Hide and Speak

Parth Mehta, Swathi Jadav, Srinivasan Sathiamurthy, Vaibhav Agarwal
Carnegie Mellon University, Pittsburgh, PA
parthvim@andrew.cmu.edu, sjadav@andrew.cmu.edu
ssathiam@andrew.cmu.edu, vagarwa2@andrew.cmu.edu

Abstract

Steganography is a practice of encrypting a Secret Message within a carrier also known as a public message. We also try to extend our work by hiding multiple messages in a carrier signal. There are many traditional approaches for steganography with well-known algorithms, but this also makes them susceptible to Steganalysis (the practice of decrypting a hidden message). We, therefore, propose using deep learning for this task which makes it very difficult to find and decrypt what kind of function the model is performing. In this paper, we propose a method for hiding a secret message (M) inside or inside a carrier message (C). And the new message sounds just like the carrier message, which is then used to recover the hidden message. We also expand our work by encoding more multiple messages in single audio.

1 Introduction

Our work centers around the foundations laid by [1] that look into deep learning methods for audio steganography, which is the art of hiding hidden messages inside carrier messages. The premise of their research is focused on hiding messages inside a carrier signal such that the message embedded carrier is indistinguishable from the original carrier and the recovered hidden message must also be similar to the original hidden message.

For this task [1] looks into deep learning methods, where they use the magnitude of STFT. Short-Time Fourier Transform (STFT) is a matrix of real and imaginary values composed of windowed Fourier transforms. Although using the magnitude of STFT, is the most common method, there arises a problem while reconstruction of audio because the phase data is lost. This requires further processing in the form of phase reconstruction methods like Griffin-Lim[6].

Hence, we propose a new method where instead of using the magnitude of STFT, we use the real and imaginary parts from the STFT matrix and concatenate them channel-wise. As a result of this, each of our audio files now has two channels. This also solves the problem of phase reconstruction as both magnitude and phase can be interpreted from their real and imaginary counterparts. In doing so, we hope that the decoder will produce fewer artifacts for the hidden messages.

Initially, we have our audio files in the time domain, which are converted into the frequency domain by taking its STFT. This process can be done using a Convolution Neural Network (CNN) as well where kernel size acts as the window size and stride acts as the hop length. We are using a window size of 256 and a hop length of 128.

The architecture of our deep learning model consists of three parts, a carrier encoder, a carrier decoder, and a message decoder. The purpose of the carrier encoder is to find redundancies or empty spaces in our original carrier where we can hide our messages. The purpose of the carrier

decoder is to output a signal that is similar to the carrier signal but has messages hidden in them and the purpose of the message decoder is to recover the hidden message.

2 Literature Review

Most of the other prior works have focused on hiding messages in other media such as image, and have used a single message to be hidden in the carrier. Audio is a special case in Steganography, due to the exceptional amount of redundancies in representation due to availability of Phase,Amplitude and frequency features, these gives us much more freedom with respect to encoding hidden messages. Additionally, the techniques employed in Audio Steganography conceal hidden messages in the phase of the carrier message [2], embedding in lowest significant byte [3], and various other transform domains [4]. These methods however generalize poorly for one encoder and decoder across multiple messages, since perturbing a message after its fourier transform and reconstructing the phase before the inverse fourier transform results in audible artifacts, and it is not reasonable to create a single model for this level of exactness across many messages. Thus, [1] concludes that exploitation of specific forms of redundancies is sub-optimal. To that end, they propose learning a generalized encryption and decryption model.

One of the main obstacles of a generalized encryption and decryption model that has been previously addressed is data creation. To this end, the cryptographical nature of our problem requires adverserial training samples since we do not want someone with the wrong key to have access to the hidden messages. Along these lines, several papers [1, 7] have suggested adding adversarial examples in our data training process, such as messages paired with the wrong key to return the wrong response. This wrong response can be the correct response with slight perturbations, such as in adversarial learning, or more drastic such as minimizing its difference between background noise. They have also suggested using adversarial training examples to encode useful information, such as the keys themselves.

For the actual phase modification process of hiding our hidden message in the carrier, it is important to note that the phase discontinuity of an audio signal is perceptible to the human ear when the phase relation between each frequency component of the signal is dramatically changed [5]. Hence, it is important to ensure that phases align between time frames after modifications. Additionally, if the phases of a signal do not align, it is not possible to fit a complex exponential across the time frames to accommodate that modified signal [6]. Several methods exist to counter this effect, such as reassigning a band of frequencies in the neighborhood of signals that cross time frames, dampening signals near the time frame boundaries to a zero frequency, estimating phases from the frequency-time information, or using a sigma delta modulator. Among these, the most effective for quantization index modulation phase encoding is to force the phase shifts to go to zero at the frame boundaries [5], which eliminates most artifacts. However, this method also loses important information about boundary signals, since it dampens their frequencies to zero.

The approach our predecessors took to address this problem was to use differentiable short time fourier transforms and inverse short time fourier transforms. Although STFT directly encodes the frequencies of signals across the time frames, the interpretation of phases is more implicit. In fact, in most circumstances, the phase information from the STFT is not considered directly, and instead the derivative of the phase over time is looked at instead, since it is equivalent to the local instantaneous frequency [7]. Our predecessors employ this property extensively by constraining the STFT and ISTFT layers to be differentiable.

Another way to encode phase-related information is to include the magnitude and phase spectrum as input features of the relevant layers [11]. However, in doing so, one must be careful about identical frequencies that occur at different phase shifts. For this, one can use a representation of the raw phase information that is invariant to small phase shifts, such as pooling layers for CNNs.

3 Dataset Description

We are using the TIMIT dataset. The NIST Speech Dataset contains the complete Texas Instruments/Massachusetts Institute of Technology (TIMIT) [9] acoustic-phonetic corpus of read speech. TIMIT was designed to provide speech data for the acquisition of acoustic-phonetic knowledge and the development and evaluation of automatic speech recognition systems. It contains audio from 8 major dialect regions of the United States. 70% of the speakers are male and 30% are female. We have split the data into train/validation/test sets with each folder containing 3447, 306, 331 audio files. Each audio file has a sampling rate of 16000 which equates to 1 second length. We randomly generate pairs of carrier and message signals from the dataset.

4 Notations

These are the notations that we will be using in the following parts of the paper.

We will represent our Encoder as (*Ec*), our carrier decoder as a (*Dc*), and message decoder as (*Dm*). We consider our audio STFT as (*X + jY*) where *X* is the real and *Y* is the imaginary component. We will represent our concatenated carrier as *C = [Xc, Yc]* and message as *Mn = [Xmn, Ymn]* (*n* subscript is used for message number). And our reconstructed carrier and message are $\hat{C}$ and $\hat{M}$ respectively.

We are trying various things, the first is hiding 1 message inside a carrier. Next, is hiding 2 messages in a single carrier and also hiding 3 messages inside a single carrier. For this approach we are using multiple decoders, that is one decoder per message. Another approach that we tried is that we give each message a condition and the decoder operates on the message based on the condition, this is called a conditional decoder which helps to decode multiple messages by using only a single decoder.

We defer all discussion about model structures in the next section.

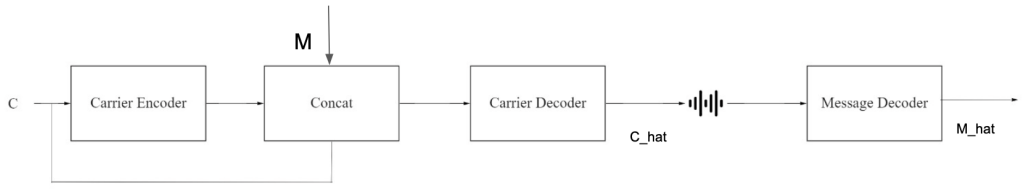


Figure 1: Network Architecture

5 Model Description

The model from our baseline [1] has 3 components: We follow their architecture, but make changes to how and the kind of data that is fed to the model. For all models gated convolutions are used. In gated convolution, the input passes through two branches both having a CNN but one having an extra $Sigmoid$ activation and the outputs from the two branches are multiplied to get the final output of the gated block.

5.1 Encoder

The Carrier encoder gives the encoded representation of the carrier signal. The architecture has 3 gated blocks. The encoder takes 2 channels(1 real , 1 imaginary) as input and gives out 64 channels as output. The output from the encoder is concatenated with the original carrier signal and also the message signals. This gives a tensor of channel dimension: $2+64+2*n$ incase of multiple decoders. And in case of a conditional decoder, we have $2 + 64 + n + (1 + n)$ were n is number of hidden messages and 64 is the output number of channels of the encoder block.

5.2 Carrier Decoder

The carrier decoder takes in the concatenated signals as the input and outputs two channels that are similar to the original carrier. it to channel output which is similar to the original carrier signal. The carrier decoder has 4 blocks of gated convolutions.

5.3 Message Decoder

The message decoder takes in the reconstructed carrier and outputs of the hidden message signal which is similar to the original message. The message decoder has 6 blocks of gated convolutions. We used two approaches for this, one in which we use multiple decoders that is one decoder for each message and foe the second approach we have a conditional decoder where we concatenate and condition or a key at the start and depending on the key the message decoder decodes the message.

$\boldsymbol{LossFunction}$ for multiple decoders

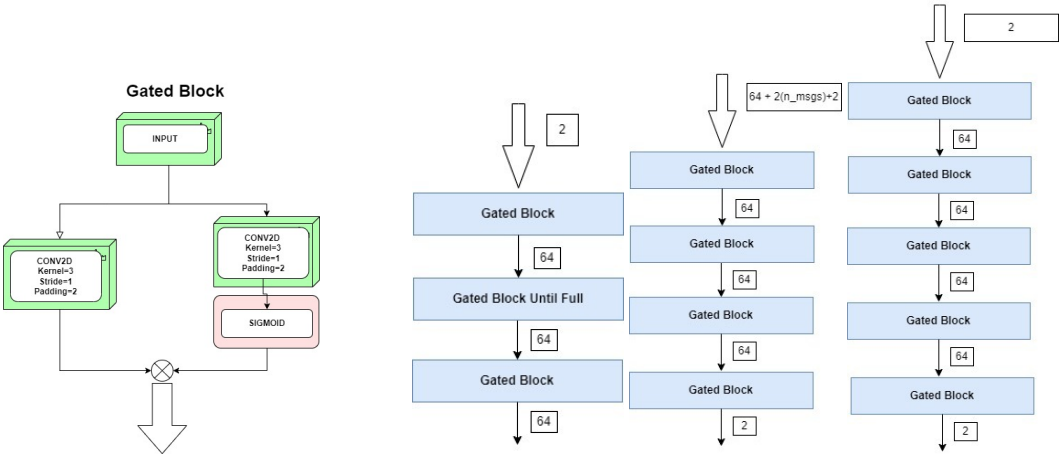
$$\mathcal{L}(C, M) = \lambda_C \cdot \text{MSE}\left(C, D_C(E(C, M))\right) + \lambda_m \cdot \sum \text{MSE}\left(M, D_M(D_C(E(C, M)))\right)$$

where MSE is the mean square error, and λ_c, λ_m are hyperparameters for tuning carrier and message losses.

$\boldsymbol{LossFunction}$ for conditional decoder

$$\mathcal{L}(C, M) = \lambda_C \cdot \text{MSE}\left(C, D_C(E(C, M))\right) + \lambda_m \cdot \sum \text{MSE}\left(M, D_M(D_C(E(C, M)), k)\right)$$

Here k is the condition or key that is used in case of a conditional decoder to find which enables multiple message decoding using a single decoder. Over here we concatenate each message with one-hot encoded tensor of the size of number of messages.



(a) Gated Block (b) Carrier Encoder (c) Carrier Decoder (d) Message Decoder

Figure 2: Architecture

6 Results

Here we present the results of our model. As it can be seen from the table below we obtain best results in the case of 1 hidden message. As we increase the number of hidden messages the Mean Square Error loss increases and the Signal to Noise ratio(SNR) decreases drastically. We use SNR to compare how intelligible our reconstructed audios are as compared to the original audios.

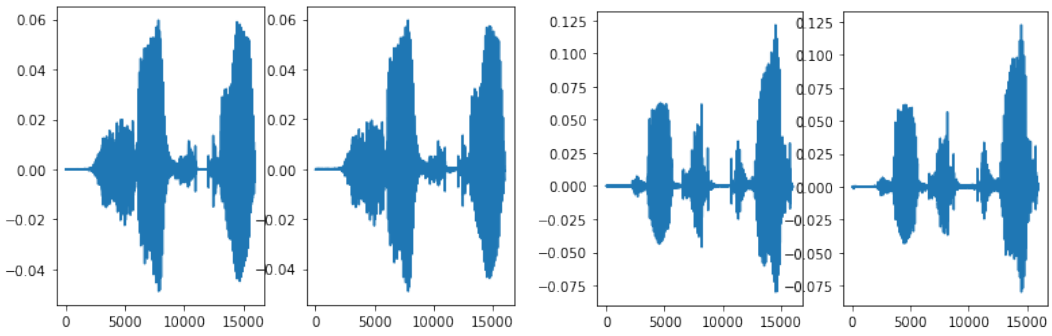
As evident from the waveform diagrams below the recovered carrier and message waveform and STFT looks identical to the original carrier and message respectively for 1 hidden message. As we increase the number of hidden message the difference between original vs reconstructed

waveform increases resulting in distortion of the reconstructed signals. This is expected as to accommodate multiple messages the network has to encode more information in the same carrier signal.

When comparing our results to our baseline[1], our experiment gives better result for 1 hidden message. For 3 hidden messages, our model has lower loss for both carrier and hidden messages. When we listen to the reconstructed hidden messages 2 out of 3 messages are intelligible but the third is muffled. Whereas in our baseline, the third message could also be heard clearly. For the case of 2 hidden message hidden conditionally using a single decoder the reconstructed carrier looks identical to the original carrier, the first hidden message is reconstructed with slight distortions but the second hidden message is has significant distortions. When we listen to the reconstructed messages the first is intelligible but the second is unintelligible.

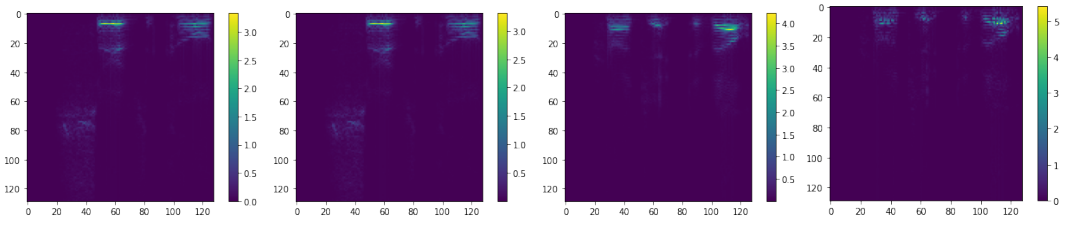
Table 1: Results

Case	Carrier loss	Carrier SNR	Message loss	Message SNR
1 Message	0.0001	24.32	0.0004	134.00
2 Message	0.0018	13.39	0.0113	5.66
3 Message	0.0009	16.46	0.0303	2.28
2 Messages - Conditional	0.0005	16.08	0.0126	3.83



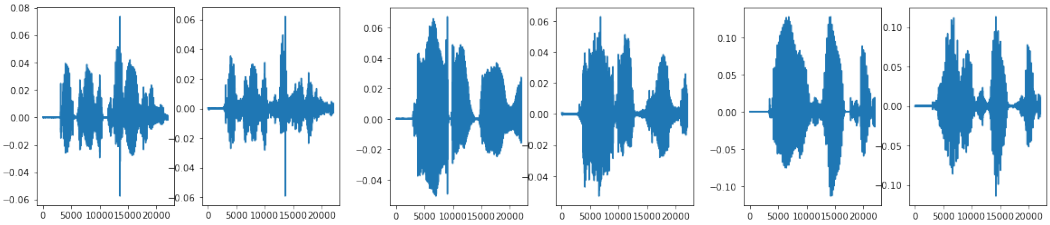
(a) Carrier (b) Message

Figure 3: 1 hidden message - Original vs Reconstructed Waveform



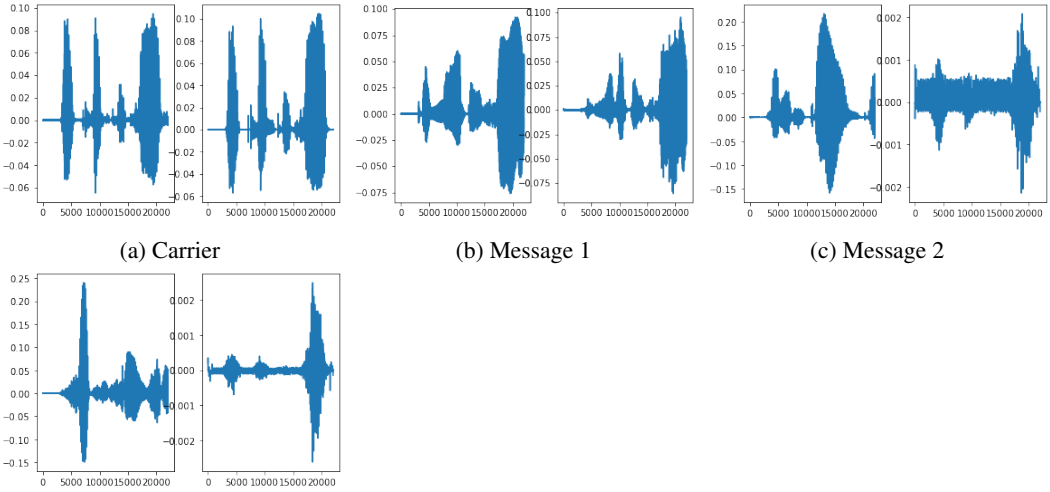
(a) Carrier Original (b) Reconstructed Carrier (c) Message Original (d) Reconstructed Message

Figure 4: 1 hidden message - Original vs Reconstructed STFTs



(a) Carrier (b) Message 1 (c) Message 2

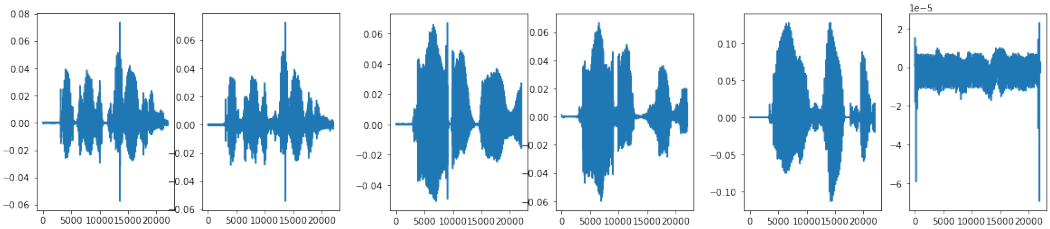
Figure 5: 2 hidden messages - Original vs Reconstructed Waveform



(a) Carrier (b) Message 1 (c) Message 2

(d) Message 3

Figure 6: 3 hidden messages - Original vs Reconstructed Waveform



(a) Carrier (b) Message 1 (c) Message 2

Figure 7: 2 hidden messages conditional decoder - Original vs Reconstructed Waveform

7 References

[1] Kreuk, Adi, Raj, Singh, and Keshet.Hide and Speak: Deep Neural Networks for Speech Steganography. In ReseachGate (2019)

[2] Dong, X., Bocko, M. F., and Ignjatovic, Z. Data hiding viaphase manipulation of audio signals. In Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on, volume 5, pp. V–377. IEEE, 2004.

[3] Shirali-Shahreza, S. and Shirali-Shahreza, M. Steganography in silence intervals of speech. In International conference on intelligent information hiding and multimedia signal processing, pp.

605–607. IEEE, 2008.

[4] Djebbar, F., Ayad, B., Meraim, K. A., and Hamam, H. Comparative study of digital audio steganography techniques. EURASIP Journal on Audio, Speech, and Music Processing, 2012(1):25, 2012.

[5] Xiaoxiao Dong, Mark F Bocko, and Zeljko Ignjatovic, "Data hiding via phase manipulation of audio signals," in Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on. IEEE, 2004, vol. 5, pp. V–377.

[6] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 2, pp. 236–243, 1984.

[7] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in European Conference on Computer Vision. Springer, 2018, pp. 682–697

[8] Peter Balazs, Dominik Bayer, Florent Jaillet, Peter Søndergaard, "The pole behavior of the phase derivative of the short-time Fourier transform, Applied and Computational Harmonic Analysis." Volume 40, Issue 3, 2016, Pages 610-621,

[9] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," NASA STI/Recon technical report n, vol. 93, 1993.

[10] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang and T. Sainath, "Deep Learning for Audio Signal Processing," in IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 2, pp. 206-219.

[11] Kreuk, Felix and Adi, Yossi and Raj, Bhiksha and Singh, Rita and Keshet, Joseph, "Hide and Speak: Towards Deep Neural Networks for Speech Steganography," arXiv:1902.03083, 2020.