methods. This can be explained in two ways. Firstly, by adding the original phase information on top of the generated magnitudes (which can be quite different from the original) the resulting time domain signal can be distorted. In addition, the SNR compares the result to the original audio signal. The Adv. Unet, however, generates realistic examples not necessarily similar to the original. In-painted noise could, for example, be arbitrarily different, even though it would audibly be the same.

[2] https://github.com/soumith/ganhacks

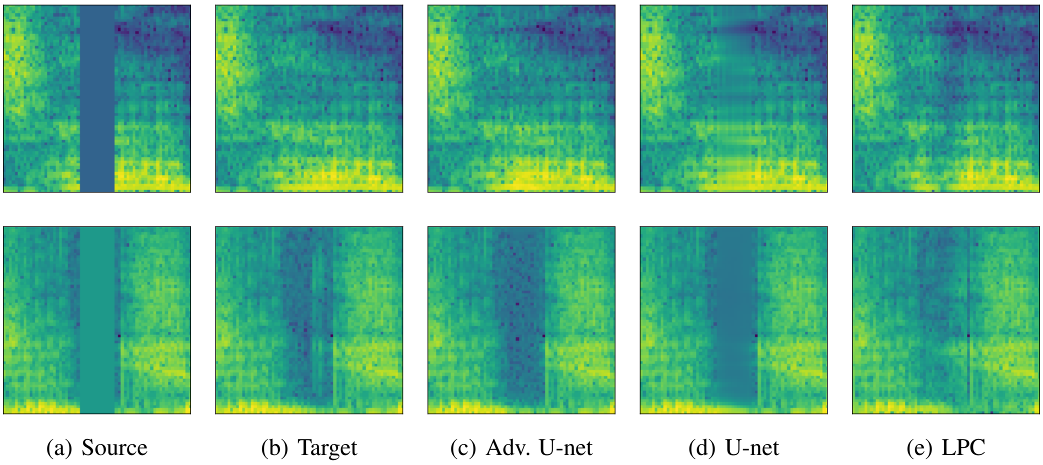


Figure 5. Audio reconstructions visualized via the normalized frequency spectrum of the audio snippets. The columns from left to right show: the audio with the missing gap; the original audio; U-Net (adversarial) reconstruction; U-Net reconstruction and LPC reconstruction. Each row shows a different sample from the validation set.

5.1. Varying Context and Gap Size

In this second experiment, the effects of the gap and context size on the quality of the reconstruction are investigated. Due to limitations on available computational resources, however, these could not be investigated with the Adv. U-Net model. Instead we consider them using the U-Net model which can give an idea about how the effects when using the Adv. U-Net model.

Specifically, multiple experiments similar to the main experiment were performed with different gap and context sizes. Gapsizes of 32ms, 64ms, 128ms and 256ms on image patches of size 85ms, 170ms, 380ms and 680ms were used (given that the gap fits in the overall patch). Note that not every experiment could be run for the same number of iterations as the main experiment, as experiments with larger context sizes are computationally more expensive. The training curves are shown in Figure 6 ordered by the gapsize.

Immediately obvious is that a larger context sizes improves performance in all cases in terms of convergence speed (and most likely in final performance, although this is not clear yet given the limited amounts of iterations trained for). This result is as expected, as only more information is available. We can see, however, that this effect is minimal for the smaller gap sizes, and increasing the context beyond a certain point has diminishing results (as can be seen in 6(c)). However, a larger context is of significant important once the gap-size increases. This is especially noticeable looking at 6(d) where there is more than a full point of difference between the MSE after about 10.000 training steps.

Finally, in Figure 7 the final performance for the gap-sizes 32ms, 64ms and 128ms is displayed where the context is similar in size. From this we can clearly see that increasing the gap-size has a big detrimental effect on training. The difference could become less, however, when training for a longer period of time, as the models were far from converged. To show this, the gradients (computed using finite-differences) were also shown the Figure 7, and show that for especially the larger gap-sizes, training was still very much ongoing (where the 128ms gap-size experiment still had a gradient 2.95 times larger than the 32ms gap-size experiment).

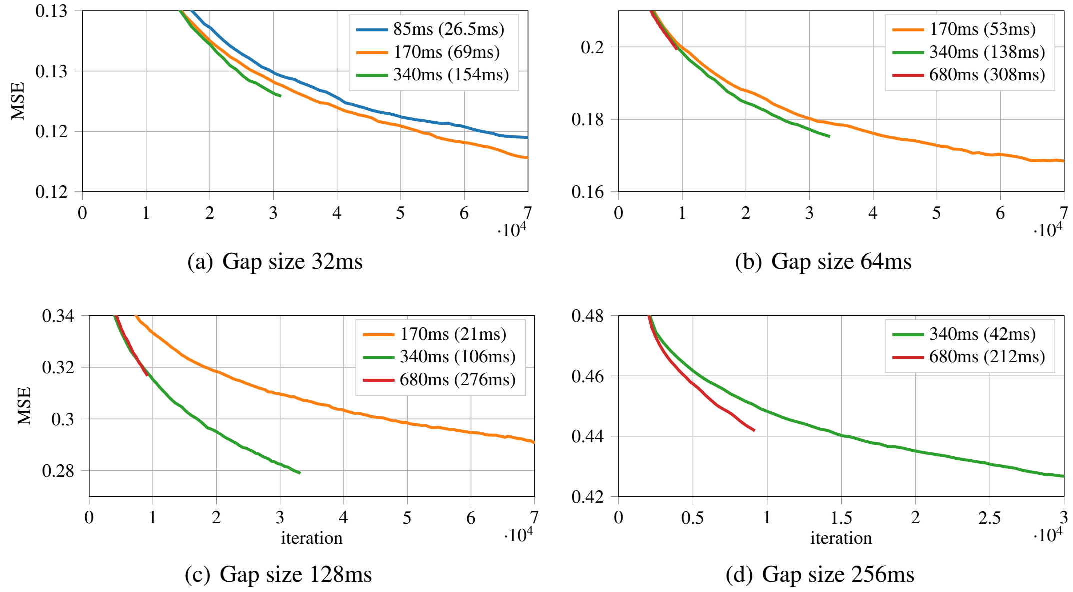
6. Conclusions

This work explored the task of audio inpainting using conditional GANs in the frequency domain. To do so, the performance was compared with two baselines LPC and a DNN approach without an adversarial loss. Although audio samples could not be generated due to the setup as discussed in subsection 4.3, the results in the frequency domain show that an adversarial approach does not only provide competitive results with more traditional approaches but also generates sharper reconstructions that closely approximate the target. By design, the used performance measures were not capable, however, of capturing this improvement, and the SNR measures could have further been degraded due to using the original phase data of the audio samples.

Finally, the effects of varying the size of the missing gap and available context available have also been explored. From the results shown, the main takeaway is that the importance of larger contexts increases with the size of the gap size as could be expected intuitively.

6.1. Future Work

This work shows promising results that support the usage of cGANs to reconstruct missing gaps in speech data. However, to fully evaluate the potential of this approach, a more thorough experimentation is required. Most importantly, the approach should be performed on data with a higher frequency resolution, such that actual audio samples can be produced. This will allow for arguably the most impor-



(a) Gap size 32ms

(b) Gap size 64ms

(c) Gap size 128ms

(d) Gap size 256ms

Figure 6. Training curves for various gap and context sizes. Each subplot shows various context sizes for a constant gap size (denoted by the caption). The legend labels show the size of the full image in milliseconds and the available context on either side in the parenthesis ($\frac{1}{2}\left(\text{size}_{image} - \text{size}_{gap}\right)$)
.

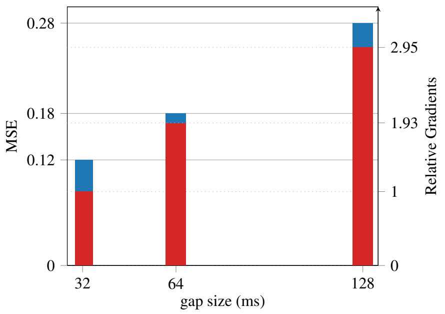


Figure 7. Final MSE (left/blue) and final relative gradients (right/red) for various gap sizes with an (almost) constant context of 130ms (±24; from left to right 106ms, 138ms, 154ms). Gradients are relative with respect to those of gap size 32ms (the leftmost bar). Gradients were computed numerically using the finite difference method.

tant evaluation of the performance in the audio domain: listening to the produced samples.

In addition to this, the experiments discussed in subsection 5.1 could also be executed using the Adv. U-Net approach directly so more conclusive statements can be made about the adversarial case. These experiments should also be run until convergence as the Figure 6 suggest that the models were still improving. Additionally, exploration of a single cGAN model‘s ability to reconstruct different gap sizes would be a natural next step. Reconstructing images with different sized gaps is possible as the cGAN model is trained to output the complete target image, and not just the gap.

Besides this changes to the proposed model and architecture could also be explored. The proposed U-Net model utilizes bilinear upsampling followed by a convolutional layer in the decoder, which may be a limiting factor for learning fine-grained details. Changing these to instead using deconvolutional layers would give the cGAN greater expressive power. Furthermore, the cGAN model is trained to minimise the L1 loss between the output and target images. This, however, may be a sub-optimal approach as it places the same importance on every frequency. If the focus is to reconstruct an audio segment such that it is *natural* sounding it may be beneficial to place a higher weighting on the reconstruction losses originating from the frequency ranges that humans are more sensitive to (i.e. lower frequencies).

Finally, exploring how the cGAN model performs on different types of data would be a beneficial future research direction. It would be particularly interesting to explore the following questions:

- Is the context information prior to the gap sufficient for high quality reconstructions? This is of particular interest for real-time applications, such as VoIP, where the context data after the gap is not available.

- What is the optimal amount of audio data to encode into one data point (or ‘pixel’)? Encoding fewer ms of audio data per pixel, provides the model with more information and will likely improve reconstruction quality, but at the same time, also increases the number of ‘pixels’ that have to be inpainted, making the task more difficult.

- Is it possible to train a single network to reconstruct gaps across different audio domains, for example on instrumental and speech data?

References

Barnes, Connelly, Shechtman, Eli, Finkelstein, Adam, and Goldman, Dan B. Patchmatch: a randomized correspondence algorithm for structural image editing. In *SIGGRAPH '09*, 2009.

Bertalmio, Marcelo, Sapiro, Guillermo, Caselles, Vincent, and Ballester, Coloma. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pp. 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5. doi: 10.1145/344779.344972. URL http://dx.doi.org/10.1145/344779.344972.

Bertalmío, Marcelo, Vese, Luminita A., Sapiro, Guillermo, and Osher, Stanley. Simultaneous structure and texture image inpainting. In *CVPR*, 2003.

Bhandare, Ashwin, Bhide, Maithili, Gokhale, Pranav, and Chandavarkar, Rohan. Applications of convolutional neural networks. *International Journal of Computer Science and Information Technologies*, 7(5):2206–2215, 2016.

Chan, Tony F. and Shen, Jianhong. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436 – 449, 2001. ISSN 1047-3203. doi: https://doi.org/10.1006/jvci.2001.0487. URL http://www.sciencedirect.com/science/article/pii/S1047320301904870.

Cocchi, G. and Uncini, A. Subband neural networks prediction for on-line audio signal recovery. *Trans. Neur. Netw.*, 13(4):867–876, July 2002. ISSN 1045-9227. doi: 10.1109/TNN.2002.1021887. URL http://dx.doi.org/10.1109/TNN.2002.1021887.

Criminisi, Antonio, Pérez, Patrick, and Toyama, Kentaro. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13:1200–1212, 2004.

Donahue, Chris, Li, Bo, and Prabhavalkar, Rohit. Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition. *arXiv:1711.05747 [cs, eess]*, November 2017. URL http://arxiv.org/abs/1711.05747. arXiv: 1711.05747.

Efros, Alexei A. and Leung, Thomas K. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.

Esquef, Paulo A A, Valimaki, Vesa, Roth, Kari, and Kauppinen, Ismo. INTERPOLATION OF LONG GAPS IN AUDIO SIGNALS USING THE WARPED BURGâĂŹS METHOD. pp. 6, 2003.

Etter, W. Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters. *IEEE Transactions on Signal Processing*, 44(5):1124–1135, May 1996. ISSN 1053-587X. doi: 10.1109/78.502326.

Godsill, Simon H. and Rayner, P. J. *Digital Audio Restoration: A Statistical Model Based Approach*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 1998. ISBN 3540762221.

Griffin, D. and Lim, J. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, April 1984a. ISSN 0096-3518. doi: 10.1109/TASSP.1984.1164317.

Griffin, Daniel W. and Lim, Jae S. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, April 1984b. ISSN 0096-3518. doi: 10.1109/TASSP.1984.1164317.

Isola, Phillip, Zhu, Jun-Yan, Zhou, Tinghui, and Efros, Alexei A. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.

Janssen, A., Veldhuis, R., and Vries, L. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(2): 317–330, April 1986. ISSN 0096-3518. doi: 10.1109/TASSP.1986.1164824. URL http://ieeexplore.ieee.org/document/1164824/.

Jia, Jiaya and Tang, Chi-Keung. Image repairing: Robust image synthesis by adaptive nd tensor voting. In *CVPR*, 2003.

Kauppinen, Ismo and Roth, Kari. Audio Signal Extrapolation - Theory and Applications. pp. 6, 2002.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Kuleshov, Volodymyr, Enam, S. Zayd, and Ermon, Stefano. Audio super resolution using neural networks. *CoRR*, abs/1708.00853, 2017.

Marafioti, Andrés, Perraudin, Nathanael, Holighaus, Nicki, and Majdak, Piotr. A context encoder for audio inpainting. *CoRR*, abs/1810.12138, 2018.

Michelsanti, Daniel and Tan, Zheng-Hua. Conditional Generative Adversarial Networks for Speech Enhancement and Noise-Robust Speaker Verification. *arXiv:1709.01703 [cs, eess, stat]*, September 2017. URL http://arxiv.org/abs/1709.01703. arXiv: 1709.01703.

Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

Pascual, Santiago, Bonafonte, Antonio, and SerrÃă, Joan. SEGAN: Speech Enhancement Generative Adversarial Network. *arXiv:1703.09452 [cs]*, March 2017. URL http://arxiv.org/abs/1703.09452. arXiv: 1703.09452.

Pathak, Deepak, Krahenbuhl, Philipp, Donahue, Jeff, Darrell, Trevor, and Efros, Alexei A. Context Encoders: Feature Learning by Inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.278. URL http://ieeexplore.ieee.org/document/7780647/.

Prusa, Z., Balazs, P., and SÃÿndergaard, P. L. A non-iterative method for reconstruction of phase from stft magnitude. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(5):1154–1164, May 2017. ISSN 2329-9290. doi: 10.1109/TASLP.2017.2678166.

Rajman, Martin and Pallota, Vincenzo. *Speech and Language Engineering*. EPFL Press, April 2007. ISBN 978-0-8247-2219-7. Google-Books-ID: h3SpC0oBc_AC.

Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Telea, Alexandru. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9, 01 2004. doi: 10.1080/10867651.2004.10487596.

Ulyanov, Dmitry, Vedaldi, Andrea, and Lempitsky, Victor S. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.

Veaux, C., Yamagishi, J., and MacDonald, K. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit, [sound]. University of Edinburgh. The Centre for Speech Technology Research (CSTR). https://doi.org/10.7488/ds/1994, 2017.

A. Additional Figures

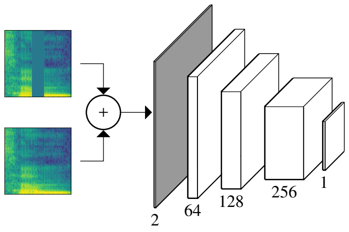
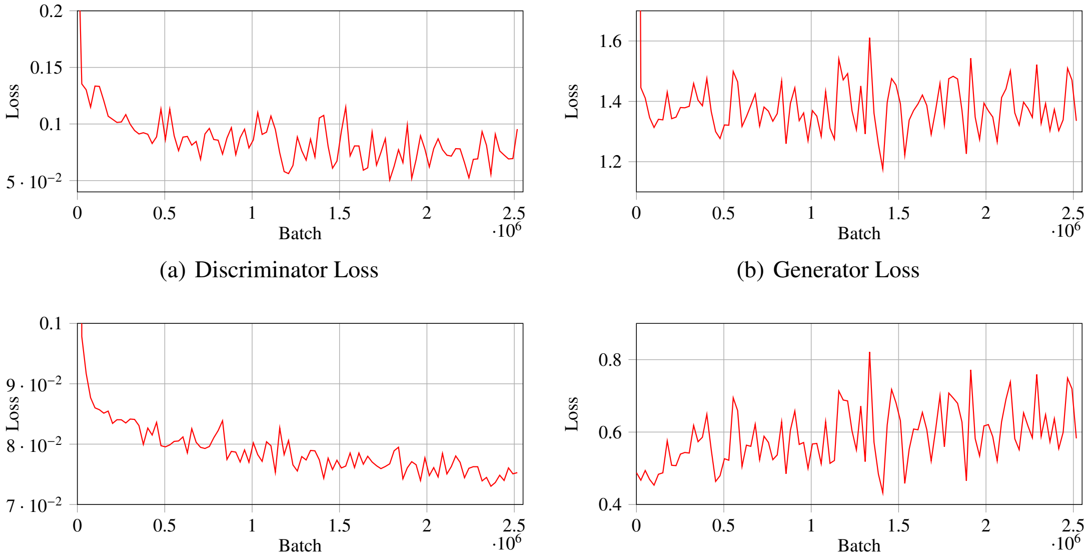


Figure 8. Discriminator architecture, with the circle node denoting channel-wise concatenation.



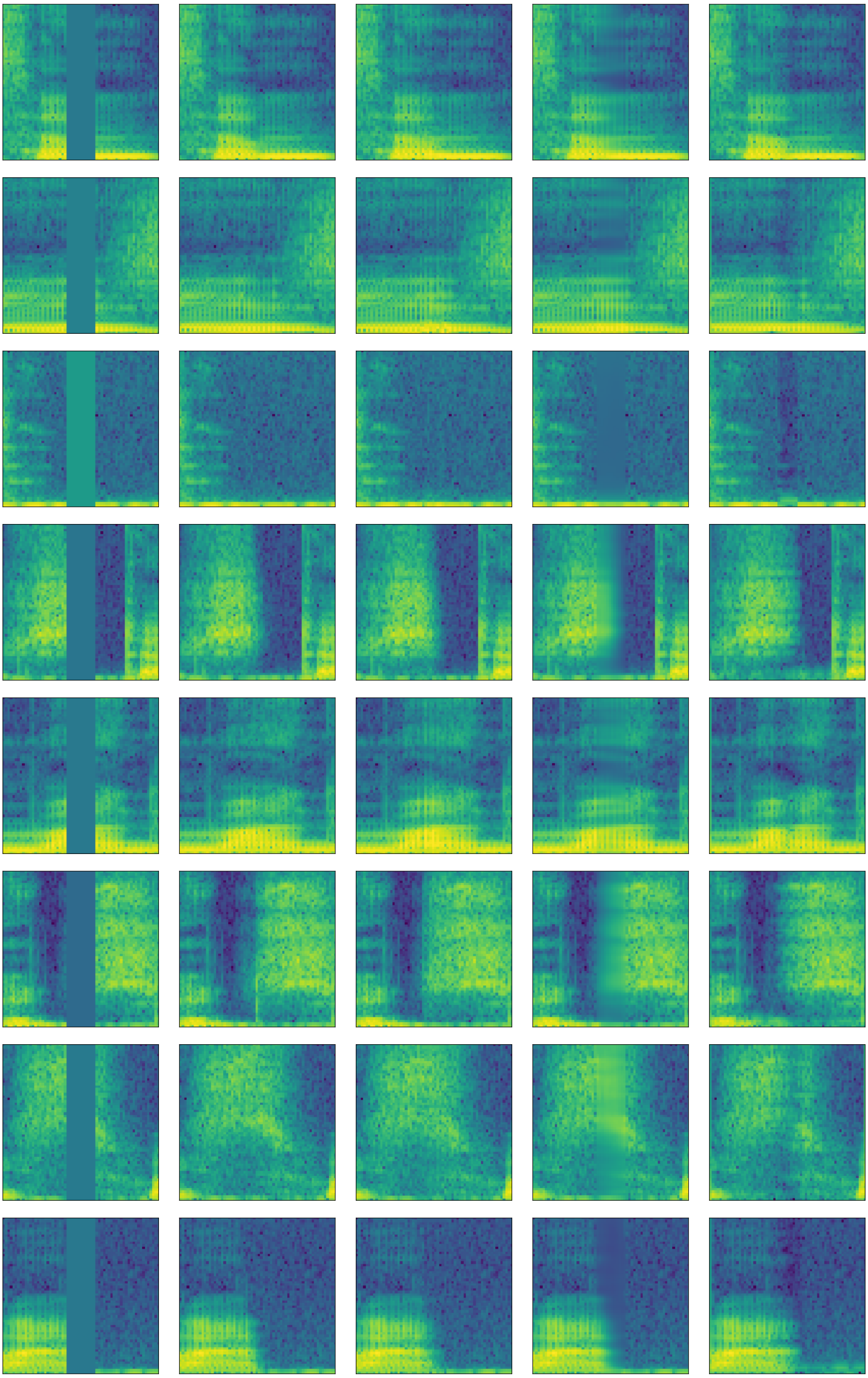
(a) Discriminator Loss

(b) Generator Loss

(c) L1 Pixelwise Loss

(d) Adversarial Loss

Figure 9. Training loss curves for the cGAN over 30 epochs.



(a) Source (b) Target (c) Adv. U-net (d) U-net (e) LPC

Figure 10. Similar to Figure 5; providing additional examples.