

HEITOR RODRIGUES GUIMARÃES

ON SELF-SUPERVISED REPRESENTATIONS FOR  
3D SPEECH ENHANCEMENT

São Paulo  
2022

HEITOR RODRIGUES GUIMARÃES

**ON SELF-SUPERVISED REPRESENTATIONS FOR  
3D SPEECH ENHANCEMENT**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

São Paulo  
2022

HEITOR RODRIGUES GUIMARÃES

# Versão Corrigida

## ON SELF-SUPERVISED REPRESENTATIONS FOR 3D SPEECH ENHANCEMENT

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

Área de Concentração:

Sistemas Eletrônicos

Orientador:

Prof. Dr. Miguel Arjona Ramirez

Coorientador:

Prof. Dr. Wesley Beccaro


São Paulo  
2022

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 26 de Agosto de 2022

Assinatura do autor: Heitor R. Guimarães

Assinatura do orientador: 

#### Catálogo-na-publicação

Guimarães, Heitor  
ON SELF-SUPERVISED REPRESENTATIONS FOR 3D SPEECH  
ENHANCEMENT / H. Guimarães -- versão corr. -- São Paulo, 2022.  
75 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Speech Processing 2.Speech Enhancement 3.Representation Learning  
4.Representation Learning 5.Spatial Audio I.Universidade de São Paulo.  
Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

## ACKNOWLEDGMENTS

I want to thank my supervisor, Prof. Miguel Arjona, for trusting and accepting me as his student and for all the mentorship and conversations on various subjects. His professionalism and ability to inspire people are encouraging.

I also thank Prof. Wesley Beccaro, my co-supervisor and a friend in this academic journey. Your empathy and tranquility got me this far, and I am very grateful to you.

I thank my wife, Jessica Guimarães, who always encouraged me and is my life partner. Finally, I dedicate this work to my parents and grandparents, who are my examples of unconditional love. I thank them for all the sacrifices they made to make this possible.

I am grateful for all the support from Itaú – Unibanco and the time given to me to continue my studies and personal development. And I thank the friends I made there who made this journey more pleasant.

*“[...] the only people for me are the mad ones, the ones who are mad to live, mad to talk, mad to be saved, desirous of everything at the same time, the ones who never yawn or say a commonplace thing, but burn, burn, burn like fabulous yellow roman candles exploding like spiders across the stars and in the middle you see the blue centerlight pop and everybody goes “Awww!”*

-- Jack Kerouac, *On the Road*

# RESUMO

Métodos baseados em redes neurais profundas ganharam uma grande importância ao se mostrarem alternativas viáveis e poderosas para diversas tarefas, em especial para tarefas de processamento da voz, como reconhecimento de fala, detecção de palavras-chaves e reconhecimento de emoções. Entretanto esses métodos possuem alguns problemas intrínsecos, especialmente no que tange à robustez na presença de fatores deletérios, como ruídos e reverberação. Neste trabalho abordamos o problema de realce da voz, que tem como objetivo ser um sistema de pré-processamento capaz de realçar as características da voz e suprimir ruídos. Algoritmos baseados em modelos estatísticos abordam isto como um problema de maximização de verossimilhança. No entanto, não há garantias de que melhorará características perceptivas, como a inteligibilidade. Estudamos o uso de representações de fala extraídas do modelo *wav2vec* como função de custo perceptiva para a tarefa de realce da voz. Nossos experimentos demonstram que o uso de modelos de aprendizado contrastivo em funções de custo, para levar em conta características perceptivas, pode melhorar o desempenho do aprimoramento de fala em ambientes 3D. Além disso, discutimos o uso de modelos no domínio do tempo e do tempo-frequência. Nossos melhores resultados são obtidos através de modelos tempo-frequência, em detrimento do custo computacional.

**Palavras-Chave** – Processamento da Voz, Aprendizado de Representações, Aprendizado não-supervisionado, Realce da voz, Áudio Espacial.

# ABSTRACT

Methods based on deep neural networks have gained significant importance by showing viable and robust alternatives for several tasks, especially for speech processing, such as speech recognition, keyword spotting, and emotion recognition. However, these methods have inherent problems, especially regarding the robustness to detrimental factors, such as noise and reverberation. In this work, we tackle the Speech Enhancement problem, a pre-processing system capable of emphasizing the speech signal while suppressing noises. Statistical-model-based algorithms approach this as a likelihood maximization problem. However, there are no guarantees that it will improve perceptual characteristics such as intelligibility. We study the usage of speech representations extracted from the *wav2vec* model as a perceptual loss function for the Speech Enhancement task. Our experiments demonstrate that using contrastive learning models to consider high-level perceptual features in loss functions can improve the performance of 3D Speech Enhancement. Moreover, we discuss the usage of models in the time and time-frequency domain. Our best results are obtained through time-frequency models, increasing the computational cost.

**Keywords** – Speech Processing, Representation Learning, Unsupervised Learning, Speech Enhancement, Spatial Audio



# LIST OF FIGURES

1	Autoregressive Generative Model: The WaveNet architecture. Adapted from (OORD; DIELEMAN, et al., 2016). . . . .	22
2	Latent Variable Model: General Setting. . . . .	23
3	Generative Adversarial Network: General Setting. . . . .	25
4	The <i>wav2vec</i> model. Adapted from (SCHNEIDER et al., 2019). . . . .	26
5	Speech waveform (1034-121119-0042.wav) from L3DAS22 dataset corrupted by an additive keyboard stroke noise from FSD50K development set (155910.wav)	33
6	Speech waveform (1034-121119-0042.wav) from L3DAS22 dataset convolved with a RIR from the Aachen AIR database 1.4 (Aula Carolina Room). . .	34
7	L3DAS RIR positions: (a) 168 fixed points from a 3D Grid (b) 84 randomly selected positions (c) 2D projection view from above. Figures from (GUIZZO; MARINONI, et al., 2022) . . . . .	45
8	The architecture of the FCN for 3D speech enhancement in a reverberant environment: (a) FCN diagram block, and (b) the overall architecture. . .	47
9	The architecture of the FC2N for 3D speech enhancement in the L3DAS22 challenge: (a) Complex convolutional block, and (b) The overall architecture.	49
10	Log-frequency spectrogram of the 1993-147964-0008_A.wav audio file: (a) noisy speech; (b) enhanced speech and (c) clean speech. . . . .	53
11	Wideband spectrogram plotted with linear frequency scales: (a) noisy speech; (b) enhanced speech by the FC2N ( <i>wav2vec</i> 2.0), and (c) clean speech. In the noisy spectrogram, the red block on the left side represents a temporal gap filled due to the reverberation, and the red block on the right illustrates the sounds of two keystrokes. . . . .	55
12	Venn diagram for Mutual information and associated entropies. Adapted from Prof. Arjona classes (PSI5813). . . . .	63

# LIST OF TABLES

1	Ablation study on the effect of the loss functions to the metrics. . . . .	51
2	Performance on the development set of the task. Comparison of different approaches: Noisy, Wiener, FaSNet, SEWUNet, STOI-LF + PFPL ( $\alpha = 5$ ), and STOI-LF + PFPL ( $\alpha = 1000$ ). . . . .	52
3	L3DAS22: Performance on the development set of the task. Comparison of different models. . . . .	54
4	Ablation study on Losses for 3D Speech Enhancement: We fixed all the components (FC2N model) and studied the impact of different losses on multiple domains (time, time-frequency and perceptual). . . . .	56
5	Ablation study on the distance metrics for the <i>wav2vec2.0</i> representations in the $\mathcal{L}_{CPL}$ . The table shows the mean and standard deviation within samples. Fixed $\alpha = 10$ , except for the Wasserstein distance where $\alpha = 1$ . . . . .	58

# LIST OF SYMBOLS

$\theta$	Learnable weights from parametric models
$\phi$	Learnable weights from parametric models
$\mathcal{H}$	Hilbert Space
$d(\cdot, \cdot)$	Distance or divergence functions
$D_\phi(x)$	Composite function of the Discriminator with parameters $\phi$
$G_\theta(z)$	Composite function of the Generator with parameters $\theta$

# LIST OF ABBREVIATIONS

ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
CBAK	Composite Background Distortion measurement
CNN	Convolutional Neural Network
CRL	Contrastive Representation Learning
DFT	Discrete Fourier Transform
DNN	Deep Neural Networks
DQN	Deep Q-Networks
EM	Expectation-Maximization
ER	Emotion Recognition
FaSNet	Filter and Sum Network
FCN	Fully Convolutional Network
FC2N	Fully Complex Convolutional Network
FFT	Fast Fourier Transform
FSD50K	Freesound Dataset 50K
GAN	Generative Adversarial Networks
IC	Intent Classification
ISCA	International Speech Communication Association
ISTFT	Inverse Short-time Fourier Transform
KLD	Kullback–Leibler divergence
KS	Keyword Spotting
L3DAS	Learning 3D Audio Sources
LSTM	Long Short-Term Memory
LVM	Latent Variable Models
MAE	Mean Absolute Error
MLSP	IEEE International Workshop on Machine Learning for Signal Processing

MSE	Mean Squared Error
NLP	Natural Language Processing
PER	Phone Error Rate
PFPL	Phone-Fortified Perceptual Loss
PR	Phoneme recognition
RBM	Restricted Boltzmann Machines
ReLU	Rectified Linear Unit
RIR	Room Impulse Responses
RL	Reinforcement Learning
RoBERTa	Robustly Optimized BERT Pretraining Approach
SE	Speech Enhancement
SEWUNet	Speech Enhancement Wave-U-Net
SID	Speaker Identification
SLP	Spoken Language Processing
SLR	Speech Representation Learning
SSL	Self-supervised learning
STFT	Short-time Fourier Transform
STOI	Short-time Objective Intelligibility
SUPERB	Speech processing Universal PERFORMANCE Benchmark
UAE	Uncertainty Autoencoders
VAE	Variational Autoencoders
VQ-VAE	Vector Quantised-Variational AutoEncoder
WER	Word Error Rate

# CONTENTS

<b>1</b>	<b>INTRODUCTION</b>	<b>13</b>
1.1	Overview and Motivation . . . . .	13
1.2	Research Questions . . . . .	14
1.3	Outline . . . . .	16
<b>2</b>	<b>SPEECH REPRESENTATION LEARNING</b>	<b>17</b>
2.1	An Overview on Learning Machines . . . . .	17
2.2	Approaches for Learning Meaningful Representations . . . . .	20
2.3	How to evaluate the quality of a representation . . . . .	28
2.4	Review on Representation Learning Methods . . . . .	29
<b>3</b>	<b>PERCEPTUAL LOSSES FOR SPEECH ENHANCEMENT</b>	<b>32</b>
3.1	The Speech Enhancement Problem . . . . .	32
3.2	Review on Speech Enhancement Methods . . . . .	35
3.3	On Loss Functions for Speech Enhancement . . . . .	37
3.4	Compound Perceptual-Loss based on Speech Representations . . . . .	41
<b>4</b>	<b>METHODOLOGY</b>	<b>43</b>
4.1	Datasets . . . . .	43
4.2	Infrastructure and Software . . . . .	46
4.3	Proposed methods for 3D Speech Enhancement . . . . .	46
<b>5</b>	<b>RESULTS AND DISCUSSION</b>	<b>51</b>
5.1	IEEE MLSP 2021 Data Challenge . . . . .	51
5.2	IEEE ICASSP 2022 Data Challenge . . . . .	54

5.3	Supplementary studies on the L3DAS22 dataset . . . . .	55
<b>6</b>	<b>CONCLUSION</b>	<b>59</b>
	<b>Appendix A – Fundamentals of Information Theory</b>	<b>61</b>
	<b>Appendix B – Multi-Resolution STFT loss function - Setup</b>	<b>64</b>
	<b>Appendix C – Variational Autoencoders</b>	<b>65</b>
	<b>References</b>	<b>68</b>

# 1 INTRODUCTION

We introduce in this chapter the main problems associated with deep learning-based enhancement methods and the research questions derived from our observations. Next, we present the structure of our work.

## 1.1 Overview and Motivation

Deep learning techniques achieve state-of-the-art results in several domains such as automatic speech recognition, image classification, and emotion recognition. Usually, those tasks rely on a large amount of labeled data to train supervised models that are task-specific. However, this approach has some significant issues: It is data inefficient (LAKE et al., 2017), not robust against adversarial attacks (GOODFELLOW; SHLENS; SZEGEDY, 2014), and neither generalizes across domains (COBBE et al., 2019). Representation Learning constitutes an alternative where those algorithms struggle and expand to a new frontier of learning algorithms. Learning intrinsic speech features directly from the input distribution in a diverse set is key to ensuring generalization (BENGIO; COURVILLE; VINCENT, 2013). Furthermore, the latent representation is fed to semi-supervised algorithms that require fewer data to learn the patterns since the passed features are already meaningful.

Generalization is an essential task in machine learning. However, unseen variations of the data can lead to poor performance in systems, which is demonstrated in the OpenAI paper (COBBE et al., 2019). Furthermore, using the same environments to train and test a Reinforcement Learning agent can lead to overfitting. For instance, using the same context (CoinRun Game), a slight variation in the background color from a level can drastically increase the agent’s error. The same is true for speech recognition models. A different channel or background noise can hurt the entire system’s performance. In order to enable generalization, representation learning methods usually find independent and disentangled subspaces. In this context, if something shifts in our input distribution,



the other dimensions of our latent representation can still capture the other factors seen during training. Altogether, unlabeled data are easy to collect, and mixed conditions are usually presented in this type of data, helping our algorithm to capture a diverse set of features.

On the side of speech processing algorithms, speech enhancement (SE) is concerned with improving the intelligibility and quality of degraded speech (LOIZOU, 2013). Speech enhancement algorithms focus on reducing or even suppressing the noise. Usually, we apply SE algorithms for two objectives: (i) to improve the quality of sound that the final users will hear and make it more pleasant for them (which is a subjective measurement); (ii) or to improve the performance of a downstream speech processing system. There are a wide variety of applications in which it is desired to enhance speech, for example, communication and teleconferencing systems, where, in addition to the background noise, the speech can be affected by the reverberation of the room, or by the noise over the communication channel (BENESTY, 2018).

In general, when using statistical-model-based algorithms (e.g., Neural Networks), the algorithm relies on the problem of maximum likelihood estimation (LOIZOU, 2013), where the objective is to design a supervised estimator that receives as input a noisy audio signal and outputs an enhanced speech signal as close as possible to the target audio signal (clean speech signal). The problem is that this estimator usually relies on a regression-based loss function that can introduce artifacts and hurt the intelligibility of the final signal.

From the discussed issues, some modern SE algorithms attempt to introduce a perceptual component to the optimization problem to directly maximize the metric of interest (e.g., intelligibility). In this work, we investigate the usage of SE models in the time and time-frequency domains and how we can use speech representations extracted from large self-supervised models to guide the optimization process toward an increase in the perceptual metrics, especially those related to intelligibility.

## 1.2 Research Questions

First, we would like to introduce unsupervised representation learning techniques through three pillars: generative modeling, self-supervision, and contrastive learning. To objectively evaluate the performance of these representations, we use the learned representations as the objective function of SE deep models. Moreover, Yang et al. created

a unified performance benchmark called SUPERB to formalize Speech Representations evaluation through a set of tasks (YANG et al., 2021). In their work, it is widely used the definition of upstream model as a neural network responsible for learning useful representations for speech data, and downstream model a small network that uses the learned representations for domain-specific tasks, such as keyword spotting. A model with a good performance on the SUPERB benchmark indicates that representations related to high-level perceptual tasks, such as phonemes and emotions, should be extracted with the upstream model.

Further, we investigate the usage of such representations for perceptual losses in speech enhancement problems (minimize the distance between representations from the enhanced and the ground truth signals). Particularly, in this work we look into 3D speech enhancement using the Learning 3D Audio Sources dataset and try to answer the following questions:

- Q1 Can we use Fully-Convolutional Networks (FCN) for noise suppression and dereverberation? Can those models improve intelligibility metrics compared to the original noisy audios?
- Q2 How do FCN models on time or time-frequency domains handle spatial SE? Both on performance and computational cost.
- Q3 Does the choice of a loss function heavily impact on the final performance?
- Q4 Can the usage of unsupervised models as perceptual losses improve the training of speech enhancement systems?
- Q5 What is the impact of different similarity metrics used to compare the latent space?

As referenced below, we had the opportunity to disclose some results as research papers related to the speech enhancement problem.

1. H. R. Guimarães, W. Beccaro and M. A. Ramírez, “Optimizing Time Domain Fully Convolutional Networks for 3D Speech Enhancement in a Reverberant Environment Using Perceptual Losses”, IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP). 2021. DOI: 10.1109/MLSP52302.2021.9596103
2. H. R. Guimarães, W. Beccaro and M. A. Ramírez, “A Perceptual Loss Based Complex Neural Beamforming for AmbiX 3D Speech Enhancement”, ISCA Archive. 2022. DOI: 10.21437/L3DAS.2022-4

## 1.3 Outline

This thesis is organized as follows. Chapter 2 delves into the theory of learning machines and representation learning necessary to follow this work. Specifically, about representation learning, we want to define it and describe how to develop an algorithm to extract good representations. Hence, we discuss how to measure its effectiveness. Chapter 3 describes the speech enhancement problem and discusses what loss functions can be used to optimize our model and consider the perceptual components using unsupervised representation learning models. Chapter 4 discusses the methodology of the experiments related to our work, what datasets we use, software, infrastructure, and all its components to replicate our analysis. Chapter 5 brings our results associated with our proposed experiments on 3D speech enhancement. Finally, chapter 6 presents our conclusions, summarizes our findings, and discusses the necessary steps to develop our work further.

## 2 SPEECH REPRESENTATION LEARNING

Data representation is crucial for the success of modern machine learning systems. In order to efficiently solve a task using machine learning, it is necessary to feed the algorithm with disentangled explanatory factors extracted from data (BENGIO; COURVILLE; VINCENT, 2013). Being able to learn these type of representations is an important step towards artificial intelligence.

To illustrate how a representation can help to solve a task, consider the equations 2.1 and 2.2. In fact both equations lead to the same result, but the representation of the first requires more efforts to solve it. Changing the representation from Cartesian to Polar coordinate systems is useful for solving this problem.

$$Z = \int_{-1}^1 \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \sqrt{x^2 + y^2} dy dx \quad (2.1)$$

$$Z = \int_0^1 \int_0^{2\pi} r^2 d\theta dr \quad (2.2)$$

A successful method to capture useful factors is handcrafted feature engineering, in which an expert in the field could design features that represents the data (KUHN; JOHNSON, 2019), similar to what was done in the equations above. However, such a method does not scale or generalise across domains (BENGIO; COURVILLE; VINCENT, 2013).

In the following sections, we will introduce what is unsupervised representation learning, why it is an important theme, define what can be considered a good representation and discuss some methods to achieve such good representation.

### 2.1 An Overview on Learning Machines

Cognition is the process of extracting knowledge and understanding from sensors (sense) and signals (experience) embedded into a context. We define learning as the

process when knowledge entails a change of associations, also known as brain representations. This work lies in the field of machine learning, which aims to develop algorithms that enable machines to learn from data.

Having intelligent machines is not a new aspiration in philosophy and science. Furthermore, replicating the human process of cognition seemed the correct answer for years to achieve this goal. One of the first models to try to explain cognition was Associationism, introduced by Plato. In this theory, our ability to think and infer is based on associations. For instance, a *function* is defined as the association of inputs to outputs, which could be an object in the Associationism theory. However, when designing a model to emulate the cognitive process, one of the fundamental problems related to Associationism is the structure where associations are stored.

In 1873, Alexander Bain, in his work *Mind and body* (BAIN, 1873), introduced the idea of Connectionism. In early 1800, humanity started to understand the brain as a mass of interconnected neurons. On top of this, Bain assumed that information is stored in the connections of the neuronal network. In his early studies, he speculated that a neural grouping should exist where neurons excite and stimulate each other based on signals, and different intensities of activation could provide different outputs in the same route. When we think about modern machine learning techniques, we understand that Neural Networks are essentially Connectionist Machines. With the recent progress in biology today, we can estimate that our brain has 8 billion neurons and 100 trillion connections. For comparison, at present, the GPT-3 (BROWN et al., 2020) is one of the largest neural networks and has 175 billion connections.

In his book, Tom Mitchell gives a formal succinct definition for what machine learning means (MITCHELL, 1997): “A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ ”.

To elucidate this definition, consider the task  $\mathbf{T}$  of emotion recognition. The experience  $\mathbf{E}$  is the labeled dataset containing speech utterances and a label annotated with emotions. The performance  $\mathbf{P}$  of the algorithm is usually a classification measurement in this case. For example, it could be accuracy or the F-score. An effective learning machine should extract patterns to estimate what should be the associated class, i.e.,  $p(T|E)$ .

When discussing types of learning, usually we find three concepts: Supervised Learning, Reinforcement Learning, and Unsupervised Learning. The main feature of supervised learning is to have labels associated with our data. For example, let  $X$  and  $Y$  be a set

of features and labels. The job of the supervised model is to construct a mapping from  $X \rightarrow Y$  to estimate the conditional probability  $p(Y|X = \mathbf{x}; \theta)$ , where  $x \in X$  and  $\theta$  are the parameters of the model. Our objective is to minimize wrong estimations as much as possible. To achieve this, we usually rely on the maximum likelihood estimation (MLE) method to estimate the parameters  $\theta$  of the model, based on a probability distribution from the input data, maximizing the likelihood function.

On the other hand, we have a class of algorithms under the category of Reinforcement Learning (RL). In this setting, we do not have labeled samples, but we do obtain those as feedbacks from signals (Reward function) based on the interaction (Action) of our algorithm (Agent) with the world (Simulation). The reward function is like a teacher who can identify good behaviors without telling what actions the agent should take. The agent generates his training data in the RL setup by interacting with the world.

In recent years, the usage of Neural Networks for image classification represented a breakthrough in the field of computer vision, and artificial intelligence (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; SIMONYAN; ZISSERMAN, 2014; HE et al., 2016). On the other hand, in academic and industrial applications, we can see a clear tendency to reduce machine learning to a pipeline where we acquire large amounts of data, construct deeper models, and acquire better hardware. However, this pipeline is counter-intuitive to how a human learns. For instance, the performance of humans against Deep Q-Networks (DQNs) in the Atari “Frostbite” game is quite similar. However, an average person learns the rules to play the game in 2 hours, and the algorithm takes close to 1000 hours. Also, small changes in the game, such as changing the color of the blocks, lead to an unuseful algorithm (LAKE et al., 2017).

For the last, we have Unsupervised Learning. In this scenario, we do not have any signals (i.e., additional information such as labels or rewards) related to our input as optimization guidance. Unsupervised learning is all about data representation. The key idea is to extract the underlying structure of data. Understanding the structure of data means that we can group similar samples, density estimation, and perform operations such as dimensionality reduction (find orthogonal spaces that express the features of our data). Usually, the methods rely on transforming the input space to another where the features are disentangled by using Kernel machines (DOMINGOS, 2020). Therefore, the notion of similarity or distance in the given space is crucial for the success of our methods.

One of the first class of algorithms to introduce the idea of representation learning are the **Kernel methods**. Those are algorithms that apply non-linear functions to data to

learn meaningful representations through a basis expansion (HOFMANN; SCHÖLKOPF; SMOLA, 2008). This new space derived from the mapping has a structure that uses linear models to solve the tasks.

Let  $\mathcal{X} \subset \mathbb{R}^D$  be a non-empty set representing our data (**attributes**) and the labels (**targets**) are given by  $\mathcal{Y} \subset \{-1, 1\}$ . We define a sample as  $\{(x_i, y_i)\}_{i=0}^m \in \mathcal{X} \times \mathcal{Y}$ . A **feature map**  $\phi$  is a function that maps the attribute space to a high dimensional dot-product space  $\mathcal{H}$ . A **Kernel**  $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$  is a similarity measure between samples, written as  $k(x, z) = \langle x, z \rangle$ . The kernel definition also lets us construct algorithms in the dot-product space, as described by equation 2.3. This property is known as **kernel trick**.

$$k(x, z) \triangleq \langle \phi(x), \phi(z) \rangle \quad (2.3)$$

In the context of machine learning it is helpful to restrict ourselves in the case of positive definite kernels, which leads us to a function space called *reproducing kernel Hilbert space* (RKHS). Hilbert spaces allow us to have infinite-dimensional kernel spaces. However, the **representer theorem** lets us reduce this to a tractable finite-dimensional problem.

Our interest in this work is on DNN-based models for Unsupervised Representation Learning and how to use it these learnt representations for downstream tasks, such as speech enhancement.

## 2.2 Approaches for Learning Meaningful Representations

At the core of this work, our goal is to use high-level features created from connectionist machines capable of learning explanatory and disentangled factors from audio data in an unsupervised fashion. Later, we use those representations to train Speech Enhancement algorithms that consider perceptual characteristics in their optimization process and be more efficient. This section describes three training schemes for unsupervised representation learning that can help us achieve our goal.

### 2.2.1 Generative Modeling

We describe generative modeling as the task of approximating the underlying data distribution from a finite set of samples. Later the learned model can be used for down-

stream inference tasks, such as sampling from the distribution to create new data points (Hence the name generative).

Let  $\mathcal{D}$  be our finite set of samples, and  $x^{(j)} \sim p_x$ , for  $j = 1, 2, \dots, |\mathcal{D}|$ , a sample from this  $n$ -dimensional data set. Our goal is to design a learning machine for the parameters  $\theta$  from a model family  $\mathcal{M}$  in such a way that  $p_\theta$  is as close as possible from  $p_x$ , as described in the equation 2.4.

$$\hat{\theta} = \underset{\theta \in \mathcal{M}}{\operatorname{argmin}} d(p_x, p_\theta), \quad (2.4)$$

where  $d(\cdot)$  is a distance between the probability distributions.

It is interesting to notice that we are handling probabilistic models on high-dimensional data, and our model is trying to learn the joint distribution over  $\mathcal{D}$ . Density estimation and outlier detection are some of the tasks where generative models can be applied. In this work, we are especially interested in the application of generative models for Representation Learning.

We have different approaches to construct deep generative models. The first we will talk about is **autoregressive models**. In this approach, we model the joint distribution as the product of the conditional distributions obtained after applying the chain rule of probability, as described in equation 2.5. This formula can also be viewed as a Bayesian Network (ERMON; SONG, 2021).

$$p(\mathbf{x}) = \prod_{i=1}^n p(x_i | x_1, x_2, \dots, x_{i-1}) \quad (2.5)$$

This approach has the benefit of being easy to model, and no sampling is required during training time. However, the generation process for the next element in the sequence is very slow, and this type of model is better at modeling local structure than global structures (MNIH, 2020). In this setting the model is trained to maximize the likelihood.

In the context of speech processing, an example of generative autoregressive model in the time-domain is the **WaveNet** (OORD; DIELEMAN, et al., 2016), a neural network capable of synthesizing high-quality speech. The sampling procedure is conditioned on all previous samples, and the model is trained on thousands of samples per second of audio. The generated sample is fed back into the model to predict the next sample. To implement this, the WaveNet relies on dilated causal convolutions, which means that no future information is used to generate a sample at time  $t$ .



The benefit of using a causal convolution instead of a recurrent connection such as Recurrent Neural Networks (RNN), or Long short-term memory (LSTM), is that convolutions are faster to train. On the other hand, to capture long-range information, it is necessary to have multiple layers, increasing the complexity of the model. To solve this, Oord et al. proposed the usage of dilated convolutions, where we apply the filters over a larger receptive filter. For a more concrete understanding, the figure 1 illustrates the WaveNet model as described here.

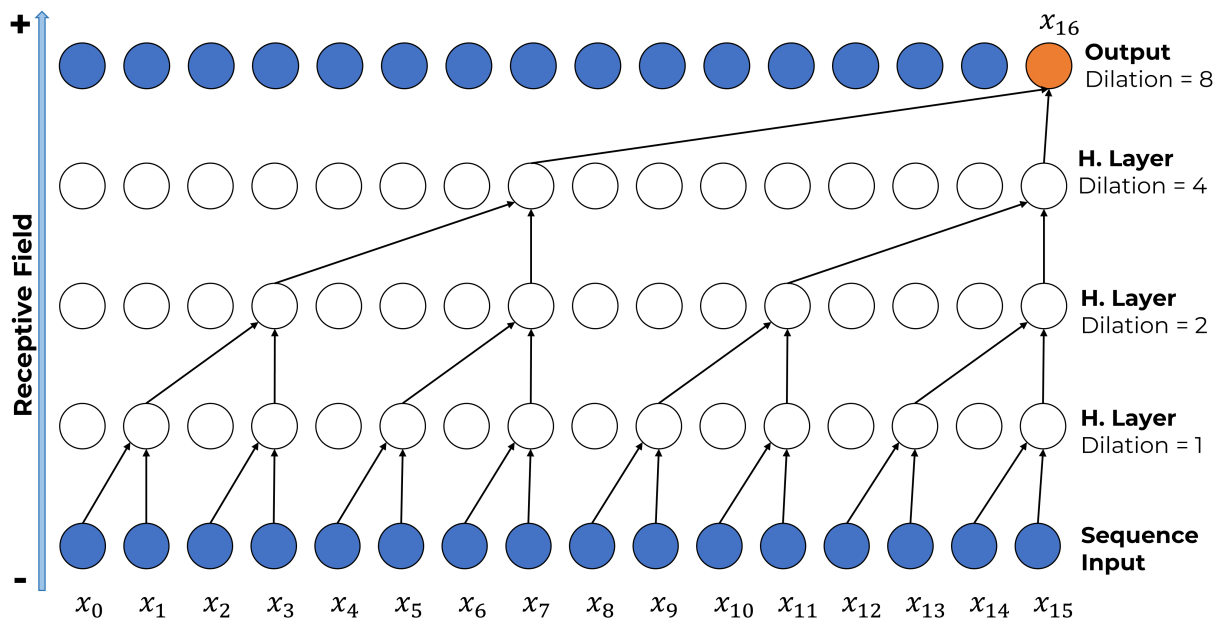


Figure 1: Autoregressive Generative Model: The WaveNet architecture. Adapted from (OORD; DIELEMAN, et al., 2016).

It is also possible to create a conditional model based on some characteristics (e.g., speaker identity) as shown in equation 2.6. In this case, we are not anymore in an unsupervised setting, and the WaveNet model conditioned on speaker identity is used to generate samples from a specific individual,  $\mathbf{h}$ .

$$p(\mathbf{x}|\mathbf{h}) = \prod_{i=1}^n p(x_i|x_1, x_2, \dots, x_{i-1}, \mathbf{h}) \quad (2.6)$$

The second approach to train generative models is **latent variable models** (LVM). The key idea is the introduction of a latent variable,  $\mathbf{z}$ , that defines a distribution over the observations and may be read as an explanation for the data sample, therefore as a carrier of meaningful information (representation). In figure 2, we have the general schematics for the LVM.

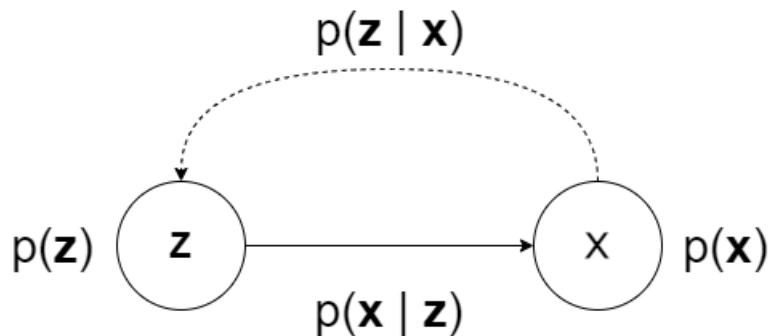


Figure 2: Latent Variable Model: General Setting.

Now we need to introduce some notations for further development. We call  $p(\mathbf{z})$  the **prior distribution** of the latent variable, and  $p(\mathbf{x}|\mathbf{z})$  the **likelihood** that maps latent variables to observations. It is interesting to notice that with these two distributions we define the **joint distribution**  $p(\mathbf{x}, \mathbf{z})$  as described in equation 2.7, and we have the model fully characterized. The **posterior distribution**  $p(\mathbf{z}|\mathbf{x})$  are the possible latent variables that could generate the given observation. The last concept is the **marginal likelihood**  $p(\mathbf{x})$ , which describes the data distribution.

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) \quad (2.7)$$

The process to generate data is quite simple with this model: We sample latent representations from the prior  $\mathbf{z} \sim p(\mathbf{z})$ , and then generate from the likelihood as  $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$ . In our work, we are concerned with **inference**, which is the inverse process of mapping from the data to meaningful latent representations.

From the definition of conditional probability, we can derive how to compute  $p(\mathbf{z}|\mathbf{x})$ , as described in equation 2.8. The problem associated with this process is how to efficiently compute the  $p(\mathbf{x})$ .

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x}, \mathbf{z})}{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}} \quad (2.8)$$

To solve further the equation 2.8, as in autoregressive models, we rely on maximum likelihood estimation (MLE) to learn a model that approximates the marginal likelihood. Based on the hypothesis that our data is independent and identically distributed (i.i.d.), we define the likelihood function as the product of probabilities of data points. For computational efficiency, we use a log-transformation in this product and maximize the

sum of log-probabilities, as described in equation 2.9. To solve this, we use iterative approaches such as Gradient Descent or Expectation-Maximization (EM).

$$\hat{\theta} = \operatorname{argmax}_{\theta \in \mathcal{M}} \sum_{\mathbf{x} \in \mathcal{D}} \log p_{\theta}(\mathbf{x}) \quad (2.9)$$

An important step towards the optimization problem is how to compute the gradient of  $\log p_{\theta}(\mathbf{x})$ . From the equation 2.10, we observe that it is necessary to solve the sub-problem of computing the posterior distribution to estimate the gradient.

$$\begin{aligned} \nabla_{\theta} \log p_{\theta}(\mathbf{x}) &= \frac{\nabla_{\theta} p_{\theta}(\mathbf{x})}{p_{\theta}(\mathbf{x})} \\ &= \frac{\int \nabla_{\theta} p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z}}{p_{\theta}(\mathbf{x})} \\ &= \frac{\int p_{\theta}(\mathbf{x}, \mathbf{z}) \nabla_{\theta} \log p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z}}{p_{\theta}(\mathbf{x})} \\ &= \int p_{\theta}(\mathbf{z}|\mathbf{x}) \nabla_{\theta} \log p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z} \\ &= \mathbb{E}_{p_{\theta}(\mathbf{z}|\mathbf{x})} [\nabla_{\theta} \log p_{\theta}(\mathbf{x}, \mathbf{z})] \end{aligned} \quad (2.10)$$

That leads us to a trade-off between tractable models, in which we can make an exact inference process, and intractable models, in which we need some approximate inference to train the model. Tractable models (e.g., Mixture and Invertible models) usually have a straightforward training process because it is easy to do inference, but they are less powerful. Intractable models are more expressive but rely on hard assumptions about the data or latent distributions. One of the most popular methods of intractable models is the Variational Autoencoder (VAE) and more details can be found at appendix C.

The last approach is **implicit models**, where the most representative member of this class is the Generative Adversarial Networks (GANs) (GOODFELLOW; POUGET-ABADIE, et al., 2014). Unlike autoregressive or LVM, these models are not trained with maximum likelihood, here instead we use adversarial training.

In this setting, we have two components that constitute the GANs: The Generator  $G_{\theta}$ , and the Discriminator  $D_{\phi}$ . The task of the Generator is to generate, from a random variable  $\mathbf{z} \sim p(\mathbf{z})$ , a realistic sample that is as similar as possible to  $\mathbf{x} \sim p(\mathbf{x})$ . Determining whether this sample came from the original dataset or is a fake example from the Generator is the primary goal of the discriminator.

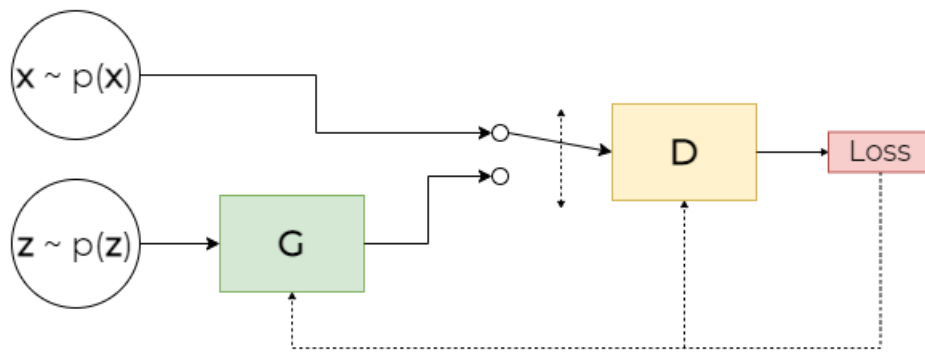


Figure 3: Generative Adversarial Network: General Setting.

The described dynamic is a minimax game, where the Generator’s goal is to fool the Discriminator. Statistically, we can see this objective as a two-sample test (ERMON; SONG, 2021). Let  $S_1 = \mathcal{D} = \{\mathbf{x} \sim p(\mathbf{x})\}$  and  $S_2 = \{\mathbf{x} \sim p_\theta\}$ . We accept the null hypothesis ( $p(\mathbf{x}) = p_\theta$ ) when the difference between  $S_1$  and  $S_2$  is less than a threshold  $\alpha$ . Therefore, the Generator tries to minimize the test objective while the Discriminator tries to maximize, as described by the operation 2.11 below.

$$\min_{\theta} \max_{\phi} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\log (D_{\phi}(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log (1 - D_{\phi}(G_{\theta}(\mathbf{z})))] \quad (2.11)$$

Currently, GANs are successful in several domains where realistic synthesis is a must. On the other hand, GANs are hard to set up the training procedure for a stable optimization process, and this type of model suffers from mode collapse, in which the generator repeatedly generates a sample from a class (e.g., generate speech samples from a single speaker).

## 2.2.2 Self-Supervised Learning

Self-supervised learning (SSL) is the most popular method to learn good features from unlabeled data. One of the critical aspects of SSL is that we obtain supervisory signals from data itself by exploiting knowledge of the data modality. The most popular method for supervision is to predict an unobserved part of the input. For instance, this is the mechanism present in state-of-the-art models for natural language processing (NLP) such as BERT (DEVLIN et al., 2018) and RoBERTa (LIU et al., 2019).

While generative modeling is interested in sampling diverse and high-quality samples that resemble the original data distribution, SSL focuses on extracting meaningful representations. For example, language models receive a short text with a few masked words

as input. The task is to predict this masked word, and by doing this, the system learns features related to the word context (DEVLIN et al., 2018).

It is interesting that generative models are self-supervised models, but usually with different goals. For example, in the generative modeling setting, we are interested in the model capability to generate high-quality and diverse samples, while SSL is concerned with obtaining disentangled explanatory factors from data to later use on multiple downstream tasks (LECUN; MISRA, 2021).

### 2.2.3 Contrastive Learning

Contrastive Representation Learning (CRL) is one of the most prominent techniques for SSL. In this methodology, we use a dissimilarity metric to arrange related samples close to each other in the latent space. All CRL models are SSL models but with special loss functions.

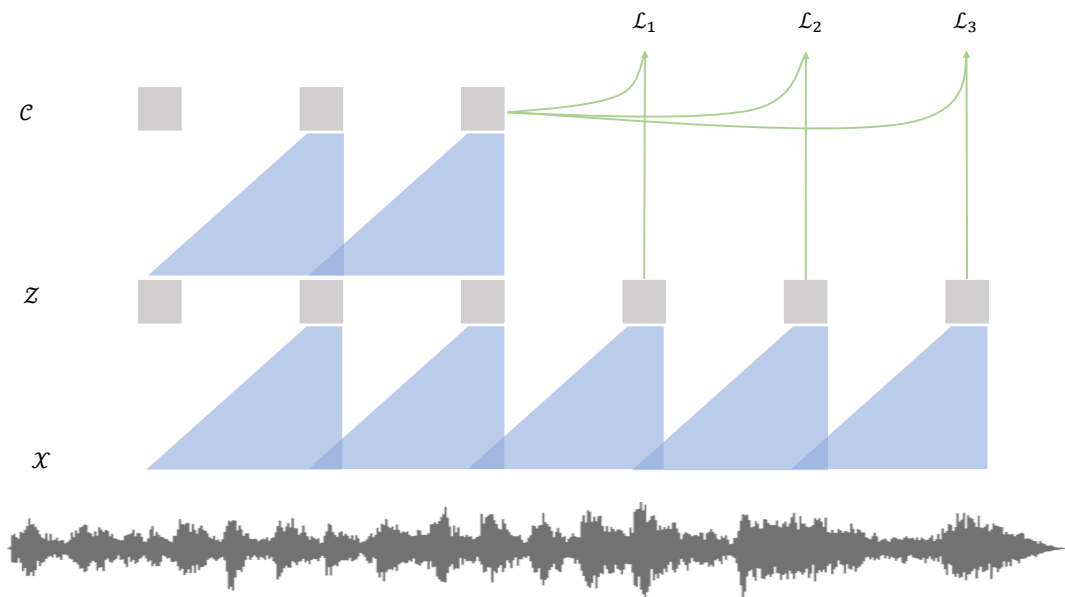


Figure 4: The *wav2vec* model. Adapted from (SCHNEIDER et al., 2019).

To further investigate the usage of CRL, in this section, we are going to discuss the *wav2vec* (SCHNEIDER et al., 2019) and *wav2vec2.0* (BAEVSKI et al., 2020) models. The *wav2vec* model is an unsupervised pre-training technique to extract meaningful representations from 16 kHz raw audio and feed these representations to acoustic models for speech recognition systems instead of the usual filterbank features. In high-dimensional signal modeling, predicting the next step is often a task associated with understanding

the local smoothness of the signal. However, to model long-range temporal attributes, it is necessary to infer more global features of the signal that are useful for more challenging tasks (e.g., phonemes and intonation for ASR, and Emotion Recognition) (OORD; LI; VINYALS, 2018). This idea combined with contrastive loss is the base of the *wav2vec* model.

The *wav2vec* model consists of two fully convolutional neural networks stacked on top of each other, as shown in diagram 4. The first one is the encoder network, a mapping  $f$  from the raw waveform sample  $x_i \in \mathcal{X}$  to an intermediate representation  $z_i \in \mathcal{Z}$ , in the form  $f : \mathcal{X} \mapsto \mathcal{Z}$ . The encoder network comprises five blocks of causal convolutions with 512 feature maps, a group normalization, and a ReLU activation function, with kernel and stride sizes varying per layer. The downsampling factor of the encoder network makes each representation  $z_i$  capture 10 ms of the speech signal.

The next important component is the context network, a mapping  $g : \mathcal{Z} \mapsto \mathcal{C}$ . The key idea of this network is to map different features ( $z_i \dots z_{i-v}$ ) into a single context vector  $c_i = g(z_i \dots z_{i-v})$ , considering a receptive field of size  $v$ . This network has nine layers composed of the same blocks of the encoder network. The representations from  $\mathcal{C}$  are the input features for downstream tasks later.

The contrastive characteristic takes place in the loss function of the method. The idea is to compare samples (hence the name contrastive), and distinguish between the sample  $z_{i+k}$  that is  $k$  steps into the future from ten distractors sampled from a uniform distribution  $p_n$ . Our goal is to minimize the loss  $\mathcal{L}_k$ , as shown in equation 2.12. The final loss function is the sum of all  $k = 1, \dots, K$  steps. The  $\lambda$  factor is set to the relative number of negative samples, and  $\sigma$  is the sigmoid function.

$$\mathcal{L}_k = - \sum_{i=1}^{T-k} \left( \log \sigma(\mathbf{z}_{i+k}^\top h_k(\mathbf{c}_i)) + \lambda \mathbb{E}_{\tilde{\mathbf{z}} \sim p_n} [\log \sigma(-\tilde{\mathbf{z}}^\top h_k(\mathbf{c}_i))] \right) \quad (2.12)$$

Using the representations from  $\mathcal{C}$ , it is possible to train models for downstream tasks using less labeled data while keeping good performance. This setup was one of the breakthroughs introduced by Schneider (SCHNEIDER et al., 2019).

The authors modify two significant components in *wav2vec2.0* model. The model uses the same encoder network of the *wav2vec* model, a convolutional network with five layers. On top of those features, the authors proposed the usage of the Gumbel-Softmax distribution (JANG; GU; POOLE, 2016) to extract discrete features, which we obtain through a property, a “temperature parameter”, so that we smoothly anneal to reshape

the continuous distribution to a categorical one.

In addition to this modification, the previous context network is replaced by a 12-layer Transformer network for the base model and a 24-layer network for the large model. Furthermore, the loss function in this new model is modified to include both the contrastive loss, similar to the previous work, and also a component related to the diversity loss of the learned codebook of the quantization module. This is done using the entropy maximization of the Gumbel-Softmax distribution, vital to encourage the model to use as many as possible different codes from the codebook instead of collapsing into a small subgroup.

## 2.3 How to evaluate the quality of a representation

There are different approaches to understanding if a latent representation is functional. The most popular is semi-supervised learning, in which we feed the learned representation to a set of downstream tasks related to the data modality. When using this approach, we aim for data efficiency and generalization. Another important aspect of this methodology is that we are now performing a finetuning process and therefore we need to feed the algorithm with substantially less labeled data than a usual supervised learning pipeline.

For instance, the *wav2vec 2.0* model (BAEVSKI et al., 2020) is trained as a contrastive task using 960 h of unlabeled data from the LibriSpeech dataset (PANAYOTOV et al., 2015) and learns to produce high-level latent representations which, among other factors, capture rich phonetic information (HSIEH et al., 2020). The finetuning process is done on labeled subsets of clean data with 100 hours, 10 hours, 1 hour, and may even be reduced to 10 minutes of labeled data for an automatic speech recognition (ASR) task.

Another possible evaluation is model analysis, in which our focus is on interpretable machine learning, in which we try to understand what the model learns. Depending on the use case, this branch is critical due to regulatory policies, such as in the banking industry.

### 2.3.1 A Performance Benchmark for SLR

In this work, we focus on semi-supervised learning. Even though this is a widespread methodology, the challenge associated with it was the lack of a benchmark to compare its methods against the literature. To handle it, the *Speech processing Universal PERFORMANCE Benchmark* (SUPERB) was proposed by Yang (YANG et al., 2021). The idea is

to unify a set of experiments related to speech processing and compare multiple models, developing a tool for other researches. For instance, based on this tool we describe in more details five different downstream tasks that are considered the basis of SUPERB: **Content tasks** (Phoneme Recognition and Keyword Spotting), **Speaker task** (Speaker Identification), **Semantic task** (Intent Classification) and **Paralinguistics task** (Emotion Recognition).

Phoneme recognition (PR) is a task of transcribing an utterance into phonemes. Therefore, the SUPERB includes an alignment modeling to avoid inaccurate forced alignment. The Phone Error Rate (PER) metric is used to evaluate the model performance. The other content-related task is keyword spotting (KS). It is a classification problem where we detect preregistered keywords in a set of ten classes. For this task, the evaluation metric is the standard accuracy. KS is also an essential task for on-device machine learning, and model size can be another critical factor.

The speaker identification task (SID) is a multi-class classification problem where we aim to identify the speaker in each utterance. We have a fixed set of speakers in the SUPERB benchmark that appears for both training and test. To evaluate the model, we use the accuracy metric.

We have the intent classification (IC) on the semantic task, which is the task of assigning a label to an utterance into predefined classes. It is a multi-class classification problem with three labels: action, object, and location. Again, this problem is evaluated using the accuracy metric.

Emotion recognition (ER) is a paralinguistic classification problem with, in general, four different classes: neutral, happy, sad, angry. Our predictors are the latent representations of each utterance, and our model’s evaluation metric is accuracy.

## 2.4 Review on Representation Learning Methods

SLR is currently a hot topic in the machine learning community. The early studies of statistical methods for representation learning can be associated with Kernel Methods. Kernel methods are algorithms that apply non-linear functions to data to learn meaningful representations through a basis expansion (HOFMANN; SCHÖLKOPF; SMOLA, 2008). This new space derived from the mapping has a structure that allows us to use linear models to solve the tasks.

The usage of neural networks for representation learning defined a breakthrough in



the field. Restricted Boltzmann Machines (RBM) is an energy-based model based on the Boltzmann distribution, a key concept in statistical mechanics that allows us to compute the probability of an energy-state given an initial condition. This method was used in (HINTON; SALAKHUTDINOV, 2006) as a pretraining method.

Autoencoders are also popular unsupervised methods for representation learning in which we learn continuous latent representations that capture the knowledge from the original input. Gosh *et al.* (GHOSH *et al.*, 2016) used Stacked Denoising Autoencoders as a pretraining for categorical emotion recognition and achieved compatible results compared with state-of-the-art models.

Different strategies were proposed to learn meaningful representations. Contrastive learning is one of the most popular approaches in which similar samples are close to each other in the latent space than dissimilar ones. As above mentioned, the *wav2vec* (SCHNEIDER *et al.*, 2019) is a self-supervised CNN-based model that takes raw audio as input and computes a latent representation that can be input to a speech recognition system.

Transformers and methods such as BERT (DEVLIN *et al.*, 2018) have gained attention in the NLP community. HuBERT (HSU *et al.*, 2021) is a model that encodes unmasked speech waveforms to continuous latent representations, which maps them to an acoustic modeling problem. The next step in the algorithm is to learn how to capture long-range temporal relations in order to reduce the prediction error. The second part of the algorithm is related to a language model.

Continuous representation is not the only option. Oord *et al.* (OORD; VINYALS, *et al.*, 2017) demonstrated the usage of Vector Quantization on VAE (VQ-VAE) and introduced the idea of using an embedding layer as a discrete latent space in the Variational Autoencoder with the addition of a skip-connection like operation between the encoder and decoder to allow the backpropagation mechanism to work even though a discrete space introduces a non-differentiable operation.

Based on this concept, Chorowski, Oord *et al.* (CHOROWSKI *et al.*, 2019) proposed the usage of VQ-VAE with a WaveNet (OORD; DIELEMAN, *et al.*, 2016) decoder. The learned discrete representations can capture phonetic content and are later used for phoneme recognition with a good performance.

Finally, Uncertainty Autoencoders (UAE) have been shown to be a promising technique against VAE. UAEs were introduced in the context of compressed sensing and in their work, Grover and Ermon (GROVER; ERMON, 2019) demonstrated formal and ex-

perimentally that UAE is also a generative model and can be a technique for unsupervised representation learning where the compressed measurements are latent representations. One of the biggest advantages of UAE against VAE is that UAE has no explicit prior distribution over the latent space and therefore we do not need a KL divergence regularization term in the loss function. This is an advantage because it ensures that the network is learning a good representation even in the presence of powerful distribution decoders.

### 3 PERCEPTUAL LOSSES FOR SPEECH ENHANCEMENT

In this chapter, we formalize the speech enhancement problem, first for a general case and then for the spatial scenario, discussing current methods to tackle 3D speech enhancement. Finally, we describe the most popular loss functions for statistical-model-based speech enhancement.

#### 3.1 The Speech Enhancement Problem

When recording audio, it is usual that the information captured by the microphone is corrupted by noises, environmental or artificial sounds, that degrade the quality and intelligibility of speech. Similar to (LOIZOU, 2013), we define the speech enhancement task as a system capable of isolating the speech signal, improving some perceptual metric, while reducing noises to a minimum and including a minimum amount of processing artifacts.

In this work we only consider additive noises. Let  $x$  be a clean speech signal, and  $s$  the noise signal. We define the noisy waveform in equation 3.1.

$$y[n] = x[n] + s[n] \tag{3.1}$$

There are multiple sources of noise signals that constitute  $s[n]$ . This signal can be viewed as a stationary signal, whose characteristics do not change over time (e.g., a mechanical fan), or as a non-stationary (e.g., typing or background speech). Figure 5 shows a spectrogram of a clean speech signal, the isolated noise of typing in a computer keyboard, and both at a signal-to-noise ratio (SNR) of 5 dB.

From this spectrogram, in the middle image, we can observe a vertical pattern of high energy across multiple frequencies when a person stroke the keyboard. There is also a narrow band white noise around the 1 kHz frequency related to the recording procedure.

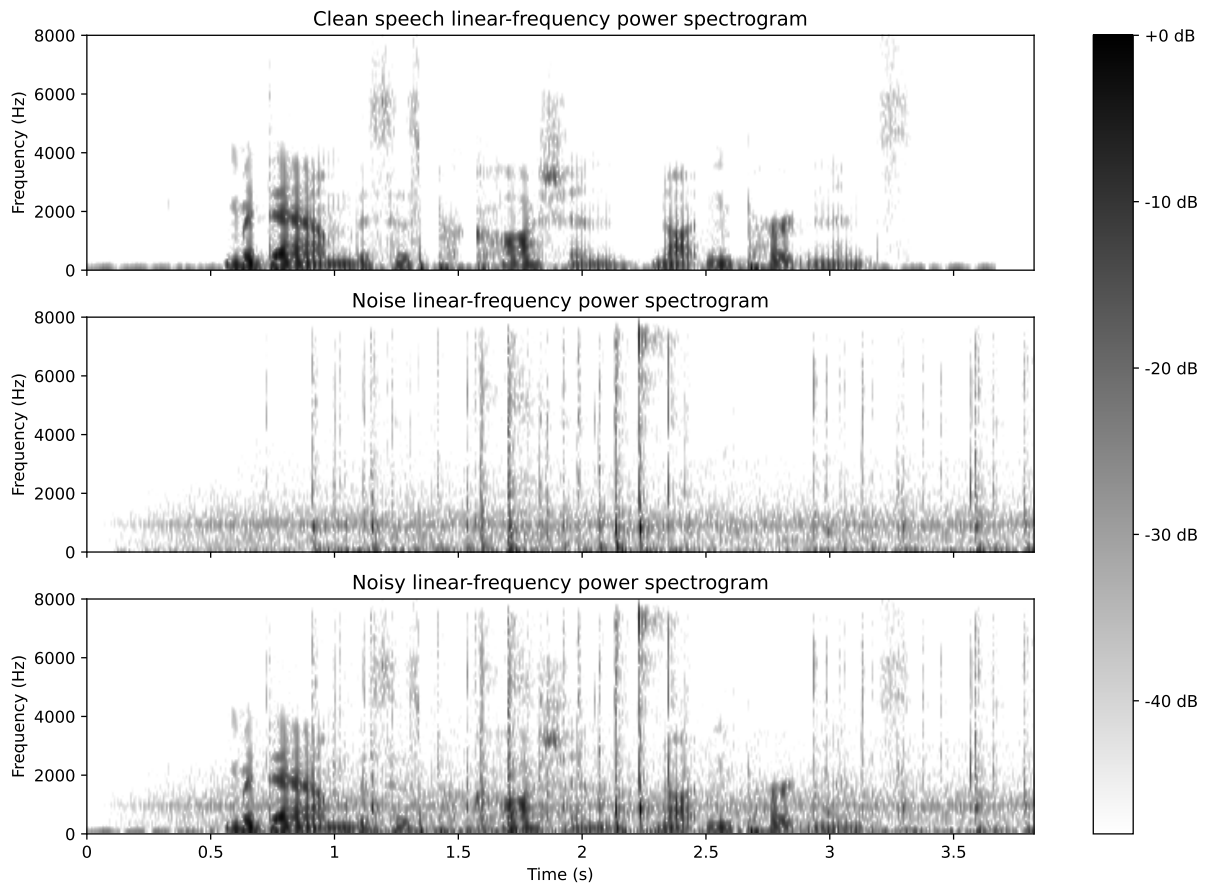


Figure 5: Speech waveform (1034-121119-0042.wav) from L3DAS22 dataset corrupted by an additive keyboard stroke noise from FSD50K development set (155910.wav)

When we add the noise with clean speech, we produce the image at the bottom, and it is possible to observe the keystrokes and the speech with little effort.

Another detrimental factor for speech is a reverberant room. Reverberation is a physical phenomenon related to multiple reflections of sound waves in the room. We have a slight delay (less than 50 ms) when capturing the sound and its reflection in this case. Therefore, the reverberation is perceived as a continuous and elongated sound for the listener. In a scenario where we create an artificial dataset of reverberant speech and have access to clean speech, we need to capture the room properties to understand its features and how reflections will take place. One way to do this is by recording an impulsive signal (e.g., a gunshot) in the desired room to obtain the room impulse response (RIR)  $h[n]$ , which later we convolve with a clean speech signal. If the clean speech is affected both by noises and reverberation, the noisy signal is given by equation 3.2.

$$y[n] = x[n] * h[n] + s[n] \quad (3.2)$$

In figure 6, we show the spectrogram of a clean speech signal and the reverberation effect by using the RIR measurement of a church-like room called Aula Carolina. From the spectrogram, it is possible to observe the effects of the wave reflections. The bottom image is blurrier than the first one, especially in the low frequencies. More broadly, this reflection effect scatters the sound in the time-axis.

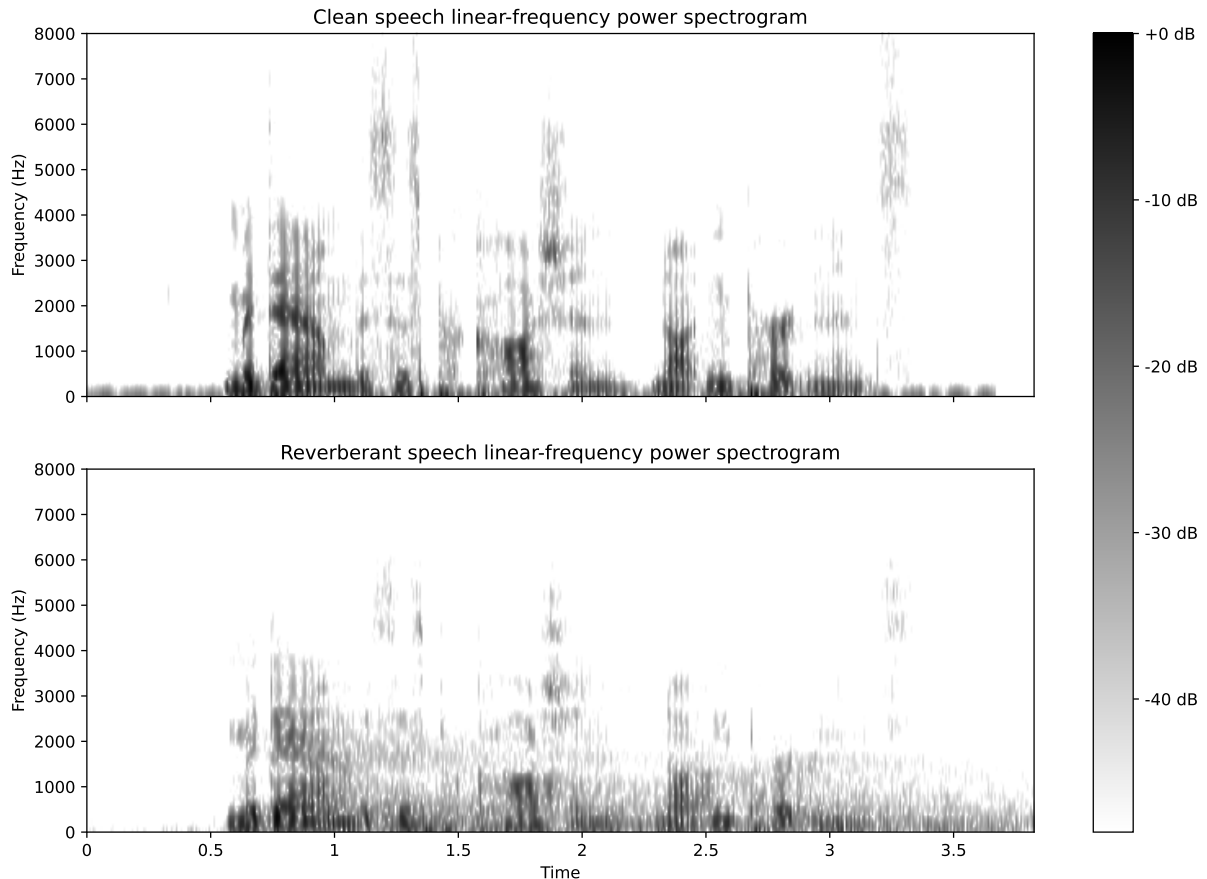


Figure 6: Speech waveform (1034-121119-0042.wav) from L3DAS22 dataset convolved with a RIR from the Aachen AIR database 1.4 (Aula Carolina Room).

Our objective is to estimate a signal  $\hat{x}$  that is as close as possible to  $x$ , or  $\hat{x} \approx x$ . Most approaches consider this problem as the minimization of a distance on a metric space  $\ell_n^p = (\mathbb{R}^n, d_p)$ , where  $d_p$  is a metric defined by the equation 3.3.

$$d_p(x, \hat{x}) = \sum_k^n |x_k - \hat{x}_k|^p \quad (3.3)$$

It is essential to notice that some algorithms based on the minimization of a  $d_p$  metric may introduce some speech distortion while reducing background noises, which may degrade speech intelligibility. For this reason, recently, speech enhancement methods have been tackling the problem of directly optimizing towards intelligibility or quality metrics.

Different algorithms have been proposed for SE. In general, these algorithms can be divided into (LOIZOU, 2013): spectral subtractive algorithms, statistical-model-based algorithms (e.g., Minimum Mean Square Error (MMSE) algorithms, or Wiener filter), subspace algorithms, and binary mask algorithms.

In the recent years, model-based approaches for SE have been widely adopted by the speech processing community to develop powerful tools, especially methods based on neural networks (VALENTINI-BOTINHAO et al., 2016; PARK; LEE, 2017; NIKZAD et al., 2020; SANTOS; FALK, 2018). With this class of algorithm, we can design a non-linear estimator to reconstruct the clean speech given the waveform from the noisy-speech as input or its time-frequency representation.

Deep Neural Network (DNN) models have been proven to be a useful strategy for SE (LU et al., 2013; XU et al., 2015). Macartney and Weyde (MACARTNEY; WEYDE, 2018) studied the use of the Wave-U-Net architecture, an end-to-end learning method used originally for audio source separation. The authors achieved improvements in several metrics, such as perceptual evaluation of speech quality (PESQ), evaluation of the signal distortion (CSIG), background noise intrusiveness (CBAK), overall signal quality (COVL), and segmental signal-to-noise ratio (SSNR). Nikzad et al. (NIKZAD et al., 2020) proposed the residual-dense lattice network (RDL-Net), based on convolutional neural networks (CNN) with residual links (ResNets) and causal dilated convolutional units. The authors demonstrated good results in terms of mean opinion score (MOS) predictors (CSIG, CBAK, and COVL), and also in PESQ and STOI.

## 3.2 Review on Speech Enhancement Methods

In this section, we describe some methodologies found in the literature to tackle the problem of noise suppression and dereverberation. Approaches around noise removal on monaural audio recordings are the most common problem and are the basis for 3D SE. Methodologies using Convolutional Neural Networks (CNN) are among the most prominent techniques to solve this problem. Some proposals were made in recent years on time-domain SE techniques, where the model receives a one-dimensional waveform as input. Initially, the Wave-U-Net model was developed for music source separation, but quickly it was adapted for speech enhancement (MACARTNEY; WEYDE, 2018; GUIMARÃES; NAGANO; SILVA, 2020). The Wave-U-Net is a fully 1D convolutional with an architecture that we can understand as an encoder-decoder model with skip-connection between the two components. In the contracting path, we have a layer that

consists of a one-dimensional strided convolutional block and an activation function with a LeakyReLU function. After each block, we increase the receptive field of our network by a factor of 2 and repeat this layer pattern multiple times until we reach a bottleneck layer.

On the other hand, the decoder path consists of an upsampling operation of the learned feature map, restoring the initial input shape at the end of the process. Some authors proposed the usage of linear-interpolation as an upsampling method instead of Transposed convolutions, which reduced the computational cost of the model without a significant performance degradation (GUIMARÃES; NAGANO; SILVA, 2020). Another mechanism, the pre-training of the network as an autoencoder, was also a helpful weight initialization technique. One of the advantages of this model compared to models that work on the time-frequency domain is the simplicity and the low computational cost, making it even more suitable for real-time scenarios.

More recently, approaches around 3D SE are gaining interest with the signal processing community (GUIZZO; GRAMACCIONI, et al., 2021; GUIZZO; MARINONI, et al., 2022). This scenario is more complex because both noises and reverberation must be handled. A successful method based on STFT representation is the beamforming U-Net for AmbiX audio recordings (REN et al., 2021). The network’s input is the STFT representation without discarding the phase information. To use this information, the authors concatenate the magnitude and phase along the frequency axis of the STFT and perform a set of two-dimensional convolutional blocks, similar to the U-Net network. In the final layer of the network, the estimated binary mask is multiplied by the original B-format STFT from the noisy signal. We sum the results over the channel axis to get a monaural audio representation. To obtain the signal back in the time domain, the ISTFT is applied to the output.

To handle the reverberation component of the SE process, techniques based on Recurrent Neural Networks (RNN) are gaining attention. For the problem of 3D SE, where the input is a multichannel audio signal, and the enhanced output should be monaural, some authors proposed a two-stage pipeline based on Convolutional Recurrent Network (CRN) in a U-Net scheme (LI et al., 2022).

The first component is a network responsible for noise removal and dereverberation that receives as input a multichannel input that is transformed to the STFT representation. First, the real and imaginary components are concatenated along the channel dimension. The expansive path of this network contains two separated decoders responsi-

ble for estimating a binary mask for the real and imaginary parts, respectively. The next step is to multiply the mask by the original input and obtain a multichannel output. This component, that receives as input and returns as output a multichannel signal, is called the MIMO Network (LI et al., 2022).

The second stage of the proposed model is the "Multiple-Input, Single Output", or MISO Network (LI et al., 2022). In this component, the goal is to go from a multichannel audio representation to a monaural one. For this, the authors use a two-layer LSTM network. Again, the input is the STFT representation with the real and imaginary parts concatenated along the channel axis. The output of this part is a feature map with 4-channels that will be fed into a second CRN network that will output a monaural representation (i.e., a feature map with two channels representing the real and imaginary components). Finally, the ISTFT is used to obtain the signal back in the time domain.

Other approaches combining signal processing techniques with DNN models showed to be successful for 3D SE (LU et al., 2022; ZHANG et al., 2022). For example, the linear acoustic echo canceller (LAEC) and time delay compensator (TDC) are used as a pre-processing step before feeding the STFT representation to the DNN model (ZHANG et al., 2022). The usage of Multichannel Wiener Filtering (MCWF) is proposed as an intermediate step in the work of Lu et al. (LU et al., 2022). First, a neural network is used to estimate the complex coefficients directly from the STFT representation. This feature map is the input of the MCWF, and both the feature map and the output from the MCWF are fed into a second DNN model to estimate the final output directly.

In the next section we discuss the most popular loss functions for neural networks when tackling the SE problem. Most of the approaches rely on a regression loss that consists of  $d_p$  metric, but we will discuss some viable alternatives to optimize directly by means of perceptual metrics.

### 3.3 On Loss Functions for Speech Enhancement

When designing a neural network for the SE task, our objective is to find a set of parameters  $\theta$  that composes an estimator  $\hat{x} = f(y; \theta)$ , that minimizes a distance metric between  $x$  and  $\hat{x}$ , or maximizes a perceptual metric, such as the Short-Time Objective Intelligibility (STOI) measure. Most neural networks are optimized with the Gradient Descent method, so our loss function must be differentiable. Similar to the work of (GUSÓ et al., 2022), who studied the impacts of loss functions on the problem of music



source separation, we further investigate losses for speech enhancement.

One of the most used functions for this objective is a reconstruction loss based on the  $d_p(\cdot, \cdot)$  directly on the time domain. Considering a batch of  $N$  samples, we present the equations for  $\mathcal{L}_1$  (i.e., Mean Absolute Error) and  $\mathcal{L}_2$  (i.e., Mean Squared Error) in equations 3.4 and 3.5, respectively.

$$\mathcal{L}_{1\text{-time}} = \frac{1}{N} \sum_k |\hat{x}_k - x_k| \quad (3.4)$$

$$\mathcal{L}_{2\text{-time}} = \frac{1}{N} \sum_k |\hat{x}_k - x_k|^2 \quad (3.5)$$

Another common technique is to use a regression loss on a time-frequency representation. Consider the magnitude of the Short-time Fourier Transform (STFT) of a signal  $x$  given by equation 3.6, where  $n$  represents a discrete time-index,  $\omega$  a continuous frequency variable,  $x(m)$  is the input signal and  $w(n - m)$  is the analysis window shifted by  $n$  samples. In this work we use a Hamming window to compute the STFT.

$$S(n, \omega) = |X(n, \omega)| = \left| \sum_{m=-\infty}^{\infty} x(m)w(n - m)e^{-j\omega m} \right| \quad (3.6)$$

Similar to the previous approach, we apply a point-wise regression loss on top of this representation, creating our time-frequency loss functions as represented by equations 3.7 and 3.8.

$$\mathcal{L}_{1\text{-freq}} = \frac{1}{N} \sum_{n, \omega} |\hat{S}(n, \omega) - S(n, \omega)| \quad (3.7)$$

$$\mathcal{L}_{2\text{-freq}} = \frac{1}{N} \sum_{n, \omega} |\hat{S}(n, \omega) - S(n, \omega)|^2 \quad (3.8)$$

For the last, the usage of multi-resolution STFT (MRSTFT) is another popular choice for a spectrogram-based loss function (YAMAMOTO; SONG; KIM, 2020). The key idea behind this loss function is to use an STFT loss over a set  $\mathcal{O}$  of different parameters, namely the number of FFTs, window length and hop size, to generate the STFT representation.

The STFT loss is composed of two parts: The first one is a spectral convergence

method described on equation 3.9.

$$\mathcal{L}_{\text{sc}} = \frac{\|S - \hat{S}\|_F}{\|S\|_F} \quad (3.9)$$

The second is a magnitude loss, very similar to the  $\mathcal{L}_{1\text{-freq}}$  but we apply the  $\log_{10}$  operator in both inputs, as shown in equation 3.10.

$$\mathcal{L}_{\text{mag}} = \sum_{n,\omega} |\log_{10} \hat{S}(n, \omega) - \log_{10} S(n, \omega)| \quad (3.10)$$

The final MRSTFT is the sample mean from the convex combination of both losses over all possible configurations of  $\mathcal{O}$ , as shown in 3.11. In this work we set both losses to have the same importance into the final metric ( $\lambda_1 = \lambda_2 = 0.5$ ). The hyperparameters choose for this loss can be found at B.

$$\mathcal{L}_{\text{MRSTFT}} = \frac{1}{N|\mathcal{O}|} \sum_{i=1}^{|\mathcal{O}|} (\lambda_1 \mathcal{L}_{\text{sc}} + \lambda_2 \mathcal{L}_{\text{mag}}) \quad (3.11)$$

Both of the time and time-frequency domain approaches above consider a regression problem. The first perceptual metric that we consider is the usage of the  $\mathcal{L}_{\text{stoi}}$ , which is obtained directly from the STOI metric (TAAL et al., 2010), with some adaptations to improve computational performance, as proposed in (MANUEL, 2021). In this soft version, we detect silent frames and store this information as a mask tensor. The mask is applied before the mean operation in (3.13). An open source implementation can be found in (MANUEL, 2021).

The STOI metric is calculated using (3.12) and (3.13). The variable  $X_j$  indicates the  $j^{\text{th}}$  one-third octave band from the discrete Fourier transform (DFT) of the noisy signal, and  $Y_j$  can be defined in a similar form for the clean audio. The  $Y'_j$  represents the  $Y_j$  vector normalized and clipped. The  $\mu$  represents the means of the representations.

$$d_j(m) = \frac{(X_j - \mu_{X_j})^T (Y'_j - \mu_{Y'_j})}{\|X_j - \mu_{X_j}\| \|Y'_j - \mu_{Y'_j}\|} \quad (3.12)$$

The STOI metric is an average over all ( $M$  total time frames and  $J$  total one-third octave bands) estimated linear correlation coefficients, as defined in equation 3.13.

Therefore, we construct the STOI metric as an average over the estimated linear

correlation coefficient, as defined in equation (3.13).

$$d_{stoi} = \frac{1}{JM} \sum_{j,m} d_j(m) \quad (3.13)$$

As a loss function, the STOI-LF can be described as the negative of the metric as indicated in (3.14).

$$\mathcal{L}_{stoi}(y, \hat{y}) = -d_{stoi} \quad (3.14)$$

It is important to notice that most of the operations required are differentiable and it is possible to compute the gradients, except the  $Y'$ , which relies on a min operation. However, this application is possible due to subgradient operation that is no much more expensive than a ReLU activation derivative. Hence it is possible to use this as loss function without general performance degradation.

Deep Feature Loss is another technique to compare high-level features in a neural network’s loss function instead of directly maximizing the likelihood in the reconstruction process. In this methodology, we extract an embedding from the  $j^{\text{th}}$  layer of the neural network (usually the last layer or a combination of all layers) and compute a distance metric between the embeddings from the clean signal against the enhanced signal (JOHNSON; ALAHI; FEI-FEI, 2016). Usually, the chosen distance metric is the  $\mathcal{L}_2$  loss and has been used in simultaneousness with a reconstruction loss, such as the  $\mathcal{L}_{2\text{-freq}}$  on the magnitude STFT (SAHAI; WEBER; MCWILLIAMS, 2019). The work of (GERMAIN; CHEN; KOLTUN, 2018) shows us that this type of loss function has a superior objective quality metrics for the speech enhancement task.

In this work, when referring to Deep Feature Loss (DFL), we will use an  $\mathcal{L}_{2\text{-freq}}$  reconstruction loss and the MSE between the embeddings extracted from the last layer of a model  $\phi_m$ , as described by equation 3.15. Notice that  $\phi_m$  is differentiable, and therefore we can use it in the backpropagation mechanism. The  $\lambda$  is a weight parameter to control how strong is the feature loss regularization.

$$\mathcal{L}_{DFL} = \mathcal{L}_{2\text{-freq}} + \frac{\lambda}{N} \sum_k (\phi_m^j(\hat{x}) - \phi_m^j(x))^2 \quad (3.15)$$

Based on the above losses, we propose a specific loss to optimize intelligibility based on the earlier work of the Phone-Fortified Perceptual Loss (PFPL) (HSIEH et al., 2021),

which we discuss in the next section. Our goal is to maximize the STOI metric and reduce the Word-Error Rate (WER) from *wav2vec2.0* based speech recognition systems. Also, this loss function links our previous chapter on Speech Representation Learning with the task of Speech Enhancement.

### 3.4 Compound Perceptual-Loss based on Speech Representations

We propose using representation learning methods on speech enhancement tasks. In our initial experiments, we evaluate the usage of the learned latent spaces as a guide for the loss function in the enhancement tasks. Since similar samples should be close in the feature space of an SRL model, we presume the distance from the latent vector of the enhancement system’s output should be close to the vector generated from the clean speech.

This is the basis for the Phone-Fortified Perceptual Loss (PFPL) (HSIEH et al., 2021). As demonstrated, the PFPL is computed based on latent representations of the *wav2vec* model (SCHNEIDER et al., 2019). The PFPL uses the Wasserstein distance during training and is defined by the Kantorovich-Rubinstein dual form of Wasserstein distance (3.16). The PFPL is given by the following equation:

$$\mathcal{L}_{\text{PFPL}}(y, \hat{y}) := \|y - \hat{y}\|_1 + \sup_{f \in \mathcal{F}} [ \mathbb{E}_u [f(c)] - \mathbb{E}_v [f(\hat{c})] ], \quad (3.16)$$

where  $f$  is a Lipschitz continuous function;  $c = \Phi_{\text{wav2vec}}(y)$  and  $\hat{c} = \Phi_{\text{wav2vec}}(\hat{y})$  are the outputs of the encoder *wav2vec* model,  $\Phi_{\text{wav2vec}}$ , of the clean speech and the enhanced speech, respectively;  $u$  and  $v$  are the densities of the  $c$  and  $\hat{c}$  features in the latent space. The PFPL also includes a mean absolute error (MAE) loss applied to increase the CBAK metric performance (HSIEH et al., 2021). In their work, other metrics are also used to measure the distance between the vectors in the representation space (e.g., MAE).

In a different way, we propose an optimized objective function in equation (3.17). The key idea was creating an approximation to directly optimize the intelligibility metrics of STOI and the WER from the Speech Recognition system on top of the *wav2vec2.0* model. For this, we assume that similar latent representations for the clean and enhanced signals should lead to the same transcriptions for both waveforms and therefore approximate the WER, which is not differentiable.

$$\mathcal{L}_{\text{CPL}}(y, \hat{y}) = \mathcal{L}_{\text{stoi}} + \alpha \left[ \sup_{f \in \mathcal{F}} [ \mathbb{E}_{\mu} [f(c)] - \mathbb{E}_{\nu} [f(\hat{c})] ] \right], \quad (3.17)$$

where  $c$  and  $\hat{c}$  are extracted using the *wav2vec1.0* or *wav2vec2.0* model, and  $\alpha \in \mathbb{R}$  is a weight factor.

## 4 METHODOLOGY

This chapter describes the methods and materials necessary to reproduce our experiments. First, we briefly introduce the datasets used in this work and the infrastructure necessary to run the experiments. Finally, we discuss our proposed methods to perform 3D speech enhancement.

### 4.1 Datasets

The following sections detail the datasets used in this work for speech representation learning and speech enhancement in the monaural and tridimensional scenarios.

#### 4.1.1 Librispeech Corpus

The Librispeech dataset (PANAYOTOV et al., 2015) is a set of utterances with approximately 1,000 hours of speech extracted from audiobooks and carefully designed. The data consists of audios with a sampling rate of 16 kHz, and 16 bit resolution. The data was split into train, development, test sets, and clean and noisy audio subsets. The train set consists of 960 hours of utterances divided into subsets of 460 hours of clean audios and 500 hours with audio with less quality (channel and environmental noises). The other 40 hours are divided between the development and test sets, with clean and noisy subsets.

Originally the dataset was proposed for automatic speech recognition (ASR) tasks, and therefore has transcripts aligned with the audio to train supervised models. In this corpus, we have approximately 977K unique words. Unsupervised models currently use this same dataset but discard the associated transcripts.

In this work, we use the Librispeech dataset to train our custom SSL models. Training models capable of capturing the underlying distribution of language itself is challenging, and we need an extensive utterance corpus to discover patterns and fit large models. Also, this dataset is used for training speech recognizers as can be found in the literature on

robust models such as *wav2vec* (SCHNEIDER et al., 2019), *wav2vec2.0* (BAEVSKI et al., 2020), and the HuBERT (HSU et al., 2021).

### 4.1.2 3D Audio Sources

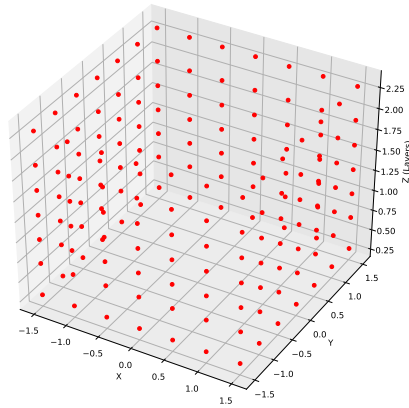
For the task of speech enhancement in a 3D environment, we use the datasets provided by the Learning 3D Audio Sources (L3DAS) project from the Sapienza University of Rome. The data was made publicly available through a data competition (GUIZZO; GRAMACCIONI, et al., 2021) in the IEEE International Workshop on Machine Learning for Signal Processing (MLSP). The L3DAS21 dataset contained multiple-source and multiple-perspective B-format Ambisonics audio recordings, with 16 bit-AmbiX wav files having a sampling rate of 16 kHz and was designed based on clean speech sounds extracted from the Librispeech. The noises are from the Freesound Dataset 50K (FSD50K) (FONSECA et al., 2020) corpus, a public dataset with 100 hours of manually labeled audio, consisting of 50 thousand clips distributed across 200 classes.

In the L3DAS21 dataset, the noises are drawn to represent fourteen transient noise classes and four continuous noises: computer keyboard, drawer open/close, cupboard open/close, finger-snapping, keys jangling, knock, laughter, scissors, telephone, writing, chink and clink, printer, female speech, male speech, alarm, crackle, mechanical fan, and microwave oven.

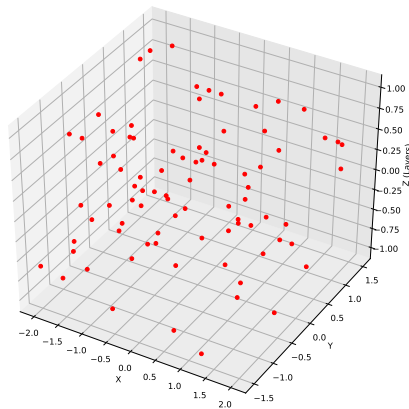
Those sounds are convolved with 252 room impulse responses (RIR) collected in different positions of an office-like environment, as shown in Figure 7. These synthetic tridimensional sounds aim to create plausible 3D scenarios to produce possible real-life situations in which sound and background noises coexist in the same 3D reverberant environment.

More recently, the L3DAS project released a new enhanced version of the data challenge, called L3DAS22 (GUIZZO; MARINONI, et al., 2022). In this challenge, some aspects of the dataset were improved for the speech enhancement task. The most prominent one is on data collection: There are more data because the authors allowed signals up to 12 seconds long from the clean subset of the Librispeech, reaching a total duration of more than 80 hours and up to 40,000 utterances.

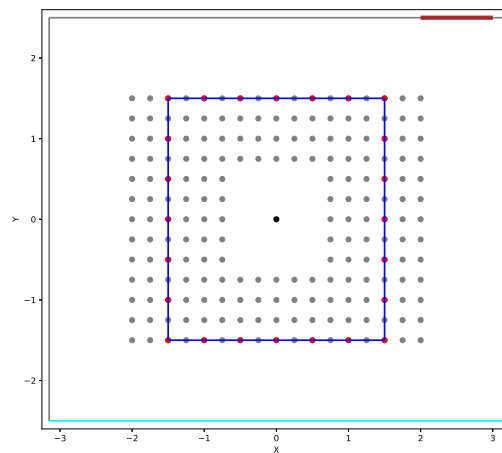
It is guaranteed that each file always presents a speech signal. Up to 3 background noises from FSD50K can corrupt the signal (same noise categories from the previous challenge), but with a 25% probability of being a continuous noise. Also, the signal-to-



(a) 3D Grid



(b) Random positions



(c) 2D projection

Figure 7: L3DAS RIR positions: (a) 168 fixed points from a 3D Grid (b) 84 randomly selected positions (c) 2D projection view from above. Figures from (GUIZZO; MARI-  
NONI, et al., 2022)



noise ratio is always greater than zero dB, which means that the voice is the prominent signal. Finally, the presented subset is balanced with a similar proportion of male and female speakers.

## 4.2 Infrastructure and Software

In this work, we use a desktop computer with 32 GB of RAM, an Intel i5 9th generation processor with six cores, and a single NVIDIA RTX 2060 Super GPU card with 8 GB to train our models. For the large models, we use pre-trained weights (e.g., *wav2vec2.0*) available at Huggingface’s Transformers (WOLF et al., 2019) open source library.

Our models are implemented using the Pytorch framework with the open-source SpeechBrain toolkit (RAVANELLI et al., 2021), an easy-to-use tool for speech-related tasks in research and development (R&D) environments, allowing us to spend more time in research than boilerplate coding. SpeechBrain is built on top of Pytorch and allows us to compare other model implementations. The source code and all the necessary parameters for reproducibility are available at:

1. <https://github.com/Hguimaraes/SE3D>
2. <https://github.com/Hguimaraes/3Denoiser>

## 4.3 Proposed methods for 3D Speech Enhancement

The following subsections describe the experiments associated with the Learning 3D Audio Source (L3DAS) Challenges, using the 2021 and 2022 datasets. We participated in each data challenge associated with IEEE signal processing conferences to train and test our methodology as described below. In both experiments, we propose perceptual losses to assess our model efficiency, but each methodology works with different input representations, namely, the time domain and a time-frequency domain. We also performed a third set of experiments on the L3DAS22 dataset to further investigate the impacts of different loss functions.

### 4.3.1 IEEE MLSP 2021 Data Challenge

In this section we describe our initial approach for the 3D Speech Enhancement scenario using the L3DAS21 dataset. Using the subset of 100 hours, we propose the training

of a Fully Convolutional Network (FCN) for the SE task. Figure 8 shows the sequential model with 7 FCN blocks, as indicated in Figure 8 (a), extended with 1 convolutional layer at the end of the model. Each FCN block (FCN-B) consists of a 1-D convolution layer with 55 filters, followed by a 1-D instance normalization, and a LeakyReLU activation function with a slope of 0.1. Reflection paddings were used in order to preserve the audio size. The choice of using reflection padding instead of a usual zero-padding is to avoid the creation of border artifacts while using the convolution operator, as demonstrated in (GUIMARÃES; NAGANO; SILVA, 2020).

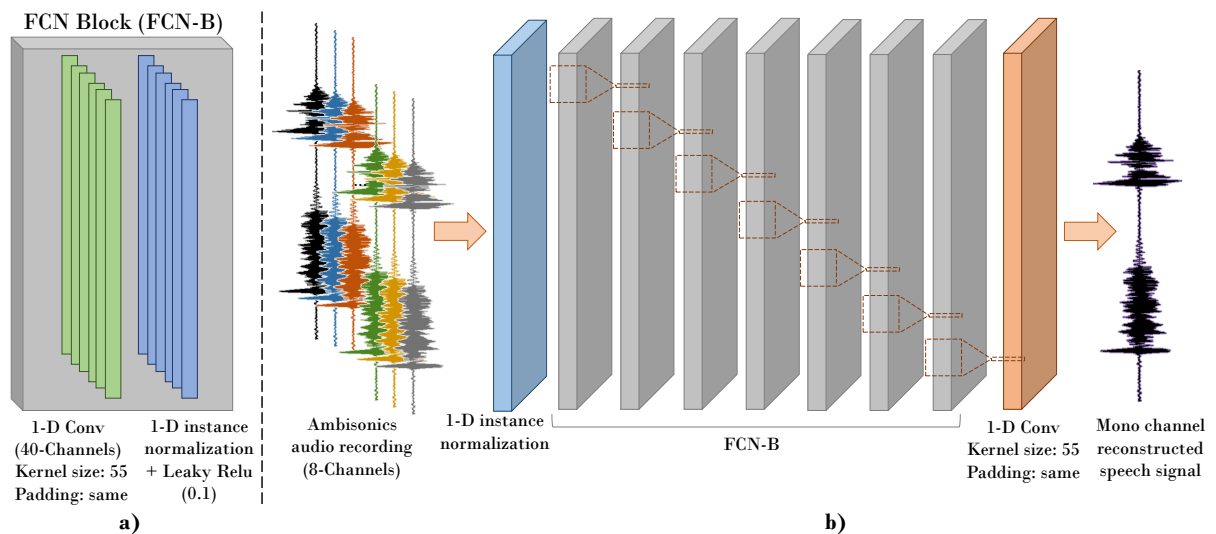


Figure 8: The architecture of the FCN for 3D speech enhancement in a reverberant environment: (a) FCN diagram block, and (b) the overall architecture.

The model input has eight channels, each containing the raw waveforms of the 8-Ambisonics channels. The model output reconstructs only one channel, representing a clean mono speech. We randomly select 2-s segments (32,000 samples) in each epoch in the training phase. On the other hand, the audio is kept with the original size for the validation and test phase. Being invariant to the waveform input size is one of the advantages of the FCN strategy. The model was trained for 50 epochs, resulting in approximately 10 hours of training.

The evaluation metric,  $M$ , for this task is a combination of STOI, which estimates the intelligibility of the output speech signal, and WER, computed to assess the effects of the enhancement for speech recognition purposes. The final metric for this task is given by (4.1), which lies in the 0-1 range, with higher values indicating better performance. This is the metric to evaluate the quality of the enhancement proposed by the authors of the data challenge.

$$M = [\text{STOI} + (1 - \text{WER})]/2 \quad (4.1)$$

The design of the proposed loss functions is an approximation to directly optimize the given metric. We rely on this scheme since the metric is not differentiable and could not be used as a loss function directly.

In the first set of experiments, we would like to evaluate the importance of the loss function with respect to our task. We investigate the usage of usual reconstruction losses such as Mean Absolute Error (MAE) and Mean Squared Error (MSE), against perceptual losses: The STOI-LF, PFPL, and ours.

Next, we study the behavior of different models against ours. We chose four approaches for the comparison: Noisy, Wiener, FaSNet, and SEWUNet. The *Noisy* approach consists of a simple average operation in all the channels of the input audio and the direct comparison against the target. This approach can be used as a reference that all models should improve upon.

The Filter and Sum Network (FaSNet) (LUO et al., 2019) is the proposed baseline (GUIZZO; GRAMACCIONI, et al., 2021) for the challenge. The FaSNet is a time-domain neural beamforming with high-performance for low-latency scenarios. The model works on a two-stage approach, where first, it learns adaptive filters for a reference channel and computes the filters for the remaining channels. Then, the filtered output is summed across all channels to generate the final output. Compared to traditional beamformer techniques, the authors improve the SNR for the reverberant speech enhancement task.

The SEWUNet is also used as an alternative time-domain model. We modified the first layer of the architecture, compared with the original architecture (GUIMARÃES; NAGANO; SILVA, 2020), to accept 8-channel input audio for the enhancement process. We also did not use the weight initialization as proposed in the paper and our loss function was used instead of the early  $L_1$  loss. This modification improved the results observed on the development set metrics.

### 4.3.2 IEEE ICASSP 2022 Data Challenge

In this section, we describe our approach for the L3DAS22 data challenge. This work shares some ideas from our previous work (compounded perceptual losses) (GUIMARÃES; BECCARO; RAMÍREZ, 2021) and inspirations from the work that achieved first place at the L3DAS21 challenge (REN et al., 2021) (e.g., using time-frequency representation

as input of the network without discarding phase-information). To train our model, we use the subset of 100 hours due to computational resources, and we propose the training of a Fully Complex Convolutional Network (FC2N) for the SE task. Our approach is based on a single microphone (mic A), similar to the work of Ren (REN et al., 2021). The system’s input is a 16 bit AmbiX 16 kHz waveform transformed to a time-frequency representation using a one-sided short-time Fourier transform (STFT) for each channel. We arrange the tensors to be in the format  $B \times N \times T \times 8$ , where  $B$  is the batch size,  $N$  is the number of frequencies,  $T$  is the total number of frames, and 8 is the number of channels. In this representation, the first four channels represent the real parts of our STFT, and the others are the imaginary parts.

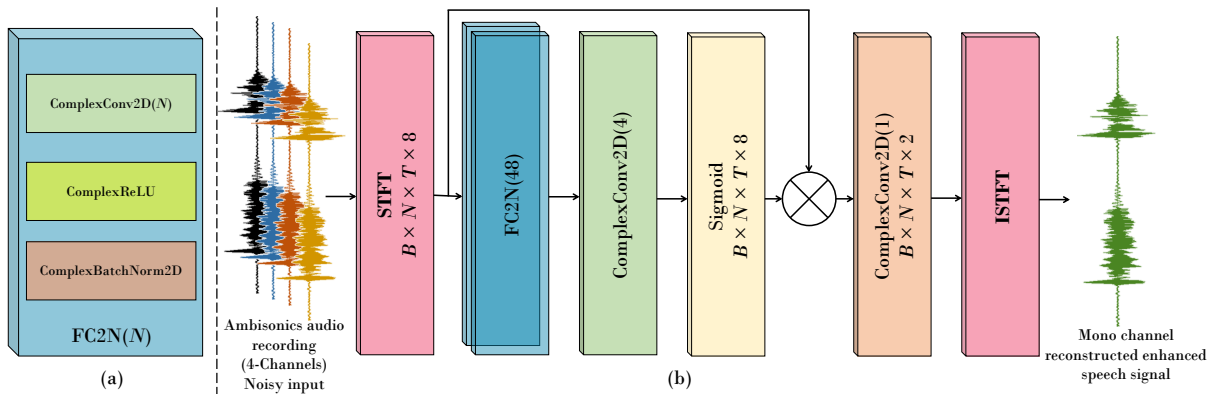


Figure 9: The architecture of the FC2N for 3D speech enhancement in the L3DAS22 challenge: (a) Complex convolutional block, and (b) The overall architecture.

Figure 9 shows the proposed architecture that estimates a mask to multiply with the input representation. The network has five blocks, consisting of a Complex 2D Convolution, a Complex Batch Normalization operator, and a Complex ReLU activation function, except for the last block, which contains a Sigmoid activation function. The output also has the shape  $B \times N \times T \times 8$  and is multiplied (point-wise) with the original STFT representation. Then, we apply a single Complex 2D Convolution to transform it to a monaural representation  $B \times N \times T \times 2$ , and whose output is used to reconstruct the waveform using the inverse short-time Fourier transform (ISTFT) function.

The components from FC2N are implemented as described in (TRABELSI et al., 2017). In a similar fashion, we define a Complex filter matrix  $\mathbf{W} = \mathbf{A} + i\mathbf{B}$  and a complex vector  $\mathbf{h} = \mathbf{x} + i\mathbf{y}$ . In the following items, we define the Complex convolution and activations used by our network. More details on the batch normalization can be found at (TRABELSI et al., 2017).

**Definition 4.3.1** (Complex Convolution). We define the complex convolution as the dot

product between the complex filter matrix  $\mathbf{W}$  and the complex vector  $\mathbf{h}$  as:

$$\begin{aligned} \mathbf{z} &= \mathbf{W} * \mathbf{h} = (\mathbf{A} + i\mathbf{B}) * (\mathbf{x} + i\mathbf{y}) \\ &= (\mathbf{A} * \mathbf{x} + i\mathbf{A} * \mathbf{y}) + (i\mathbf{B} * \mathbf{x} + i^2\mathbf{B} * \mathbf{y}) \\ &= (\mathbf{A} * \mathbf{x} - \mathbf{B} * \mathbf{y}) + i(\mathbf{B} * \mathbf{x} + \mathbf{A} * \mathbf{y}) \end{aligned}$$

It also can be describe as a matrix multiplication operation as shown in equation 4.2.

$$\mathbf{z} = \begin{bmatrix} \mathcal{R}(\mathbf{W} * \mathbf{h}) \\ \mathcal{I}(\mathbf{W} * \mathbf{h}) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} * \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \quad (4.2)$$

**Definition 4.3.2** (Complex ReLU). The CReLU is a holomorphic function, therefore can be used in the backpropagation mechanism.

$$\text{CReLU}(\mathbf{z}) = \text{ReLU}(\mathcal{R}(\mathbf{z})) + i\text{ReLU}(\mathcal{I}(\mathbf{z})) \quad (4.3)$$

**Definition 4.3.3** (Complex Batch Normalization). It is an extension of the Batch Normalization operation to accelerate the convergence of the learning method in the complex domain and to avoid overfitting. The formulation is given by:

$$\text{CBN}(\mathbf{z}) = \mathbf{\Gamma}(\mathbf{V})^{-\frac{1}{2}}(\mathbf{z} - \mathbb{E}[\mathbf{z}]) + \boldsymbol{\beta}, \quad (4.4)$$

where  $\boldsymbol{\beta}$  is the shift parameter (Complex number),  $\mathbf{\Gamma}$  is a  $2 \times 2$  positive semi-definite matrix acting as a scale parameter, and  $\mathbf{V}$  a covariance matrix defined as follow:

$$\mathbf{V} = \begin{bmatrix} \text{Cov}(\Re\{\mathbf{z}\}, \Re\{\mathbf{z}\}) & \text{Cov}(\Re\{\mathbf{z}\}, \Im\{\mathbf{z}\}) \\ \text{Cov}(\Im\{\mathbf{z}\}, \Re\{\mathbf{z}\}) & \text{Cov}(\Im\{\mathbf{z}\}, \Im\{\mathbf{z}\}) \end{bmatrix} \quad (4.5)$$

Overall, in this set of experiments, our objective was to maximize the same metric provided in the first competition (Eq. 4.1). Currently, compared to the previous experiments, we aim to experiment with different network architectures, use different representation learning models for the perceptual loss (e.g., *wav2vec2.0*), and apply different distance metrics in the feature space. We also study the impact of different loss functions on this problem.

## 5 RESULTS AND DISCUSSION

In this chapter we discuss the results associated with the experiments proposed in the previous sections to tackle 3D Speech Enhancement.

### 5.1 IEEE MLSP 2021 Data Challenge

In our first set of experiments, we identified that the loss function has a fundamental role in making the deep models improve the metric result. In fact, in our analysis, the loss function was a more critical component than the network architecture itself when using a time-domain model. In Table 1, we compare different losses and their impact on the metrics. In this ablation study, we fixed the network architecture, as presented in Figure 8, and the Adam optimizer.

Table 1: Ablation study on the effect of the loss functions to the metrics.

Loss	STOI	WER	$M$
$L_1$	0.49	0.98	0.26
$L_2$	0.52	0.93	0.29
STOI-LF	0.82	0.36	0.73
PFPL	0.68	0.56	0.56
$L_1 + \text{PFPL}$ (HSIEH et al., 2021)	0.60	0.68	0.46
<b>STOI-LF + PFPL</b> ( $\alpha = 5$ )	0.82	0.36	0.73
<b>STOI-LF + PFPL</b> ( $\alpha = 1000$ )	0.83	0.35	0.74

In the first experiments, we investigated the usage of the regression losses  $L_1$  and  $L_2$ . The best results were obtained with  $L_2$  loss, achieving a value of STOI equal to 0.52, WER equal to 0.93, and  $M$  equal to 0.29. These results suggest that  $Lp$ -norm functions were not able to adequately train the model.

Table 1 also presents the comparison of three perceptual loss functions: STOI-LF,  $L_1 + \text{PFPL}$  (HSIEH et al., 2021), and STOI-LF + PFPL with  $\alpha = 5$  and  $\alpha = 1000$ .

For small values of  $\alpha$ , the PFPL + STOI-LF and STOI-LF are almost identical since the STOI-LF is predominant in the total value of the loss function. In both cases, the model achieves a STOI score of 0.82, WER equal to 0.36, and  $M$  equal to 0.73.

The best results were obtained with STOI-LF + PFPL ( $\alpha = 1000$ ), achieving a value of STOI equal to 0.83, WER equal to 0.35, and  $M$  equal to 0.74. The training with PFPL has a faster convergence, which can be adequate to obtain trained models with few epochs.

The model with STOI-LF + PFPL ( $\alpha = 5$ ) was the one submitted for the challenge evaluation. The improvement achieved with the variation of the  $\alpha$  value was studied after the challenge deadline.

The next step was to compare how different models perform on this task. We chose four approaches with the one here proposed: Noisy, Wiener, FaSNet, and SEWUNet. The results on the development set are presented in Table 2.

The *Noisy* approach consists of a simple average operation in all the channels of the input audio and the direct comparison against the target. This approach can be used as a reference.

Table 2: Performance on the development set of the task. Comparison of different approaches: Noisy, Wiener, FaSNet, SEWUNet, STOI-LF + PFPL ( $\alpha = 5$ ), and STOI-LF + PFPL ( $\alpha = 1000$ ).

Approach	STOI	WER	$M$
Noisy	0.57	0.43	0.57
Wiener	0.39	0.43	0.48
FaSNet	0.72	0.46	0.62
SEWUNet	0.79	0.40	0.69
<b>Ours</b> (loss with $\alpha = 5$ )	0.82	0.36	0.73
<b>Ours</b> (loss with $\alpha = 1000$ )	0.83	0.35	0.74

Our model using the FCN architecture and the proposed loss function achieved the best results in our experiments. Compared to the SEWUNet, we proposed a simpler model that also operates directly on the time domain, without re-sampling operations. Moreover, the scores obtained in the test set of the challenge for STOI, WER, and the  $M$  were  $\{0.83, 0.31, 0.76\}$ , respectively, placing us in the second position in the challenge.

Figure 10 illustrates the effectiveness of the speech enhancement model through a 7 s test utterance corrupted by a reverberant noise. The diagram shows the initial noisy

speech log-frequency spectrogram, the enhanced speech spectrogram predicted by the model, and the clean speech spectrogram (original file). The proposed model was able to produce a smoothed version of the speech spectrogram.

One can observe the presence of noise components distributed all over the spectrogram of the noisy speech signal, essentially in low frequencies, lower than 256 Hz. The enhanced speech has had much of the noise removed, not affecting the harmonics.

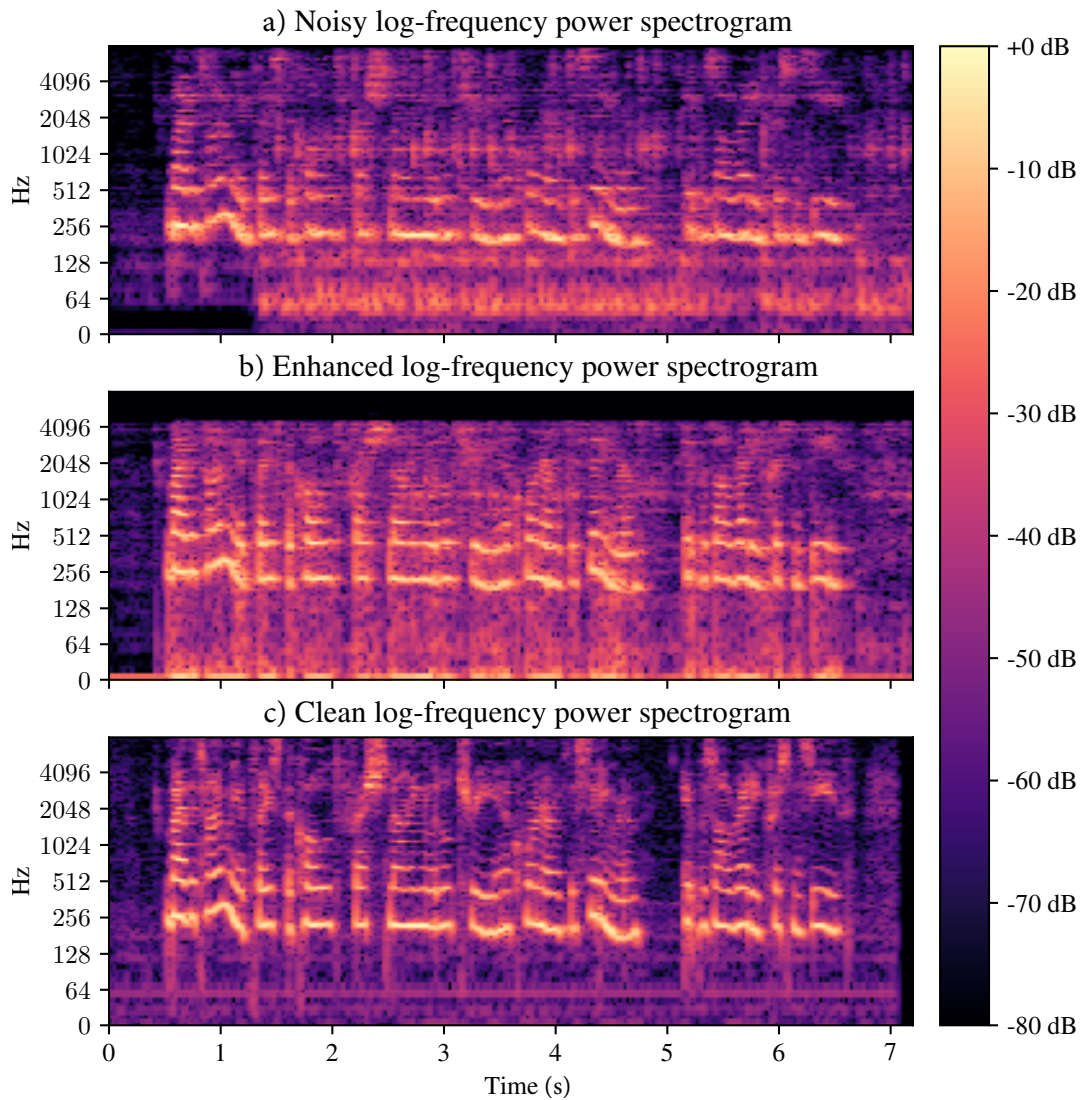


Figure 10: Log-frequency spectrogram of the 1993-147964-0008\_A.wav audio file: (a) noisy speech; (b) enhanced speech and (c) clean speech.



## 5.2 IEEE ICASSP 2022 Data Challenge

These are the partial results on the first set of experiments in the L3DAS22 challenge, presented in table 3. The usage of time-frequency representations achieves a considerable improvement over our previous experiments. We also can observe an improvement on the WER metric that will be investigated in further experiments, but we attribute it to the usage of the *wav2vec2.0* model.

Table 3: L3DAS22: Performance on the development set of the task. Comparison of different models.

Approach	STOI	WER	$M$
FaSNet	0.72	0.46	0.62
TD-FCN	0.83	0.35	0.74
Beamforming U-Net	0.87	0.25	0.81
<b>STFT-FC2N</b>	0.86	<b>0.18</b>	<b>0.84</b>

Another important aspect is the usage of complex networks, where we directly optimize our STFT representation’s real and imaginary components. In this approach, we do not need to use the noisy phase components to invert the STFT representation in the end. Instead, the Beamforming U-Net (REN et al., 2021) concatenates the real and imaginary components in the exact representation, similar to adding new frequencies to the STFT, and uses regular two-dimensional convolutions. Our results indicate that Complex networks are an adequate approach for this type of representation.

On the other hand, we have a tradeoff between metric and time performance compared to our previous time-domain representation. Using the same hardware, we increased the time to complete a single epoch from 12 minutes to 2 hours. This time can be prohibitive in some scenarios, especially in competitions where we must quickly iterate.

In Figure 11(a)-(c), we show the wideband spectrogram of the utterance 1993-147964-0008\_A.wav audio file, spoken by a female speaker and transcribed as “by the time we had placed the cold fresh-smelling little tree in a corner of the sitting room it was already Christmas eve”. In Figure 11(a), we can observe the presence of reverberation accompanied by an additive noise of keystrokes produced by a computer keyboard. The keystrokes (clicks) can be seen in noisy spectrogram by repetitive patterns (spaced approximately by 150 ms) with wide spectral distribution. Besides, a comparison of the noisy, Figure 11(a), and the clean speech spectrogram, Figure 11(c), indicates that a large number of temporal

gaps were filled due to the reverberation. The enhanced spectrogram obtained with the reconstructed speech signal (i.e., the output of the FC2N model), shown in figure 11(b), reveals clearer spectral characteristics that are an attenuation of the keyboard typing sounds and also a dereverberation process by partially removing the reverberant artifacts that appear as temporal smearing.

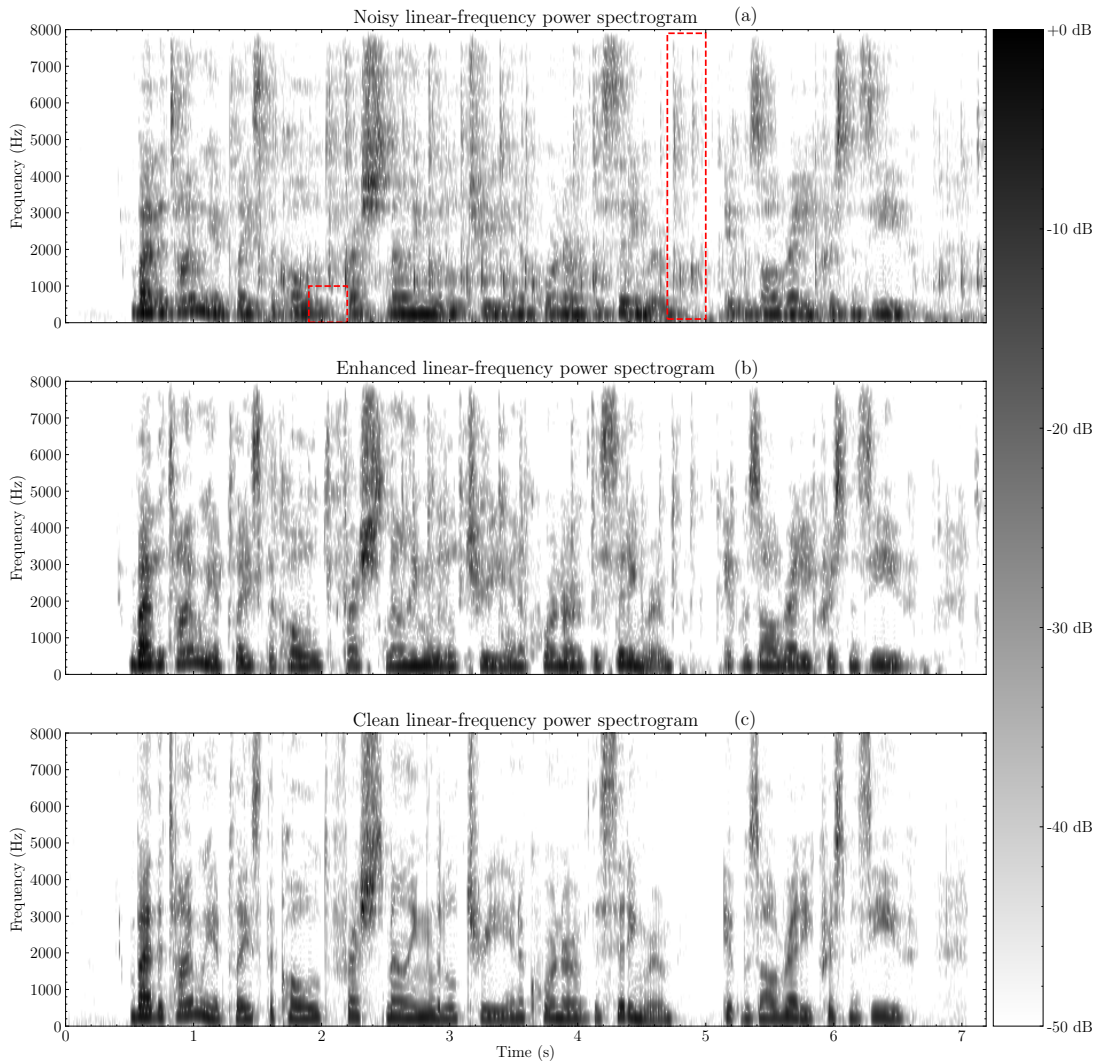


Figure 11: Wideband spectrogram plotted with linear frequency scales: (a) noisy speech; (b) enhanced speech by the FC2N (*wav2vec* 2.0), and (c) clean speech. In the noisy spectrogram, the red block on the left side represents a temporal gap filled due to the reverberation, and the red block on the right illustrates the sounds of two keystrokes.

### 5.3 Supplementary studies on the L3DAS22 dataset

After the ICASSP 2022 challenge, we further performed experiments on the L3DAS22 dataset to remark the impact of multiple loss functions on the intelligibility metrics.

Bellow, in table 4, we report the performances obtained for time, time-frequency and perceptual losses.

Table 4: Ablation study on Losses for 3D Speech Enhancement: We fixed all the components (FC2N model) and studied the impact of different losses on multiple domains (time, time-frequency and perceptual).

Loss	STOI	WER	$M$
$\mathcal{L}_{1-time}$	0.45	0.99	0.23
$\mathcal{L}_{2-time}$	0.57	0.89	0.34
$\mathcal{L}_{1-freq}$	0.78	0.26	0.76
$\mathcal{L}_{2-freq}$	0.75	0.40	0.68
$\mathcal{L}_{MRSTFT}$	0.73	0.39	0.67
$\mathcal{L}_{DFL}(wav2vec1.0)$	0.78	0.31	0.73
$\mathcal{L}_{DFL}(wav2vec2.0)$	0.77	0.33	0.72
$\mathcal{L}_{PFPL}$	0.45	0.99	0.23
$\mathcal{L}_{STOI}$	0.86	0.20	0.83
$\mathcal{L}_{CPL}(wav2vec1.0)$	0.84	0.22	0.81
$\mathcal{L}_{CPL}(wav2vec2.0)$	<b>0.86</b>	<b>0.19</b>	<b>0.84</b>

The experiments that compute regression losses directly on the clean and the reconstructed signals in the time-domain did not improve the noisy baseline (i.e., not using any algorithm). Moreover, it is interesting to notice that, even with the value loss function decreasing, many artifacts were introduced, harming the WER metric. We also hypothesize that the small differences introduced by the distance metric induce a gradient vanishing behavior, thus preventing the network from taking significant steps to find the minimum and make fine-grained adjustments. Finally, it is essential to state that those time-domain loss functions did not achieve great results with our proposed fully-convolutional architecture and optimization mechanism. However, it can be a viable alternative in other configurations.

On the other hand, the time-frequency losses showed us a promising result for the L3DAS22 dataset, especially the  $\mathcal{L}_{1-freq}$  loss, which is a consistent result with the speech enhancement and source separation literature (GUSÓ et al., 2022; GUIMARÃES; NAGANO; SILVA, 2020; PANDEY; WANG, 2018). Observing our three loss functions, we can see that the STOI metric has a different but not significant as the WER metric. We computed the  $\mathcal{L}_{MRSTFT}$  loss using five different STFT configurations, and more details on the used parameter can be found in the appendix B. The usage of narrowband and wideband spectrograms for the  $\mathcal{L}_{MRSTFT}$  loss seemed not to improve the final intelligibility metric

but slightly enhanced the WER compared to the  $\mathcal{L}_{2-freq}$  loss.

Our experiments on perceptual losses showed us the best result in the final metric, except for the  $\mathcal{L}_{PFPL}$ . For the PFPL, it is essential to remember that the Wasserstein distance on the latent space acts as a regularization for the reconstruction loss  $\mathcal{L}_{1-time}$ . Therefore, since the MAE on the time-domain did not show a reasonable performance, it is expected that the  $\mathcal{L}_{PFPL}$  would not either.

In our experiments with the Deep Feature Loss ( $\mathcal{L}_{DFL}(\cdot)$ ), we observed a significant improvement on the final metric compared to the pure regression loss  $\mathcal{L}_{2-freq}$ . To obtain this result, we used the embedding from the last transformer layer of the SSL models and set the hyperparameter  $\lambda = 10$ . By adding this comparison of high-level features of the latent space, we observed an increase in both the STOI metric but more notable in the WER, with a 22.5% and 17.5% improvement for the *wav2vec1.0* and *wav2vec2.0* model, respectively.

The final metric  $M$  is strongly related to intelligibility rather than a quality metric. The STOI loss function showed to be a robust objective function for both the L3DAS21 and L3DAS22 datasets. Our proposed compounded loss function ( $\mathcal{L}_{CPL}(\cdot)$ ) introduces a penalty term based on the distance of the latent representations from the clean and enhanced signals. Like the Deep Feature Loss, we used the MSE to compute the distance between the representations and set the hyperparameter  $\alpha = 10$ .

Both regularizers showed a similar metric  $M$ , but we expected that the *wav2vec2.0* should lead to a better WER metric. In fact, we can observe a slight improvement over the WER compared to only the  $\mathcal{L}_{STOI}$ , but it is not significant compared to the computational cost introduced. Since the  $\mathcal{L}_{STOI}$  loss function already compasses intelligibility characteristics, we hypothesized that this term could obfuscate the WER approximation. In future work, we would like to use a soft-attention mechanism to filter what are the most important dimensions (i.e., a learnable weight to select features) from the latent space from the *wav2vec2.0* in order only to use the most important ones to compute the distances and optimize for the metric  $M$ . In our current setup, we use all the features from the *wav2vec2.0* with the same importance, and some of them could not be relevant to our enhancement task.

Next, we evaluate the impact of choosing the proper distance function to measure similarity between two representations, and the results are presented in table 5. The Kullback–Leibler divergence (KLD) was introduced earlier in the manuscript with the VAE model, and an explanation can be found in the appendix A. Experimentally, we

measured the mean and standard deviation within each sample. Then, we used the analysis of variance (ANOVA) test with a 1% significance level to assess if the results were statistically significant. Note that, for ANOVA, it is essential to test for normality; in this work, we use the D’Agostino-Person test to assure that, also with 1% significance level. Our results indicate no significant impact on using different distance metrics between the representations. However, in the context of the challenges where the dataset is well-defined, and the scores rank the teams, we observed that the MSE and KLD showed better results for comparing the latent representations from the *wav2vec2.0* model as demonstrated in table 5.

Table 5: Ablation study on the distance metrics for the *wav2vec2.0* representations in the  $\mathcal{L}_{CPL}$ . The table shows the mean and standard deviation within samples. Fixed  $\alpha = 10$ , except for the Wasserstein distance where  $\alpha = 1$ .

Distance	STOI	WER	$M$
MAE	$0.85 \pm 0.054$	$0.20 \pm 0.216$	$0.83 \pm 0.123$
MSE	$0.86 \pm 0.053$	$0.19 \pm 0.211$	$0.84 \pm 0.121$
KLD	$0.86 \pm 0.054$	$0.19 \pm 0.215$	$0.84 \pm 0.125$
Wasserstein	$0.85 \pm 0.054$	$0.20 \pm 0.215$	$0.83 \pm 0.123$

In this part of the work, we did not extensively evaluate how the changes of the  $\alpha$  can impact the final metric  $M$  using this FC2N model. However, we speculate that the previous studies on the L3DAS21 dataset should also hold here. Increasing the  $\alpha$  value is beneficial because more regularization is introduced into the loss functions, forcing the model to improve the WER metric. However, large values can lead to a dominant term in the loss function, making the STOI to have a negligible impact on the final value.

## 6 CONCLUSION

In our set of experiments on 3D speech enhancement for the L3DAS21 data challenge, we first proposed a new model of FCN using as input the time domain speech signal. This strategy decreases the time of the training process and avoids phase problems during the reconstruction of the signal as usual in models that map input to output spectrograms.

We also evaluated the influence of the loss function in the SE task. A simple yet powerful model was capable of achieving a score of 0.73 in the competition metric (based on STOI and WER combination) on the development set, which represents a 20% improvement on the provided baseline (FaSNet) for the first challenge. With this novel loss function strategy, it is possible to improve the performance of DNN models for SE without modifying the structure, only changing the optimization method. Based on this method, the proposed model achieved second place in Task 1 of the L3DAS21 challenge.

On the experiments related to the second data challenge, we investigate the usage of deep complex networks for speech enhancement directly on the STFT representation. Our results show that this type of network represents a competitive alternative to traditional methods on time-domain or spectral magnitude. Furthermore, we also investigate the usage of the *wav2vec2.0* as a feature extractor for our perceptual losses, as well as the usage of  $L_2$  norm to compute the distance between the learned representations. With this strategy, we can achieve a score of 0.843 in the proposed metric.

As expected by literature on speech enhancement methods, the usage of neural networks with both time or time-frequency domain inputs can improve the final intelligibility metric. However, models with an STFT input seem to achieve the best results currently, even though that models that directly operate on the waveform are gaining popularity, primarily due to computational cost. Furthermore, compared to our time-domain approach, the model using STFT is four times slower due to the adjustments necessary to make the model converges.

Lastly, we studied the impacts of different loss functions on the L3DAS22 dataset. In

conclusion, we highlight the usage of perceptual loss functions to improve intelligibility metrics, in order: first, the compounded loss function  $\mathcal{L}_{CPL}(\text{wav2vec2.0})$  with the MSE distance in the latent space; the  $\mathcal{L}_{STOI}$  loss function; and the  $\mathcal{L}_{DFL}(\text{wav2vec1.0})$ . For deep complex models estimating a mask for the STFT input, we do not recommend using loss functions directly on the time domain because it may need a huge effort to make the model converge towards an intelligibility improvement.

All the tested metrics (MAE, MSE, KLD, or Wasserstein) achieved good results with your designed system in the ablation study of distance metrics. Although no significant statistical discrepancies are found using different distance metrics between representations, in a restricted challenge scenario, where the number of samples is limited and the importance of slight performance deviations, the MSE and KLD metrics are preferred in this scenario.

For future work, for the 3D SE task, we would like to investigate the usage of a soft-attention mask on the extracted representations from the *wav2vec2.0* model. Also, it would be interesting to experiment if our previous conclusions on the loss functions hold for different types of models (e.g., spectral subtraction models) and more advanced deep complex architectures.

# APPENDIX A – FUNDAMENTALS OF INFORMATION THEORY

This section is a shallow introduction to some concepts related to information theory. The concepts related to entropy, relative entropy, and mutual information are fundamental in communication systems (e.g., data compression and transmission) and deep neural networks. Moreover, those concepts are deeply related to the theories developed by Shannon.

In these notes, we define some connections to the definition of entropy in thermodynamics and derive the definition axiomatically by properties that a random variable satisfies (COVER, 1999).

**Definition A.0.1** (Independence).  $I(A, B) = I(A) + I(B)$

**Definition A.0.2** (Monotonicity).  $A \geq B \implies I(B) \geq I(A)$

**Definition A.0.3** (Non-negativity).  $I(\cdot) \geq 0$

**Definition A.0.4** (Certainty).  $P(A) = 1 \implies I(A) = 0$

Let  $X$  be a discrete random variable with a probability mass function (pmf)  $p_X(x_k)$  with an alphabet  $\mathcal{X}$ , where  $x_k \in \mathcal{X}$ , and  $|\mathcal{X}| = K$ . We define self-information of the symbol  $x_k$  in the equation A.1. Note that this definition complies with all the previous properties.

$$\begin{aligned} I(x_k) &= \log_2 \left( \frac{1}{p(x_k)} \right) \\ &= -\log_2 p(x_k) \end{aligned} \tag{A.1}$$

We define entropy as a measure of the uncertainty of a random variable, or the expected-value of the self-information over its alphabet, as described by equation A.2, where the entropy is measured in bits per symbol.



$$\begin{aligned}
H(X) &= \mathbb{E}[I(X)] \\
&= \sum_{k=1}^K p_x(k) I(x_k) \\
&= - \sum_{k=1}^K p(k) \log_2 p(k)
\end{aligned} \tag{A.2}$$

Furthermore, we extend the entropy measurements for a pair of random variables  $X$  and  $Y$ , with alphabets  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively.

**Definition A.0.5** (Joint Entropy).

$$H(X, Y) = - \mathbb{E}[\log_2 p(x, y)] \tag{A.3}$$

**Definition A.0.6** (Conditional Entropy).

$$\begin{aligned}
H(Y|X) &= - \mathbb{E}[\log_2 p(y|x)] \\
&= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(y|x)
\end{aligned} \tag{A.4}$$

Consequently, based on the previous definitions, we can now specify the **relative entropy**, a measurement to define the similarity between two probability distributions described as the expected value of the logarithm of the likelihood ratio, as shown in equation A.5. Relative entropy is a fundamental concept to understand latent variable models further since we introduce restraints over the distribution related to the latent space of this type of model.

**Definition A.0.7** (Relative Entropy or Kullback–Leibler divergence).

$$\begin{aligned}
D(p||q) &= \mathbb{E}_p \left[ \frac{p(x)}{q(x)} \right] \\
&= \sum_{x \in \mathcal{X}} p(x) \log_2 \left( \frac{p(x)}{q(x)} \right)
\end{aligned} \tag{A.5}$$

Finally, we introduce our last definition, the **mutual information**. The mutual information between two random variables is the information we obtain from one of them about the other.

**Definition A.0.8** (Mutual Information).

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{p(x,y)} \left[ \log_2 \left( \frac{p(X,Y)}{p(X)p(Y)} \right) \right] \\ &= D( p(x,y) \parallel p(x)p(y) ) \end{aligned} \tag{A.6}$$

Notice that when  $X$  and  $Y$  are independent,  $p(X,Y) = p(X)p(Y) \implies I(X;Y) = 0$ . To summarize, we show in figure 12 a Venn diagram summing up the discussions of the above definitions about a pair of random variables.

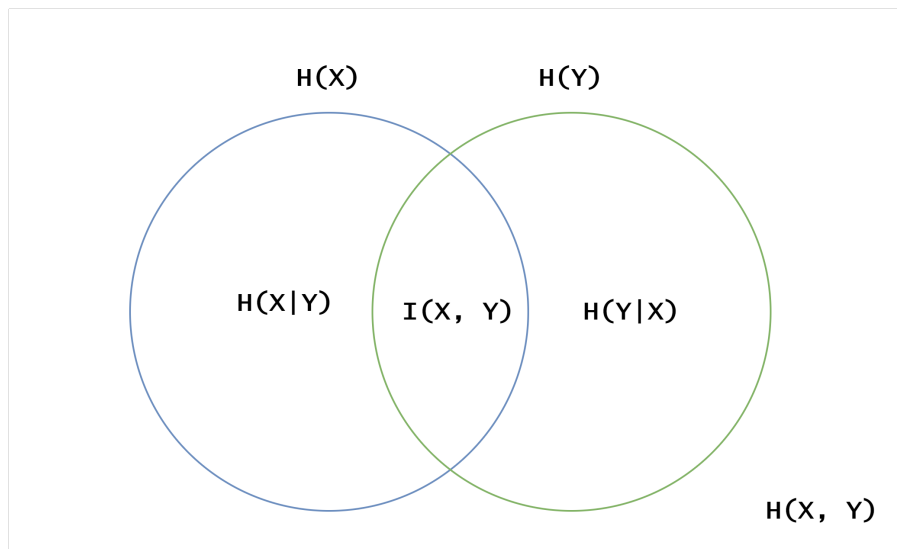


Figure 12: Venn diagram for Mutual information and associated entropies. Adapted from Prof. Arjona classes (PSI5813).

Another significant result from the mutual information related to supervised learning is introduced by Naftali (TISHBY; ZASLAVSKY, 2015). Let a pair of random variables  $X \in \mathcal{X}$  and  $Y \in \mathcal{Y}$  be our explanatory and exogenous variables, respectively. Their relevant information is defined as the mutual information  $I(X;Y)$ , where a statistical dependence exists between them.  $Y$  implicitly determines the relevant and irrelevant features of  $X$ , which an optimal representation of  $X$  should capture while compressing the irrelevant ones that are noises to predict  $Y$ .

## APPENDIX B – MULTI-RESOLUTION STFT LOSS FUNCTION - SETUP

To implement the  $\mathcal{L}_{\text{MRSTFT}}$  loss, we defined 5 sets of parameters related to the STFT construction, defined as follow:

- **STFT A:** A FFT size of 512, a window size of 32 ms, and 50% hop (256 samples)
- **STFT B:** A FFT size of 512, a window size of 6 ms, and 10.4% hop (10 samples)
- **STFT C:** A FFT size of 1024, a window size of 60 ms, and 10% hop (96 samples)
- **STFT D:** A FFT size of 1024, a window size of 10 ms, and 10% hop (16 samples)
- **STFT E:** A FFT size of 2048, a window size of 30 ms, and 33.3% hop (160 samples)

All of the sets used an Hann window and  $\lambda_1 = \lambda_2 = 0.5$ . All the values were chose empirically based on common configurations for STFT extraction.

## APPENDIX C – VARIATIONAL AUTOENCODERS

In the chapter 2, we introduced the difference between tractable and intractable models. Intractable models are more expressive but rely on hard assumptions about the data or latent distributions. In order to enable approximate inference, usually we rely on two methods. The first one is the Markov Chain Monte Carlo (MCMC), which is the best method for generate samples from the exact posterior. It is a general method that gives the exact answer in the limit of infinite time. However, it is computationally expensive and its convergence is hard to detect. On the other hand, we have Variational Inference, which we approximate the posterior with a tractable distribution.

The main idea is to approximate the exact posterior  $p_\theta(\mathbf{z}|\mathbf{x})$  with a variational posterior that we can control  $q_\phi(\mathbf{z}|\mathbf{x})$ , where  $\phi$  are the variational parameters to be optimize to get close as possible to the exact posterior (MNIH, 2020). On the variational posterior  $q_\phi(\mathbf{z}|\mathbf{x})$ , it is necessary:

1. To choose a distribution where we can sample from it
2. To be able to compute  $\log(q_\phi(\mathbf{z}|\mathbf{x}))$
3. The derivatives with respect to  $\phi$  exists

This is not a necessary condition, but usually we rely on a variational posterior that can be factorized, i.e.  $q_\phi(\mathbf{z}|\mathbf{x}) = \prod_i q_\phi(\mathbf{z}_i|\mathbf{x})$ .

The fundamental aspect about using a variational posterior is the induction of a **variational lower bound**  $\mathcal{L}_{\theta,\phi}$  on the marginal log-likelihood  $\log p_\theta(\mathbf{x})$ . In this setting, we train a model to maximize  $\mathcal{L}_{\theta,\phi}$  by optimizing both  $\theta$  and  $\phi$  (ERMON; SONG, 2021). The name lower bound is because it is guaranteed that is bellow the value of the marginal log-likelihood, even though that we can not compute  $\log p_\theta(\mathbf{x})$  directly. The derivation of the marginal log-likelihood is described in equation C.1.

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int q_\phi(\mathbf{z}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})} d\mathbf{z} \\
&\geq \int q_\phi(\mathbf{z}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})} d\mathbf{z} \\
&= \mathbb{E}_{q_\phi(\mathbf{z})} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})} \right]
\end{aligned} \tag{C.1}$$

From the equation C.1, an important result is that

$$\mathcal{L}_{\theta, \phi} = \mathbb{E}_{q_\phi(\mathbf{z})} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})} \right] \leq \log p_\theta(\mathbf{x})$$

which means that, for an arbitrary  $q_\phi(\mathbf{z})$ , we have a lower-bound on the marginal log-likelihood. Moreover, we can get a closer boundary by maximizing this expression with respect to  $\phi$  (MNIH, 2020).

Since  $q_\phi(\mathbf{z})$  is an arbitrary distribution, the most popular choice for this is the variational posterior  $q_\phi(\mathbf{z}|\mathbf{x})$ . This is called the **Evidence Lower Bound**. Using this distribution, we can rewrite the variational lower bound  $\mathcal{L}_{\theta, \phi}$  as:

$$\begin{aligned}
\mathcal{L}_{\theta, \phi} &= \mathbb{E}_{q_\phi(\mathbf{z})} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z})} \left[ \log \frac{p_\theta(\mathbf{x})p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \\
&= \log p_\theta(\mathbf{x}) - D(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))
\end{aligned} \tag{C.2}$$

The Kullback–Leibler divergence between the two distributions is the variational gap and represents the difference between the true and the variational posterior. Another significant result from the above equation is that we are minimizing the variational gap when maximizing the ELBO for  $\phi$ .

Towards a more practical way of training models using variational inference, the next step is to discuss how to compute the gradients for the ELBO. One of the possible methods to compute the gradients it is through an operation called **reparameterization trick**, where the main idea is to reparameterize a sample from  $q_\phi(\mathbf{z}|\mathbf{x})$  distribution as function from a sample with a fixed distribution, usually a Gaussian, Laplace or Cauchy distribution.

Variational Autoencoder (VAE) was introduced in 2014 (KINGMA; WELLING, 2013).

The idea is to use neural networks to learn the parameters  $\theta$  and  $\phi$  from the prior  $p_\theta(\mathbf{x}|\mathbf{z})$  and the variational posterior  $q_\phi(\mathbf{z}|\mathbf{x})$ , using as principle the variational inference procedure and the reparametrization trick.

The components of the VAE are (i) the prior  $p(\mathbf{z})$ , which is follows the distribution chose for the reparametrization trick; (ii) the encoder, which is responsible to compute the variational posterior  $q_\phi(\mathbf{z}|\mathbf{x})$ ; (iii) and the decoder, which is related to the likelihood  $p_\theta(\mathbf{x}|\mathbf{z})$ . Different types of neural networks (e.g., CNNs and RNNs) can be used as encoder and decoder in the VAE framework.

## REFERENCES

- BAEVSKI, Alexei et al. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. **CoRR**, abs/2006.11477, 2020. arXiv: 2006.11477. Available from: <https://arxiv.org/abs/2006.11477>.
- BAIN, Alexander. **Mind and body: The theories of their relation**. [S.l.]: D. Appleton, 1873. v. 4.
- BENESTY, Jacob. **Fundamentals of Speech Enhancement**. 1. ed. [S.l.]: Springer International Publishing, 2018. (SpringerBriefs in Electrical and Computer Engineering). ISBN 978-3-319-74523-7, 978-3-319-74524-4.
- BENGIO, Yoshua; COURVILLE, Aaron; VINCENT, Pascal. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 8, p. 1798–1828, 2013.
- BROWN, Tom B et al. Language models are few-shot learners. **arXiv preprint arXiv:2005.14165**, 2020.
- CHOROWSKI, Jan et al. Unsupervised speech representation learning using wavenet autoencoders. **IEEE/ACM transactions on audio, speech, and language processing**, IEEE, v. 27, n. 12, p. 2041–2053, 2019.
- COBBE, Karl et al. Quantifying generalization in reinforcement learning. In: PMLR. INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2019. p. 1282–1289.
- COVER, Thomas M. **Elements of information theory**. [S.l.]: John Wiley & Sons, 1999.
- DEVLIN, Jacob et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.
- DOMINGOS, Pedro. Every model learned by gradient descent is approximately a kernel machine. **arXiv preprint arXiv:2012.00152**, 2020.
- ERMON, Stefano; SONG, Yang. **Lecture notes in Deep Generative Models (CS236)**. [S.l.]: Stanford University, Sept. 2021.
- FONSECA, Eduardo et al. FSD50k: an open dataset of human-labeled sound events. **arXiv preprint arXiv:2010.00475**, 2020.

- GERMAIN, Francois G; CHEN, Qifeng; KOLTUN, Vladlen. Speech denoising with deep feature losses. **arXiv preprint arXiv:1806.10522**, 2018.
- GHOSH, Sayan et al. Representation Learning for Speech Emotion Recognition. In: INTERSPEECH. [S.l.: s.n.], 2016. p. 3603–3607.
- GOODFELLOW, Ian; POUGET-ABADIE, Jean, et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, 2014.
- GOODFELLOW, Ian J; SHLENS, Jonathon; SZEGEDY, Christian. Explaining and harnessing adversarial examples. **arXiv preprint arXiv:1412.6572**, 2014.
- GROVER, Aditya; ERMON, Stefano. Uncertainty autoencoders: Learning compressed representations via variational information maximization. In: PMLR. THE 22nd International Conference on Artificial Intelligence and Statistics. [S.l.: s.n.], 2019. p. 2514–2524.
- GUIMARÃES, Heitor R; BECCARO, Wesley; RAMÍREZ, Miguel A. Optimizing Time Domain Fully Convolutional Networks for 3D Speech Enhancement in a Reverberant Environment Using Perceptual Losses. In: IEEE. 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP). [S.l.: s.n.], 2021. p. 1–6.
- GUIMARÃES, Heitor R.; NAGANO, Hitoshi; SILVA, Diego W. Monaural Speech Enhancement Through Deep Wave-U-net. **Expert Systems with Applications**, v. 158, p. 113582, 2020. ISSN 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2020.113582>. Available from: <https://www.sciencedirect.com/science/article/pii/S0957417420304061>.
- GUIZZO, Eric; GRAMACCIONI, Riccardo F., et al. L3DAS21 Challenge: Machine Learning for 3D Audio Signal Processing. **arXiv:2104.05499 [cs, eess]**, Apr. 2021. arXiv: 2104.05499. Available from: <http://arxiv.org/abs/2104.05499>. Visited on: 14 June 2021.
- GUIZZO, Eric; MARINONI, Christian, et al. L3DAS22 Challenge: Learning 3D Audio Sources in a Real Office Environment. In: IEEE international Conference on Acoustics, Speech and Signal Processing (ICASSP). Singapore: [s.n.], May 2022.
- GUSÓ, Enric et al. On loss functions and evaluation metrics for music source separation. **arXiv preprint arXiv:2202.07968**, 2022.
- HE, Kaiming et al. Deep residual learning for image recognition. In: PROCEEDINGS of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. p. 770–778.



- HINTON, Geoffrey E; SALAKHUTDINOV, Ruslan R. Reducing the dimensionality of data with neural networks. **science**, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.
- HOFMANN, Thomas; SCHÖLKOPF, Bernhard; SMOLA, Alexander J. Kernel methods in machine learning. **The annals of statistics**, JSTOR, p. 1171–1220, 2008.
- HSIEH, Tsun-An et al. Improving perceptual quality by phone-fortified perceptual loss for speech enhancement. **arXiv preprint arXiv:2010.15174**, 2020.
- HSIEH, Tsun-An et al. Improving Perceptual Quality by Phone-Fortified Perceptual Loss using Wasserstein Distance for Speech Enhancement. **arXiv:2010.15174 [cs, eess]**, Apr. 2021. Available from: <<http://arxiv.org/abs/2010.15174>>. Visited on: 31 May 2021.
- HSU, Wei-Ning et al. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. **arXiv e-prints**, arxiv–2106, 2021.
- JANG, Eric; GU, Shixiang; POOLE, Ben. Categorical reparameterization with gumbel-softmax. **arXiv preprint arXiv:1611.01144**, 2016.
- JOHNSON, Justin; ALAHI, Alexandre; FEI-FEI, Li. Perceptual losses for real-time style transfer and super-resolution. In: SPRINGER. EUROPEAN conference on computer vision. [S.l.: s.n.], 2016. p. 694–711.
- KINGMA, Diederik P; WELING, Max. Auto-encoding variational bayes. **arXiv preprint arXiv:1312.6114**, 2013.
- KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, p. 1097–1105, 2012.
- KUHN, Max; JOHNSON, Kjell. **Feature engineering and selection: A practical approach for predictive models**. [S.l.]: CRC Press, 2019.
- LAKE, Brenden M et al. Building machines that learn and think like people. **Behavioral and brain sciences**, Cambridge University Press, v. 40, 2017.
- LECUN, Yann; MISRA, Ishan. **Self-supervised learning: The dark matter of intelligence**. [S.l.]: Facebook AI, 2021. <https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/>. Accessed: 2021-12-05.
- LI, Jingdong et al. The PCG-AIID System for L3DAS22 Challenge: MIMO and MISO Convolutional Recurrent Network for Multi Channel Speech Enhancement and Speech Recognition. In: IEEE. ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], 2022. p. 9211–9215.

- LIU, Yinhan et al. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, 2019.
- LOIZOU, Philipos C. **Speech Enhancement**. [S.l.]: CRC Press, Feb. 2013. DOI: 10.1201/b14529. Available from: <<https://doi.org/10.1201/b14529>>.
- LU, Xugang et al. Speech Enhancement Based on Deep Denoising Auto-Encoder. **Proc. Interspeech**, p. 436–440, Jan. 2013.
- LU, Yen-Ju et al. Towards Low-Distortion Multi-Channel Speech Enhancement: The ESPNET-Se Submission to the L3DAS22 Challenge. In: IEEE. ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], 2022. p. 9201–9205.
- LUO, Yi et al. FaSNet: Low-latency Adaptive Beamforming for Multi-Microphone Audio Processing. In: IEEE. 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). [S.l.: s.n.], 2019. p. 260–267.
- MACARTNEY, Craig; WEYDE, Tillman. Improved Speech Enhancement with the Wave-U-Net. en. **arXiv:1811.11307 [cs, eess]**, Nov. 2018. Available from: <<http://arxiv.org/abs/1811.11307>>. Visited on: 31 May 2021.
- MANUEL, Pariente. **Implementation of the classical and extended Short Term Objective Intelligibility in PyTorch**. [S.l.]: GitHub, 2021.
- MITCHELL, T.M. **Machine Learning**. [S.l.]: McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Available from: <<https://books.google.com.br/books?id=EoYBngEACAAJ>>.
- MNIH, Andriy. **The Deep Learning Lecture Series 2020**. [S.l.]: DeepMind and UCL, 2020.
- NIKZAD, Mohammad et al. Deep Residual-Dense Lattice Network for Speech Enhancement. **arXiv:2002.12794 [cs, eess, stat]**, Feb. 2020. Available from: <<http://arxiv.org/abs/2002.12794>>. Visited on: 31 May 2021.
- OORD, Aaron van den; VINYALS, Oriol, et al. Neural discrete representation learning. In: ADVANCES in Neural Information Processing Systems. [S.l.: s.n.], 2017. p. 6306–6315.
- OORD, Aaron van den; DIELEMAN, Sander, et al. Wavenet: A generative model for raw audio. **arXiv preprint arXiv:1609.03499**, 2016.
- OORD, Aaron van den; LI, Yazhe; VINYALS, Oriol. Representation learning with contrastive predictive coding. **arXiv preprint arXiv:1807.03748**, 2018.

- PANAYOTOV, Vassil et al. Librispeech: an asr corpus based on public domain audio books. In: IEEE. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], 2015. p. 5206–5210.
- PANDEY, Ashutosh; WANG, Deliang. On adversarial training and loss functions for speech enhancement. In: IEEE. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], 2018. p. 5414–5418.
- PARK, Se Rim; LEE, Jin Won. A Fully Convolutional Neural Network for Speech Enhancement. In: PROC. Interspeech 2017. [S.l.: s.n.], 2017. p. 1993–1997. DOI: 10.21437/Interspeech.2017-1465. Available from: <<http://dx.doi.org/10.21437/Interspeech.2017-1465>>.
- RAVANELLI, Mirco et al. SpeechBrain: A General-Purpose Speech Toolkit. **arXiv:2106.04624 [cs, eess]**, June 2021. arXiv: 2106.04624. Available from: <<http://arxiv.org/abs/2106.04624>>. Visited on: 12 June 2021.
- REN, Xinlei et al. A Neural Beamforming Network for B-Format 3D Speech Enhancement and Recognition. In: IEEE. 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP). [S.l.: s.n.], 2021. p. 1–6.
- SAHAI, Abhimanyu; WEBER, Romann; MCWILLIAMS, Brian. Spectrogram feature losses for music source separation. In: IEEE. 2019 27th European Signal Processing Conference (EUSIPCO). [S.l.: s.n.], 2019. p. 1–5.
- SANTOS, Joao Felipe; FALK, Tiago H. Speech dereverberation with context-aware recurrent neural networks. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, IEEE, v. 26, n. 7, p. 1236–1246, 2018.
- SCHNEIDER, Steffen et al. wav2vec: Unsupervised Pre-Training for Speech Recognition. en. In: INTERSPEECH 2019. [S.l.]: ISCA, Sept. 2019. p. 3465–3469. DOI: 10.21437/Interspeech.2019-1873. Available from: <[http://www.isca-speech.org/archive/Interspeech\\_2019/abstracts/1873.html](http://www.isca-speech.org/archive/Interspeech_2019/abstracts/1873.html)>. Visited on: 31 May 2021.
- SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- TAAL, Cees H. et al. A Short-Time Objective Intelligibility Measure for Time-Frequency Weighted Noisy Speech. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing. [S.l.: s.n.], 2010. p. 4214–4217. DOI: 10.1109/ICASSP.2010.5495701.
- TISHBY, Naftali; ZASLAVSKY, Noga. Deep learning and the information bottleneck principle. In: IEEE. 2015 IEEE Information Theory Workshop (ITW). [S.l.: s.n.], 2015. p. 1–5.

- TRABELSI, Chiheb et al. Deep complex networks. **arXiv preprint arXiv:1705.09792**, 2017.
- VALENTINI-BOTINHAO, Cassia et al. Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech. In: SSW. [S.l.: s.n.], 2016. p. 146–152.
- WOLF, Thomas et al. Huggingface’s transformers: State-of-the-art natural language processing. **arXiv preprint arXiv:1910.03771**, 2019.
- XU, Yong et al. A Regression Approach to Speech Enhancement Based on Deep Neural Networks. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, v. 23, n. 1, p. 7–19, 2015. DOI: 10.1109/TASLP.2014.2364452.
- YAMAMOTO, Ryuichi; SONG, Eunwoo; KIM, Jae-Min. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In: IEEE. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], 2020. p. 6199–6203.
- YANG, Shu-wen et al. SUPERB: Speech processing Universal PERFORMANCE Benchmark. **arXiv preprint arXiv:2105.01051**, 2021.
- ZHANG, Guochang et al. Multi-scale temporal frequency convolutional network with axial attention for speech enhancement. In: IEEE. ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], 2022. p. 9122–9126.