ON CONTROLLED DE-ENTANGLEMENT

TOWARDS FLEXIBILITY IN NEURAL GENERATIVE MODELS

By

SAI KRISHNA RALLABANDI

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

CARNEGIE MELLON UNIVERSITY
School of Computer Science

MAY 2021

To the Faculty of Carnegie Mellon University:

The members of the Committee appointed to examine the dissertation of SAI KRISHNA RALLABANDI find it satisfactory and recommend that it be accepted.

Alan W Black, Ph.D., Chair

LP Morency, Ph.D.

Eric Nyberg, Ph.D.

Kalika Bali, Ph.D.

ACKNOWLEDGMENT

ToDo

ON CONTROLLED DE-ENTANGLEMENT

TOWARDS FLEXIBILITY IN NEURAL GENERATIVE MODELS

Abstract

by Sai Krishna Rallabandi, Ph.D.
Carnegie Mellon University
May 2021

: Alan W Black

In this thesis, I present an argument for De-Entanglement: a property that has potential to isolate the factors of variation in the data distribution. I am interested in knowing if explicitly isolating relevant factors using such an approach is helpful with respect to downstream tasks. I first highlight three different approaches to accomplish 'De-Entanglement'. I then present one case study per approach to investigate the importance of such an approach. I conclude by arguing that while this serves as a neat framework to build systems, such an approach might not always be applicable or necessary. I conclude by arguing that while this serves as a neat framework to build systems, such an approach might not always be applicable or necessary. I conclude by arguing that while this serves as a neat framework to build systems, such an approach might not always be applicable or necessary

TABLE OF CONTENTS

LIST OF TABLES

Dedication

This dissertation/thesis is dedicated to all those willing to take the first steps on faith

Chapter One

Introduction

- Exploiting deep learning, we can extract various types of information from a data distribution.
- Extracting relevant information is useful in a large number of contexts such as story generation, targeted captioning, etc.
- De-Entanglement is a framework aimed at extracting relevant information by systematically isolating the factors of variation.
- Factors could be user defined - from attributes such as color, accent all the way upto schema such as Winograd or recipe stages.
- This thesis is concerned with developing tools to solve this
- Having developed these tools, show that can be useful for a downstream task as well. VQA, Audio search.

1.1 Motivation

Ability to extract useful information is important

Often we are faced with scenarios where we need to control generation of content. For

example, translation is a scenario where we need to focus on the particular word, etc. Summarization is another. Imagine having to deliver content through voice media such as Alexa or Siri. It would be nice to be able to summarize content and tell what is relevant instead of blurting out everything. In other cases, we need to expand. Consider any task that involves conversion of structured data such as table to unstructured such as free form text or speech. Imagine going on a space ship and the robot telling you everything is doomed. In other cases, we need to filter something out. Consider speech enhancement or source separation as example tasks.

Challenges for controlled generation

- Sparsity: The control factors are under specified

1.1.1 De-Entanglement

Imagine a property of a model of the world referred to as De-Entanglement. It is (currently) defined as the ability to isolate the factors of variation which perhaps were involved in the design of the world itself. It is easy to see that such a property is extremely desirable in a model. Consider kids playing with Lego toys as opposed to a static toy like a TeddyBear or a Barbie. The freedom to dismantle the structure apart and re-compose variants of it has been shown to improve creativity Gauntlett, 2014. The implications become even more apparent when we consider a real life application such as speech processing. It is extremely difficult to reason about speech in the time domain by inspecting the individual samples. However, transforming the same utterance into frequency domain by applying Fourier Transform - a process that isolates the contribution from individual frequencies - makes reasoning easier, to the point of even identification of the individual linguistic units within the utterance. In this context, the individual frequencies and their contributions are the factors of variation in the generative process of speech data. The observation can be extended to other types of data as well. Consider spectroscopy: The ability to spectrally decompose (visible) light enables

estimation of cosmic evolution of celestial bodies Keller et al., 2014. The argument presented above claims that such isolation should invariably help downstream tasks. However, this does not appear to be always true. Consider as example the task of adding two natural numbers. Perhaps an appropriate de-entanglement for a model aimed at completing this task involves Peano axioms Skolem, 1955. But we as humans have been conditioned to solve this task by cumulative addition of individual digits with appropriate carryover and not necessarily following Skolem, 1955. Similarly consider the inner workings of AlphaZero Silver et al., 2017. It is not clear if the self learning based algorithm is accomplishing an isolation of relevant factors of variation in the latent space. Moreover, there are scenarios where estimation of causal factors is intractable. In such scenarios, it appears hard to comment about performance with respect to a concept like de-entanglement.

Within the scope of my work, I am interested in investigating the extent to which isolation of factors of variation as mentioned above is plausible and useful in the context of Natural Language Processing(NLP). In this context, 'De-Entanglement' refers to the ability of a model to isolate the relevant causal factors of variation in the joint distribution spanned by the input and output distributions defined by the task at hand. Specifically, I am interested in answering some of the following research questions:

- What are the scenarios where de-entanglement helps solve the task?
- In cases where true, does de-entanglement help solve the task more efficiently? How is efficiency manifested? In making the model more compact? Making the algorithm faster?
- In cases where true and de-entanglement does not result in a more efficient solution, why does this happen?
- What are the scenarios where de-entanglement cannot help solve the problem? Is it due to probabilities becoming too miniscule? Is it because the calculations seem

implausible given the current compute?

- In cases where de-entanglement cannot be applied but seems reasonable, can we reformulate the problem or task so that we can apply de-entanglement?
- Are there cases where de-entanglement hurts the model? Does it do so by limiting the expressivity of models? Are there any model blindsplots in these scenarios?
- Why is de-entanglement preferable? Is it since it avoids adversarial attacks?
- What are some of the challenges for de-entanglement? What is difficult about it? sparsity? lack of ability to identify the factors of variation? example: sentiment analysis
- What are the approaches to accomplish de-entanglement?

1.2 Organization of Thesis

In this thesis, I propose to extract the following types of information from the distribution of data:

- Content. In this thesis, I specifically focus on acoustic phonetic content. I show examples from Acoustic Unit Discovery and Source separation.
- Style. In this thesis, I focus on paralinguistic style of an utterance. I show examples from sleepiness detection, valence and arousal detection. I apply the findings to code mixed speech synthesis.
- Structure. I specifically concern myself with prosodic structure of an utterance. I present examples from image captioning and emphasis based text to speech.

Chapter Two

De-Entanglement

2.1 How to accomplish de-entanglement?

The most popular approach to obtain isolation of factors of variation in neural models is by employing stochastic random variables. This approach provides flexibility to jointly train the latent representations as well as the downstream network. It has been observed that the latent representations resemble disentangled representations under certain conditions (T. Q. Chen et al., 2018; Burgess et al., 2018a; Esmaeili et al., 2018; Ansari and Soh, 2018). Note that although obtaining such degenerate representations is considered typical, it is not the only manifestation: it also manifests as continuous representations(Ravanelli and Y. Bengio, 2018) and other abstract phenomena(e.g. grounding). I argue that explicitly controlling what and how much gets de-entangled (Burgess et al., 2018a) is better than implicit disentanglement as is followed today(Locatello et al., 2018). I identify four ways to computationally control de-entanglement in encoder decoder models

- (1) By employing suitable priors about task or data distribution
- (2) By incorporating additional adversarial or multi task objectives within the model
- (3) By utilizing a different divergence objective
- (4) By employing an alternative formulation of probability density estimation

I will expand on each of these in the following chapters, providing one task from language technologies as a case study.

I posit that designing learning paradigms such that we explicitly control de-entanglement of relevant factors of variation while marginalizing the nuisance factors of variation leads to massive improvements. Such an approach, I claim, leads to further advantages in the context of both generative processes: in terms of generation of novel content and discriminative processes: in terms of robustness of such models to noise and attacks. Let us consider a typical deep learning architecture such as AlexNet(Krizhevsky, Sutskever, and Hinton, 2012). It is characterized by a series of convolutional layers (feature extraction module) followed by a pooling layer and a SoftMax layer(classification module). Note that while I mention AlexNet as an example, this abstraction can be extended to most sequence to sequence architectures with encoder as feature extraction module and decoder as the classification module(Rousseau and Tsaftaris, 2019) across modalities and tasks. It can be shown that the pooling layer acts as information bottleneck(Tishby, Pereira, and Bialek, 2000) module in such architectures. I point out(S. K. Rallabandi and A. Black, 2019) that in case of conventional Seq2Seq architectures deployed today, attention plays the role of information bottleneck module regulating the amount of information being utilized by the decoder. In (S. K. Rallabandi and A. Black, 2019; Vyas et al., 2019) I show that this module controls optimization in encoder decoder models leading to (1) Disentanglement of Causal Factors of variation in the data distribution (2) Marginalization of nuisance factors of variation from the input distribution. In case of models that employ stochasticity, two more effects can be observed : (a) Posterior collapse or Degeneration due to powerful decoders and (b) Loss of output fidelity due to finite capacity decoders. In current architectures, marginalization and disentanglement are realized implicitly and often lead to (a) and (b) when deployed in practise.

The most popular approach to obtain isolation of factors of variation in neural models is by employing stochastic random variables. This approach provides flexibility to jointly train

the latent representations as well as the downstream network. It has been observed that the latent representations resemble disentangled representations under certain conditions (T. Q. Chen et al., 2018; Burgess et al., 2018a; Esmaeili et al., 2018; Ansari and Soh, 2018). Note that although obtaining such degenerate representations is considered typical, it is not the only manifestation: it also manifests as continuous representations(Ravanelli and Y. Bengio, 2018) and other abstract phenomena(e.g. grounding). I argue that explicitly controlling what and how much gets de-entangled (Burgess et al., 2018a) is better than implicit disentanglement as is followed today(Locatello et al., 2018). I identify four ways to computationally control de-entanglement in encoder decoder models

- (1) By employing suitable priors about task or data distribution
- (2) By incorporating additional adversarial or multi task objectives within the model
- (3) By utilizing a different divergence objective
- (4) By employing an alternative formulation of probability density estimation

2.1.1 Case for Controlled De-Entanglement

I believe that complete disentanglement of input data into its independent causal factors of variation is not fully useful. A more attractive option is to control what and how much gets de-entangled in a a task dependent manner. It has to be noted that given a particular downstream task, some causal factors of variation might not be relevant, in which case modeling them would be unnecessary. Let us consider a data distribution X which consists of class examples $\{x_1, x_2..., x_n\}$, where each x_i is described by attribute-set (a, b, c). The prior distribution of X can be represented by a parameteric function g such that g maximizes the likelihood of X over the set of its attributes:

$$P_\omega(X) = g_\omega(a, b, c) \tag{2.1}$$

Note that the attribute-set can either contain individual entities or the relationships between them or both. To illustrate this, let us consider a toy-example where we build a binary classifier to predict if a given integer triplet is a Pythagorean triplet. Pythagorean triplets are a triplet of numbers that follow Pythagoras Theorem such as $\{3, 4, 5\}$ and $\{5, 12, 13\}$. In this task, the attribute-set consists of the relationship between the first two-elements of the triplet. If the model is able to discover this attribute, it can generalize for any given numbers. However, if we have a more complicated task like building a classifier for MNIST digits, then the attribute-set has multiple first and second order relations like brush strokes, shape of the digits etc. The success of modelling $P_\omega(X)$, and ultimately the success on the downstream task, relies on how well can the model isolate these individual attributes from the observed data X_t. This isolation ability becomes even more important in case we want to regenerate the digits using a generative model. Mathematically, let us consider the posterior probability of a training instance x_1 expressed as

$$P_\theta(x_1) = f_\theta(x_1) \tag{2.2}$$

where f denotes arbitrary function and θ denotes the parametric family used to model the distribution X. It can be seen that compositionality over an unseen training instant x_{new} would be possible if f is related to g. In other words, f needs to intuitively have some information about the latent causal factors of variation that generated X in the first place. In such scenarios, the test instance can be appropriately expressed as

$$P_\theta(x_{new}) = h(a, k(b, c)) \tag{2.3}$$

where h and k can be a novel combination of functions that embed these attributes in the manifold of original distribution of X. Not tracking the relevant factors of variation typically leads to model memorizing only the surface level associations leading to mode collapse and lack of diversity in the generated outputs. On the other hand, explicitly caring about the

Table 2.1 Implicit Realization of information bottleneck in popular deep learning mechanisms

Architecture	**Manifestation of Information Bottleneck**	**Type of Bottleneck**
AlexNet	Pooling	Spatial
Attention	Activation	Temporal
VAE	Priors	Spatio temporal
Neural Module Networks	Softmax over Modules	Spatial
LISA	Linguistic Priors	Temporal
BERT	Masking	Spatio temporal
XLNet	Permutation	Spatio temporal

factors of variation can be seen as a way of incorporating inductive bias into the model and has the potential to avoid such pitfalls.

2.1.2 Implicit De-Entanglement in Seq2Seq Models: Deterministic Attention vs Stochastic Attention

I will illustrate this sub section with a typical generative model of speech: Text to Speech. Consider that we are interested in building a code mixed version: a model that can accomodate two languages in a single utterance. Let us also consider a speech corpus X consisting of languages $\{l_1, ..., l_n\}$, where each l_i might comprise of multiple speakers. Let y_1,...,y_n denote acoustic frames in the target sequence y while x_1,...,x_n denote the encoded text sequence x from one of the languages. A typical attention based encoder decoder network such as Tacotron(Y. Wang, Skerry-Ryan, Stanton, et al., 2017) factorizes the joint probability of acoustic frames as product of conditional probabilities. Mathematically, this can be shown as below:

$$P_\theta(y|x) = \Pi_{t=1}^{t=n} P(y_t|x_1...x_m, s_t) \tag{2.4}$$

where s_t is a decoder state summarizing y_1,...y_{t-1}. Parameters θ of the model are set by maximizing either the log likelihood of training examples or the divergence between predicted and true target distributions. At each time step t in these models, an attention variable a_t is used to denote which encoded state of $x_1...x_m$ aligns with y_t. The most common form of attention used is soft attention, a convex combination from encoded representation of input text. It has to be noted that soft attention in such scenarios is essentially a latent deterministic variable that computes an expectation over the alignment between input and output sequences. Empirically, soft attention provides surprisingly good alignment often correlating with human intuitions. Having said that, to synthesize speech from different languages at test time, the generative process needs to disentangle appropriate individual language attributes from observed data X_{obs} and also compose them to form a coherent utterance in the voice of desired speaker. However, presence of deterministic alignment method limits the ability of models to generalize to such scenario.

On the other hand, variational attention(Deng et al., 2018) provides a mechanism to factorize this alignment and mediate the generative process of y through a stochastic variable z. In addition, both soft and hard attention mechanisms can be shown as special cases of ELBO(Deng et al., 2018). Therefore, incorporating latent stochastic variables allows us to directly optimize ELBO. In this context, model parameters are set by maximizing the log marginal likelihood of the training samples. But direct maximization of this marginal in the presence of latent variable is often difficult due to expectation involved. To address this, a recognition network q is employed to approximate the posterior probability using reparameterization. It is interesting to note that the encoder in a deterministic Seq2Seq network functions as the recognition network in latent stochastic variable models and is incentivized to search over variational distributions to improve ELBO. Intuitively, the lower bound is tight when the inferred variational distribution is closer to the true posterior of

the data. This has a sense of grounding in our understanding of the task as well. Perhaps there are a set of universal phonemes, around 120, which should be enable us to speak in any language subject to the phonotactic constraints of the language. Having such prior information greatly reduces the model size as opposed to naively using a combination of all phones from all the languages to build a polyglot model.

2.1.3 Analysis of role of priors in Latent Stochastic Models

The choice of priors plays a significant role in optimization within latent stochastic models. In this subsection, we present an analysis to show that priors control the disentanglement of causal factors of variation in such models. Let us consider the ELBO being optimized in a VAE:

$$E_{q_\phi(z|x,c)}[logp_\theta(x|c,z)] - |D_{KL}(q_\phi(z|x,c)||p_\theta(z|c))| \tag{2.5}$$

where the first term is the reconstruction error while the second is the divergence between approximate and true posteriors. Here are the four phenomenon that are manifested due to choices of priors:

(1) *Disentanglement or Factorization of causal factors of variation*

The KL divergence forces the posterior distribution output by encoder to follow an appropriate prior about the data generation process. Typically, prior space is assumed to be continuous distribution and a unit Gaussian. The global optimum value for the divergence in such cases is 0 and is reached only when both the distributions exactly match each other. Since the prior information about the data generation process typically involves some causal factors of variation of the data, this naturally is assumed to translate to a constraint on the encoder to track such factors. Thus, such models have potential to disentangle or factorize the causal factors of variation in the distribution.

(2) *Marginalization of Nuisance Factors of Variation*

It has to be noted that during training optimization is performed in expectation over mini-batches. Therefore, the expectation of KL divergence can be rewritten as related to the amount of mutual information between the latent representation and the data distribution (Makhzani and Frey, 2017). As this divergence decreases, the amount of information the encoder can place in the latent space also decreases. As a result, encoder is forced to discard some nuisance factors that may not have contributed to the generation of data. Thus, KL divergence also forces the model to marginalize the nuisance variables.

(3) *Posterior Collapse due to simple priors*

Consider the scenario where the prior is too simplistic, such as the aforementioned unit normal distribution. In such cases, the model is incentivized to force the posterior distribution to closely follow the Gaussian distribution (X. Chen et al., 2016). Typically the decoders in variational models are implemented using universal approximators such as RNNs. In the context of a TTS systems, decoder segment of the acoustic model along with the neural vocoder act as the decoders. Since such decoders are very powerful, they are able to learn or ignore the priors about data distribution themselves and hence marginalize out the latent representation input from the encoder. In other words, the prediction of next sample is based solely on the marginal distribution at the current timestep which can be implemented by learning a dictionary per time step. Therefore, the encoder is no longer forced to track the causal factors of variation in the data. This is referred to as posterior collapse or mode collapse.

(4) *Loss of output fidelity due to complex priors*

A reasonable and intuitive solution to posterior collapse is making the prior space more complex thereby pressurizing the posterior distribution to track the prior space more closely. For instance, (Burgess et al., 2018b) attempt to accomplish this by adding a hyperparameter β to promote disentanglement and gradually increasing channel capacity, something that increases loss. However, it has to be noted that simply making the prior distribution arbitrarily complex also perhaps leads to unreasonable constraints on the decoder. For in-

stance, in scenarios that have categorical distribution as their output (tasks such as language modeling, machine translation, image captioning among others) it is unintuitive to assume that the true prior that generates latent distribution is a Gaussian when the likelihood is based on discrete sequential data in such tasks. Having such strong priors directly affects the reconstruction ability in these models.

Therefore, priors in latent stochastic models play a significant role in the optimization and facilitate disentanglement of causal factors of variation on the one hand, as well as help the ability of the model to reconstruct the data distribution on the other. Having this knowledge enables us to engineer various components to tune the model behavior as per our requirements.

Chapter Three

De-Entanglement of Content - Case study with Acoustic Unit Discovery

3.1 Problem Introduction

A major bottleneck in the progress of many data-intensive language processing tasks such as speech recognition and synthesis is scalability to new languages and domains. Building such technologies for unwritten or under-resourced languages is often not feasible due to lack of annotated data or other expensive resources. A fundamental resource required to build such a stack is a phonetic lexicon - something that translates acoustic input to textual representation. Having such a lexicon, even if noisy, can help bootstrap speech recognition models, synthesis, and other technologies. Typical approaches may involve a pivot language or bootstrapping or adapting from a closely related high-resource language. But, this can be a deceptively non-trivial task due to linguistic differences which can pose inherent difficulties. For instance, it may be unreasonable to analyze a Sino-Tibetan language using English as a source. Moreover, using an additional language might make the model learn unintended surface level associations or biases between the participating languages that prevent them from generalizing across languages. Associations between these languages over a set of units that may better generalize to other languages. Therefore, in this paper we are interested in

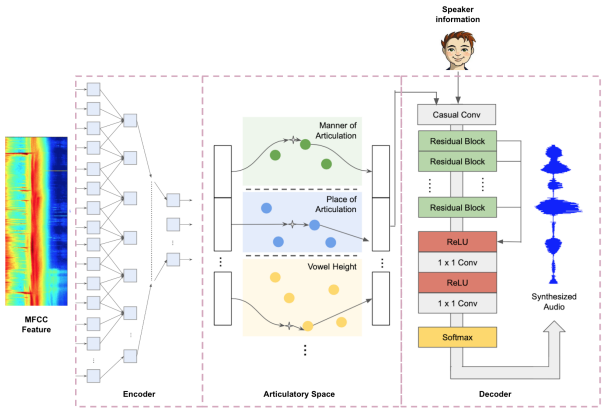


Figure 3.1 Illustration of our procedure for automatically discovering acoustic units from a speech utterance. We pass the speech utterance through a downsampling encoder. The encoded representation is hashed to a latent code based on a discrete articulatory prior bank. The code is passed to the decoder, a WaveNet using speaker embeddings as global conditioning that regenerates audio.

discovering the appropriate acoustic phonetic units.

In ZeroSpeech Challenge(Ewan et al., 2019) resynthesis is considered a good proxy task to evaluate the performance of systems when training using unsupervised approaches. To accomplish this we use neural generative models. Deep Neural Generative models have seen a tremendous amount of progress in the recent past. These models aim to model the joint probability of the data distribution and the conditioning information as a product of conditional distributions. Typical implementations of such models follow an autoregressive framework, although other formulations have been suggested as well. Such models have been shown very effective in addressing one of the major challenges with conventional vocoding techniques - fidelity. Neural generative models has been shown to generate speech that rivals natural speech when conditioned on predicted mel spectrum (Shen et al., 2017). Speech has a lot of natural variations in terms of content, speaker, channel information, speaking style, prosodic variations, etc. Accordingly, we are interested in models which have flexibility to marginalize such variations but preserve the phonetic content and distinguish meaningful differences between phonetic units.To accomplish this, we employ sequence to sequence models with latent random variables (referred to as latent stochastic models hereafter). These

models provide a mechanism to jointly train both the latent representations as well as the downstream inference network. They are expected to both discover and disentangle causal factors of variation present in the distribution of original data, so as to generalize at inference time. While training latent stochastic models, optimizing the exact log likelihood can be intractable. To address this, a recognition network is employed to approximate the posterior probability using reparameterization (Kingma and Welling, 2013a). When deployed in encoder-decoder models, this approach is often subject to an optimization challenge referred to as KL-collapse (Bowman et al., 2015), wherein the generator (usually an RNN) marginalizes the learnt latent representation. Typical approaches to dealing this issue involve annealing the KL divergence loss (Bowman et al., 2015; Zhou and Neubig, 2017), weakening the generator (T. Zhao, R. Zhao, and Eskenazi, 2017) and ensuring the recall using bag of words loss. In our work, we present an approach to deal with the KL-collapse problem by vector quantization in the latent space. Building on (Oord, Vinyals, et al., 2017a; Chorowski et al., 2019), we add additional constraints in the prior space forcing the latent representations to follow articulatory dimensions: The encoded representation is hashed to a latent code based on a articulatory prior bank designed using a discrete codebook. Our decoder is a conditional WaveNet using speaker embedding as global embedding trained to regenerate input audio using the code sequence as local information.

3.2 Background - Acoustic Unit Discovery

Let us consider a speech corpus X which consists of speakers $\{s_1, s_2..., s_n\}$. The goal of acoustic unit discovery is to come up with a set of units **U** that represent a speech utterance $x \subset$ X allowing robust resynthesis. The elements of such a set also might conform to desirable characteristics such as being injective, consistent and compact, i.e. that different inputs should have discriminant acoustic units, but expected variance such as speaker or dialect should produce the same acoustic units.

There have been numerous attempts to discover such acoustic units in an unsupervised fashion. In (Huijbregts, McLaren, and Van Leeuwen, 2011), authors presented an approach to modify the speaker diarization system to detect speaker-dependent acoustic units. (Jansen, Thomas, and Hermansky, 2013) proposed a GMM-based approach to discover speaker-independent subword units. However, their system requires a separate Spoken Term Detector. Recently, due to the surge of deep generative model, using unsupervised method such as auto-encoder and variational auto-encoder (VAE). (Badino, Canevari, et al., 2014) designed a stacked AutoEncoder using backpropagation and then cluster the representations at the bottleneck layer. To avoid quick transitions leading to repeated units, they employed a smoothing function based on transition probabilities of the individual states. (Ebbers et al., 2017) extended the structured VAE to incorporate the Hidden Markov Models as latent model. (Oord, Vinyals, et al., 2017a; Chorowski et al., 2019) proposed VQ-VAE and argue that by vector quantization the ““posterior collapse" problem could be circumvented.

3.3 *VACONDA*

3.3.1 Analysis of optimization and de-entanglement

WaveNet (**van2016WaveNet**) is an autoregressive neural model with a stack of 1D convolutional layers that is capable of directly generating audio signal. It has been shown to produce generated speech that rivals natural speech when conditioned on predicted mel spectrum (Shen et al., 2017). The input to WaveNet is subjected to corresponding gated activations while passing through each dilated convolutional layer and is classified by the final softmax layer into a μ law encoding. The concrete form of the residual gated activation function is given by following equation:

$$r_d(x) = tanh(W_f * x) \odot \sigma(W_g * x) \tag{3.1}$$

where x and $r_d(x)$ are the input and output with dilation d, respectively. The symbol $*$ is a convolution operator with dilation d and the symbol $\odot$ is an element-wise product operator. W represents a convolution weight. The subscripts f and g represent a filter and a gate, respectively. The joint probability of a waveform $\mathbf{X}$ can be written as:

$$P(X|\theta) = \prod_{t=1}^{T} P(x_t|x_1, x_2..x_{t-1}, \theta) \tag{3.2}$$

given model parameters θ. During implementation of WaveNet, the autoregressive process is realized by a stack of dilated convolutions. The final output y_t at time step t can be expressed mathematically as:

$$\hat{y_t} \sim \sum_{d=0}^{D} h_d * r_d(x) \tag{3.3}$$

where x, y represent input and output vectors; D is the number of different dilation used and d is the dilation factor; h_d is the convolution weights. This stack of convolutions is repeated multiple times in the original WaveNet. Optimization in WaveNet is performed based on the error between predicted sample and the ground truth sample conditioned on previous samples in the receptive field alongside the local conditioning. Expressing the loss function being optimized mathematically the error at sample t is:

$$l_t = Div(\hat{y_t}||y_t) \tag{3.4}$$

Here, we define the divergence similar to the (Salimans et al., 2017), To optimize this loss, the contribution from the individual convolution layers towards this global error function

must be nullified. Now let us consider the expression for intermediate output for a single filter in Eqn 3.3:

$$x_{out}(t) = \sum_{\tau=0}^{t} h(\tau)x(t-\tau) \tag{3.5}$$

where τ is the receptive field covered by the model and $h(\tau)$ represents the discrete state representation at time t. Without loss of generality and dropping the term τ for brevity, the spectral representation generated by the model can be expressed as:

$$Y(z) = H(z)X(z) \tag{3.6}$$

Considering the discrete nature of input from Eqn 3.4, an interpretation of Eqn 3.6 is that the neural autoregressive model acts as the transfer function and is discretized by convolving with the samples from original signal. It has to be noted that this is similar to the formulation of source filter model of speech, specifically the periodic components aka voiced sounds. Voiced sounds typically represented as impulse train are convolved with the transfer function to generate spectral envelope. As a corollary, from Eqn 3.4 and 3.6, we posit that the optimization in WaveNet model is performed by minimizing the divergence between true and approximate spectral envelope. Note that latent stochastic models such as VAEs are aimed to minimize the divergence between true and approximate posterior distributions of input data. The advantage with such models is the presence of stochastic random variables that capture the causal factors of variation in input based on some prior information about the distributional characteristics of data. Techniques aimed at this (Higgins et al., 2016) have shown that it is possible to effectively disentangle the factors of variation using stochastic variables. Hence, we postulate that it should be possible to augment WaveNet decoder with a suitable encoder and an appropriate prior distribution to disentangle the acoustic phonetic units from a given utterance.

However, this is a deceptively non-trivial task. If the prior is too simplistic, such as unit normal distribution, the model is trivially incentivized to force the posterior distribution to closely follow the Gaussian prior distribution (X. Chen et al., 2016), particularly early in training. This results in the decoder marginalizing out the latent variable completely, manifesting in poor reconstruction ability. On the other hand, making the prior distribution arbitrarily complex also leads to unreasonable constraints on the decoder. For instance, in scenarios that have categorical distributions as their output (tasks such as language modeling, machine translation, and image captioning among others) it is unintuitive to assume that the true prior that generates latent distribution is a Gaussian when the likelihood is based on discrete sequential data. We make an observation that dealing with speech presents a characteristic advantage - speech has both continuous as well as discrete priors. The generative process of speech assumes a Gaussian prior distribution which is continuous in nature. However, the language which is also present in the utterance can be approximated to be sampled from a discrete prior distribution. Exact manifestation of this in the linguistics can be at different levels: phonemes, words, syllables, subword units, etc. From the analysis presented in the previous section, we hypothesize that if we use background knowledge about the data distribution while designing the priors, we can help the encoder effectively disentangle the latent causal factors of variation in the data. In other words, this presents us with an opportunity to control what gets disentangled in the latent space by appropriately choosing a prior distribution. Therefore, we engineer our prior space to account for the phonetic information in the utterance by representing the prior as a discrete latent variable bank, similar to the filterbanks used for feature extraction from speech. Each discrete latent variable has a different set of states reflecting one of the articulatory dimensions. The specific design of our latent space is highlighted in Table 3.1.

Table 3.1 Articulatory Features

[] **Feature name**	**Value**	**Details**
[] vc	+ - 0	vowel or consonant
vlng	s l d a 0	vowel length
vheight	1 2 3 0 -	vowel height
vfront	1 2 3 0 -	vowel frontness
vrnd	+ - 0	lip rounding
ctype	s f a n l r 0	consonant type
cplace	l a p b d v g 0	place of articulation
cvox	+ - 0	consonant voicing
asp	+ - 0	consonant voicing
nuk	+ - 0	consonant voicing
[]		

3.4 Experiments

3.4.1 ZeroSpeech 2019 dataset

ZeroSpeech Challenge 2019: TTS without T is to propose to build a speech synthesizer without any text or phonetic labels (Sakti, Maia, et al., 2008; Sakti, Kelana, et al., 2008; Ewan et al., 2019). The systems are required to extract the symbolic representation of the raw audio, and then re-synthesize the audio using these discovered units. There are three datasets in total: (1) *Unit Discovery Dataset* provides audio from a variety of speakers and is used to unsupervised acoustic modeling, (2) *Voice Dataset* provides audio from the targeted speaker and is used for synthesizer modeling and (3) *Parallel Dataset* is intended for finetuning both the sub-systems. We have not utilized the parallel dataset for our observations in this study. The development language is English and the test language is Standard Indonesian. The system is constrained to not use any pre-existing resource or models. To

ensure that the model generalizes out of the box, the hyperparameter will be fine-tuned only on the development dataset, and the model will be trained in test language under the same parameters.

3.5 Baseline System

We have a three-stage pipeline: (1) *Unit Discovery*: We hypothesize acoustic units given a speech utterance using latent Stochastic Models; (2) *Unit Alignment*: We fine-tune the alignment between the utterance and the proposed acoustic units ; (3) *Unit Synthesis*: We build a speech synthesizer using the acoustic units and the target voice.

As proposed in (Sitaram, Palkar, et al., 2013), we take the initially discovered transcription of the acoustic units for our speech corpus and train an ASR model on it. Then we re-encode the corpus using the ASR model, and train a TTS system on it. Here we using Bi-LSTM with CTC loss as our ASR model, and tacotron (Jia et al., 2018) as TTS system.

3.5.1 VACONDA

The architecture of our model is built on top of VQ-VAE. It consists of three modules: an encoder, quantizer and a decoder. As our encoder, we use a dilated convolution stack of layers which downsamples the input audio by 64. The speech signal was power normalized and squashed to the range (-1,1) before feeding to the downsampling encoder. To make the training faster, we have used chunks of 2000 time steps. This means we get 31 timesteps at the output of the encoder. The quantizer acts as a bottleneck and performs vector quantization to generate the appropriate code from a parameterized codebook. We define the latent space $e \in R^{k \times d}$ to contain k d-dim continuous vector. Quantization is implemented using minimum distance in the embedding space. The number of classes was chosen to be 64, approximating 64 universal phonemes. We use a linear mapping to first project the 128 dimensional vector to 160 dimensions. We then perform comparisons with respect to individual articulatory

dimensions each of which is 16 in size. Assuming $z_e(x)$ denotes the encoder output in the latent space, then the input of decoder $z_d(x)$ will be obtained by ${}_jd(e_j, z_e(x))$, where d is a similarity function of two vectors. In this paper, we consider Euclidean distance as the similarity metric. Our decoder is an iterated dilated convolution-based WaveNet that uses a 256-level quantized raw signal as the input and the output from vector quantization module as the conditioning. Although using a Mixture of Logistics loss function might yield a better output, we have only used a 256 class softmax in this study. The decoder takes the output from the quantizer along with the speaker label as global conditioning and aims to reconstruct the input in an autoregressive fashion. Following IDCNNs, we have shared the parameters of all the stacks.

3.5.2 Analysis

In this section, we will discuss different design choices in the architecture, including input features and latent space constraints.

Acoustic Unit Discovery

Here we analyze the AUD performance of three different models in ZeroSpeech dataset as shown in Table 3.2. We only show the results in English since we don't have ground truth for the Indonesian language.

As in Table 3.2, the VACONDA achieves the best bit rate among three models. With such small number of unit, we could resynthesize and even convert the speech in a very high quality.

Speech Resynthesis and Conversion

The proposed model supports synthesizing the same speech in both the same speaker and a different speaker. Here we show a sample in the test dataset of Indonesian language in

Table 3.2 Performance of different systems in ZeroSpeech

[]	English	
Model	ABX score	bitrate
[] Baseline	**27.46**	74.5
Three-stage Model	34.86	68.54
VACONDA	38	**58.19**
[]		

Figure 3.2. When we feed the decoder with the same speaker identification, the decoder will generate the original audio. Otherwise, it will perform speech convertion. The three audio shares similar structure. However, the converted audio has denser waveform, suggesting it's a different speaker. For the sampled audio, please visit the our website.

3.5.3 Conclusion

In this case study, we present an approach to automatically discover acoustic-phonetic units from a speech utterance in an unsupervised fashion. We first present an analysis to show that incorporating latent random variables into neural generative models using suitable priors allows us to control what gets encoded into the latent space. Based on this, we employ articulatory features as a discrete prior bank in the latent space and obtain acoustic units that are speaker and language independent. To validate effectiveness of the discovered units, we perform discriminability tests as part of ZeroSpeech Challenge 2019.

3.6 De-Entanglement of Content - Application to Source Separation

Speech synthesis has taken some major strides in past few years especially in the form of text-2-speech synthesis (TTS) models. However, most of the work that has been carried out involves carefully recorded speech data. Generation of such vast amount of data for every application is a daunting task. On the other hand, there is a plethora of speech data that is available on the internet such as news broadcasts, press conferences, audio books etc - also referred to as *Found Data*. The only hindrance in utilizing such data for speech based machine learning models is that this found data is characterized by noise or music in the background. Presence of noise / music degrades the performance of such models. One of the solutions to this problem is source separation - separating out speech from music in the audio. There have been several attempts to accomplish this task using both classical speech processing techniques as well as deep learning models.

6287816 proposed a matrix factorization of the magnitude spectrogram of audio that utilizes the periodicity in music and sparseness in speech to separate the two. However, this technique requires a lot of hyperparameter tuning depending on the type of background music and also degrades the quality of separated speech to some extent. REPET **6269059** also involves music separation by exploiting its periodic nature but on occasions still leaves a residual music in the background. Most of the work in source separation using deep learning has been supervised **SVSGAN Disc TFGAN spen** i.e. they had both noisy and clean versions of the data. However most of the times, especially with found data, we don't have the clean version of the data.

There has also been some focus on source separation using unsupervised models. **Hsu2018Disentangli** takes the approach of data augmentation by adding different background noise to the clean

data and then training an adversarial classifier to make these augmented versions of data indistinguishable from the original speech. However, this method again requires a clean version of data first and additional data augmentation that is representative of the noise in the background. Therefore essentially, this is a semi-supervised approach that requires labels for clean and noisy data. One other semi-supervised approach is using domain adaptation **domadp** where output is made to follow the clean data domain while making the encoding for clean and noisy data domain indistinguishable using a adversarial classifier. However, this approach requires speech content in both clean and noisy version of data to be very similar for domain adaptation to occur.

We propose a completely unsupervised approach using multinode variational autoencoders (VAE) combined with robust principal component analysis (RPCA) **6287816** as a post-processing step. Our goal is to enable the use of found data for downstream TTS applications. Therefore, the data we target is predominantly speech with music in the background. We apply this approach on two datasets:- Wilderness**wildernessdataset** and Hub4. Wilderness consists of Bible recordings in 699 languages while Hub4 is a news broadcast dataset in English. Both of these datasets contain music/noise in the background. We show that the proposed approach separates out the dominant mode, speech, from a noisy audio and improves the performance of the downstream tasks irrespective of the language of the speech.

This paper is organized as follows: section 1 discusses the variational autoencoder framework, section 2 talks about source separation using VAE, section 3 addresses the extension to multinode VAE architeture and section 4 discusses post-processing using robust principal component analysis (RPCA). Section 5 analyses the source separation capacity and architectural requirements of the proposed model. Section 6 reports the performance of the proposed model for source separation and for downstream TTS applications. We conclude in section 7.

3.7 Variational Autoencoder

Variational autoencoder model in this paper follows the standard formulation consisting of an inference network with a speech encoder $p(z|x)$ and a latent space decoder $p(x|z)$, where x and z represent the input and the latent space random variables respectively.

The figure 3.3 depicts the latent variable model for variational autoencoder. The true posterior density is intractable.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{3.7}$$

We then approximate the true posterior $p(z|x)$ with a variational distribution $q(z|x)$ that has a prior $p(z)$. The objective can be represented by the evidence lower bound (ELBO) or variational lower bound on the likelihood of the data.

$$\log p(x) \geq \mathcal{L}(x) \tag{3.8}$$

$$\mathcal{L}(x) = E_q[\log(p(x|z))] - D_{KL}(q(z|x)||p(z)) \tag{3.9}$$

where $\mathcal{L}(x)$ denotes the variational lower bound on the likelihood of the data and D_{KL} is the Kullback-Leibler divergence. We write the first term as a mean squared error (MSE) between the reconstructed and the original data and the prior p(z) follows a standard normal distribution $\mathcal{N}(0, I)$.

3.8 VAE for Source Separation

As shown in the previous section, a variational autoencoder reconstructs the input data conditioned on the latent space. The latent space is constrained to follow a certain prior distribution, such as Gaussian distribution. **JMLR:v19:17-704** shows that this formulation is equivalent to minimizing the alternative lower bound function

$$\begin{aligned} &minimize \;\; n \cdot rank[\, L\,] + \parallel S \parallel_0 \\ &M = L + S \end{aligned} \tag{3.10}$$

where M is the original data matrix, L is a low-rank matrix and S is a sparse matrix and $\parallel . \parallel_0$ denotes the l_0 norm. This is shown to be equivalent to an RPCA problem if an optimum solution exists otherwise it's known to smooth out undesirable erratic peaks from the energy curve. **JMLR:v19:17-704** also presents some interesting results on VAE and it's separation properties. We rewrite some of the results what it means in our context here.

- This formulation of variational autoencoders is shown to perform robust outlier removal in the context of learning inlier points constrained to a manifold of unknown dimension. In simple terms this means, VAE has the property to remove sparse components in the input data distribution and accordingly reduces the latent space to a required (unknown) dimension.
- VAE also help smooth out undesirable minima from the energy landscape of the optimization problem which differentiates it from traditional deterministic autoencoders.

Since our goal is to enhance speech synthesis and speech recognition performances on the 'found' data, we target audio data that is predominantly speech with some music (almost uniform) in the background, for instance, news broadcasts and audio books. We will later show that the presence of the background music can effect the speech synthesis performance drastically. It's been shown that speech and music distributions in audio are quite distinct **SM_diff SM_diff1** As a result VAE has the tendency to remove the sparse outlier - music from the audio.

In case of multiple speakers in the input audio there can be multiple modes in the speech distribution as well. This can be solved by have multiple nodes in the latent space. This is possible because all latent variables are initialized at random and pick multiple speech

modes from the distribution. Later in the results section, we are going to analyze this and the requirement on the number of nodes in the latent space depending on the input data distribution. We also talk about how the performance of the output speech changes based on the intensity/loudness of the music in the background.

3.9 Multi-node VAE Model Architecture

The Figure 3.4 depicts the multi-node variational autoencoder architecture. It consists of an Bi-LSTM encoder ($LSTM_E$) for the inference network that captures latent space distributions $p(z_1|x), p(z_2|x), \ldots, p(z_k|x)$ where x is the magnitude spectrogram of the input audio, $z_1, z_2, \ldots, z_k$ are the latent variables and k is the number of latent variables. The reconstruction network is a Bi-LSTM decoder ($LSTM_D$) which generates the reconstructed input distribution at each time-step $p(x_r^t|x_r^{t-1}, z_1, z_2, \ldots, z_k)$ conditioned on the reconstructed input from the previous time-step and the latent space.

$$h_e^t, c_e^t = LSTM_E(x^t, h_e^{t-1}, c_e^{t-1}) \tag{3.11}$$

$$\mu_i^t = MLP_{\mu_i}(h_e^t)\ \forall i \in 0, \ldots, k \tag{3.12}$$

$$logvar_i^t = MLP_{\sigma_i}(h_e^t)\ \forall i \in 0, \ldots, k \tag{3.13}$$

$$q(z_i|x) = \mathcal{N}(\mu_i, \exp(logvar_i)) \tag{3.14}$$

$$h_d^t, c_d^t = LSTM_D(\phi^t, z_1^t, \ldots, z_k^t, h_d^{t-1}, c_d^{t-1}) \tag{3.15}$$

$$\phi^t = MLP_{\phi}(h_d^{t-1}) \tag{3.16}$$

$$x_r^t = MLP_{x_r}(h_d^t) \tag{3.17}$$

where Bi-LSTM refers to bidirectional long short term memory recurrent neural network, MLP refers to a multi-layer perceptron network, h_e, c_e represent hidden and cell states of the encoder LSTM, h_d, c_d represent hidden and cell states of the decoder LSTM and ϕ represents

the context from the previous time-step of the decoder. The initial hidden state and the cell state of the decoder LSTM are learnable parameters. The latent variable model for the multinode variational autoencoder is shown in figure 3.5

The modified learning objective for a multinode VAE can be represented as an extension of equation 3.9 as:

$$\begin{aligned} \mathcal{L}(x) \geq E_q[\log(p(x|z_1, z_2, \ldots, z_k))] \\ - \sum_{i=1}^{k} D_{KL}(q(z_i|x)||p(z_i)) \end{aligned} \tag{3.18}$$

3.10 Speech Enhancement

The output of the VAE network from the above formulation removes the music from the audio however, replaces the music content with random noise instead of silence. There can be multiple post processing or speech enhancement techniques used to eliminate this residual noise such as speech enhancement neural networks or classical speech processing methods. Here in this paper we use robust principal component analysis (RPCA) **6287816** to eliminate the background noise as it gives control over the quality of speech versus the amount of background noise. We follow the original formulation from the paper by expressing the speech separation as a matrix factorization problem. It represents the magnitude spectrogram of the audio signal as a sum of low rank matrix and a sparse matrix. The assumption here is that non-speech component (background noise) is low rank while the speech component is sparse.

$$\begin{aligned} minimize \parallel L \parallel_* + \lambda \parallel S \parallel_1 \\ M = L + S \end{aligned} \tag{3.19}$$

where $M \in R^{n_1 \times n_2}$ is the magnitude spectrogram of the VAE output, $L \in R^{n_1 \times n_2}$ is

a low rank matrix, $S \in R^{n_1 \times n_2}$ is a sparse matrix, $\| \ . \ \|_*$ is the nuclear norm and $\| \ . \ \|_1$ is the L_1 norm. $\lambda > 0$ is a hyperparameter that controls the rank and sparsity of L and S respectively. It is recommended in **6287816** to use $\lambda = 1/\sqrt{max(n_1, n_2)}$ to obtain the best result. However, we only need to enhance the audio a little while retaining the speech quality so we use $\lambda = 0.3/\sqrt{max(n_1, n_2)}$. Instead of the hard mask in **6287816** we used a soft mask as it resulted in a better quality and a smoother speech. The idea is to have a high value for the speech mask where the magnitude of the speech component is much greater the magnitude of the non-speech component.

$$|S| > g|L| \tag{3.20}$$

$$|S|^2 > g^2|L|^2 \tag{3.21}$$

$$|M|^2 = |S|^2 + |L|^2 \tag{3.22}$$

$$|S|^2 > g^2|M|^2 - g^2|S|^2 \tag{3.23}$$

$$|S|^2 > \frac{g^2}{1+g^2}|M|^2 \tag{3.24}$$

$$|S| > \sqrt{\frac{g^2}{1+g^2}}|M| \tag{3.25}$$

$$\frac{|S|}{|M|} - \sqrt{\frac{g^2}{1+g^2}} > 0 \tag{3.26}$$

where $g \geq 0$ is the gain factor. We came up with a Sigmoid looking threshold for the mask which is still close to the hard mask but results in smoother speech transitions.

$$W = \frac{1}{1 + \exp\left(-\alpha\left(\frac{|S|}{|M|} - \sqrt{\frac{g^2}{1+g^2}}\right)\right)} \tag{3.27}$$

where $W \in R^{n_1 \times n_2}$ represents the speech mask and the obtained speech spectrogram is

$$X_{speech}(i,j) = W(i,j)M(i,j)\ \forall i,j \tag{3.28}$$

3.11 Experiments

We applied the multinode VAE model on two datasets:- Wilderness and Hub4. Wilderness dataset consists of Bible recordings in 699 languages with music in the background. We carried out full experiments on two languages:- Dhopadhola (an African language) and Marathi (an Indian language). The results presented here are based on the model that was trained on languages different than the ones that are reported/tested. Hub4 consists of news broadcast recordings in English with various forms of noise in the background, such as music, clapping, roaring etc. We used about 2 hrs of training data for both datasets consisting 1 hr of speech only data and 1 hr of speech-music data.

For these experiments, the VAE model consists of input magnitude spectrogram of dimension 512, Bi-LSTM encoder and decoder with hidden size of 512, each of the fully connected layers for latent variables and decoder context from the previous time-step of dimension 64 and the final output layer with a dimension same as the input dimension. We trained for 50 epochs with annealing weight for KL-Divergence loss, this is explained in detail below. For both datasets, we used an ADAM optimizer with a learning rate of 1e-3.

The input data distributions for the two dataset are shown in Figure 3.8. These 2-dimensional distributions are obtained after applying PCA to the magnitude spectrogram

and plotting the histogram of the first two components. This figure shows the high density regions of the two distributions. As we can observe, the wilderness distribution has one significant high density region while the hub4 distribution consists of multiple high density regions. Hence, Hub4 data will have more dominant speech modes than the wilderness data. This is probably because there are multiple speakers in Hub4 as well as news broadcast speech has more variance as compared to bible recordings. This gives us an approximate idea that Hub4 multinode VAE model will require more nodes in the latent space than the model for Wilderness dataset.

Figure 3.9 shows the likelihood of fitting Gaussian Mixture Models as a function of the number of cluster centers. As discussed earlier, speech modes are dominant in the target data so fitting n clusters in the curve can be thought of as having $n-1$ nodes/clusters in the VAE for the speech and 1 cluster as the residual non-speech data. The multinode VAE model for the wilderness data obtained good results with just 1 VAE node or 2 clusters as can be confirmed from the graph where likelihood values are high for just 2 clusters. On the other hand, multinode VAE model for Hub4 gave good results with 3 nodes or 4 clusters. Now, as we increase the number of nodes, the peak performance does not change much, however, we attain the same peak performance for more model states:- MSE loss vs KL loss. This will be explained using training loss curves. It would have been better to use some validation parameter but since model performance for human hearing can only be analyzed by listening to the speech, we use the training metrics.

The Figure 3.12 and 3.16 give an idea of speech separation capacity of the model as we increase the number of latent variables. During training of the multinode VAE model, we anneal the KL divergence loss for latent space exponentially. We do the annealing for latent variables simultaneously. Initially, KL divergence loss is assigned a very small weight and then increased exponentially. So, it first increases (not shown in the plot as it's out of

the range of the plot) and then decreases eventually while the MSE reconstruction loss first decreases and then increases slightly. During this process there is a small window where both the losses are low enough and we are able to extract out speech from the audio. This window is determined by the threshold values for both losses. If both loss values are below their respective thresholds, we observe speech at the output.

The blue and orange shaded regions in figures 3.12 and 3.16 depict the loss values below the threshold for MSE loss and KL divergence loss respectively. Therefore, their intersection as indicated by the overlap region represents the model parameters that result in speech and music separation. For visualization, the MSE loss in the figure is averaged over all the samples as well as in the time dimension of the audio while the KL divergence loss is averaged over all the latent variables as well as across all samples. Using these loss definitions results in a threshold value of 250 for MSE loss and a threshold value of 60 for KL divergence loss for both datasets. These are just soft experimental values and may change for other datasets as well as a different loss definition. The key idea is that there exists a window where speech separation occurs.

As shown in the figure, for Wilderness data we obtain this window with just one node in the latent space. As we increase the number of nodes in the latent space, we don't see any significant improvement in the quality of the output speech, however, we do obtain a wider window where this separation occurs. As for the Hub4 dataset, we don't observe any such window with one latent node, however, we do obtain a separation window with three latent nodes and an even wider window with eight latent nodes. Observe that, these results align with the results derived using input data distribution and GMM fitting analysis. Therefore, to be totally certain about the existence of a separation window, we can always add a few more latent variables than what we obtain from our analysis of input distribution.

Let's explore the nature of the output on either side of the separation window. On the blue/left side of the window MSE loss is very low while the KL divergence loss is high, this results in the output that is close to the original input that consists of both speech and music. On the orange/right side of the window, MSE loss is high while the KL divergence loss is low. This causes the network output to be a really noisy version of the speech component of the audio.

We also observed that, as the intensity/loudness of music in the background increases in an audio or for a part of the audio, the speech separation performance for that part of the audio begins to deteriorate slightly. For example, in case of advertisement segments between news broadcasts where music tends to dominate the segment. In such cases, we can still hear some traces of music in the background when we use the same VAE model as we do for the rest of the data. However, this is not a concern as the downstream applications we target don't generally depend on data with such high intensity music.

As mentioned in **JMLR:v19:17-704** a traditional autoencoder is not able to perform the outlier removal. We verified this fact experimentally. We removed any constraint on the latent space and trained the model to minimize the reconstruction loss. We observed that the model was able to reconstruct the audio completely including both speech and music. Therefore, a traditional autoencoder is not able to smooth out the energy contour and fails to remove any outliers. We also tried to experiment with this model on songs and movie clips. As the background music in songs and movies very dense and varies significantly, we observed that our model wasn't able to separate out speech completely. The output speech contained some music in the background and the quality of speech itself was compromised.

3.12 Results

The separated speech samples for both Wilderness and Hub4 datasets can be found at [1] under folders "Wilderness" and "Hub4" respectively. We present Wilderness samples for two languages - Dhopadhola and Marathi both having different but somewhat uniform music in the background. These samples can be found inside their respective folders within the wilderness folder. We also present samples for when we removed the KL-divergence term from the loss function and trained an autoencoder instead. It can be observed from the samples that the model reconstructs the whole audio without outlier/music removal. Hub4 data samples have a lot more variation in both speech and music. All the samples come from news broadcast in English. We also ran experiments on mixed signals where a background music was added to a clean sample from the wilderness data. We performed this experiment with drums, flute, guitar and piano music in the background. The mixed audio and the separated speech samples can be found at [1].

Input Audio	**SegSNR**	**MOS**
Noisy	4.91	1.5
Clean	7.5	1.2

Table 3.3 Segmented Signal to Noise Ratio (SegSNR) and Mean Opinion Score (MOS) for Mixed Signals

One other way to asses the proficiency of the proposed method is to evaluate its performance on downstream tasks, for instance, Text-2-Speech Synthesis (TTS). We performed Text-2-Speech synthesis on original and cleaned version of Marathi language from the wilderness dataset. The TTS samples can be found at [1] as well. We observe that Text-2-Speech synthesis performance improves significantly after music is removed from the background.

[1] https://github.com/nishantgurunath/Separabl/tree/master/samples

Audio	WER Mono Voxforge	WER Tri2b Librispeech
Noisy	100.53	103.72
Clean	102.84	104.33

Table 3.4 Word Error Rate (WER) for Noisy and Cleaned Audio on Pretrained Kaldi ASR Models

In TTS samples generated from the original (noisy) version of the audio, one can clearly see the distortion in the speech and presence of music in the background. Whereas, TTS samples that were generated from cleaned samples had a very clear speech quality with no music in the background.

3.13 Conclusion

We show that Multinode VAE model helps to remove the background noise/music in the 'found data' irrespective of the language of the speech. Extensive studies on different type of speech and music data verify the effectiveness and robustness of the proposed approach. Performance of this model on Text-2-Speech synthesis applications shows the potential of such an approach that can be further extended to other speech based machine learning models such as Automatic Speech Recognition (ASR). Such an efficient source separation technique can help overcome a major cause for under utilization of 'found data'. This could mean that acoustic based machine learning models can be drastically improved by leveraging the data found on the internet. Since this approach works in a unsupervised fashion, it eliminates the need to obtain labeled data which has been major hindrance to effective utilization of 'found data'. Since 'found data' is abundant, this could also possibly further accelerate the research in this area.

3.14 Identification of Paralinguistic Styles - A Case study

Applications of Computational Paralinguistics have grown rapidly over the last decade and span both human-human as well as human-machine interactions. The ComPare Paralinguistics challenges have been playing a significant role in driving progress in the diverse use of paralinguistics. Besides the traditional task of affect recognition using suprasegmental non-verbal aspects of speech, novel tasks were introduced, such as, the detection of speaker traits, deception, conflict, eating and autism Schuller, Steidl, Batliner, Vinciarelli, et al., 2013; Schuller, Steidl, Batliner, Burkhardt, et al., 2010; Schuller, Steidl, Batliner, Hantke, et al., 2015; Schuller et al., 2017. These challenges have shown that paralinguistic information can be used not only to identify affect but also clues that are helpful to detect abnormalities indicating disorders. Paralinguistic information also has applications in other domains of speech processing such as dialog systems, speech synthesis, voice conversion, assistance systems, and eHealth systems.

In this paper, we present our approach to three of the INTERSPEECH 2018 ComPare sub-challenges Schuller, Steidl, Batliner, Marschik, et al., 2018: prediction of 1) self-assessed affect, 2) atypical affect and 3) types of crying. The *Self-Assessed Affect **(S)** Sub-Challenge* and the *Atypical Affect **(A)** Sub-Challenge* aim to classify affect from speech. In **(S)** ground-truth labels are provided by the speaker itself. The prediction of affect from speech oriented by the own assessment, could be used as a support in eHealth systems for individuals with affective disorders, such that a therapist can monitor the emotional state of their clients. In **(A)** the goal is to determine the affect of mentally, neurologically, and/or physically disabled individuals. The challenge is that some disorders also affect way people express their emotions. However, having a system able to detect distress in workplaces of disabled individuals can be helpful to make supervisors aware to suggest breaks or divide tasks in smaller ones, improving the emotional state of workers and therefore their concentration. The *Crying **(C)** Sub-Challenge* focuses on using paralinguistic information to identify affect

in vocalisations of infants. Experts in the field of early speech-language development labeled audio-video clips into three classes of vocalisations: neutral/positive , fussing, and crying.

Par3: What are the baseline models provided? what do they use? The features used in the baselines presented by the organizers of the ComParE Challenge 2018 are composed by a standard set of segmental, suprasegmental features and also utterance level acoustic representations. Typical approaches for classification and prediction of paralinguistic features include extraction of low level descriptive features followed by a machine learning model. Examples of low level descriptors are Mel-Frequency Cepstral Coefficients (MFCCs), log Mel-scale filter banks energies (FBANK) and several suprasegmental acoustic features that can be extracted using the openSMILE tool Eyben, Wöllmer, and Schuller, 2010. These features act as general purpose feature set and are expected to achieve competitive results in a wide range of paralinguistic problems. However, derived neural representations using unsupervised learning have shown impressive results on many speech and image based tasks recently Aytar, Vondrick, and Torralba, 2016. These features usually embed the task relevant information from the entire utterance in a compact form. Also end-to-end learning models have been employed in affect classification using Long Short-Term Memories (LSTMs) or Gated Recurrent Units (GRUs) Trigeorgis et al., 2016; Schuller, Steidl, Batliner, Marschik, et al., 2018.

Motivated by this, we explore different utterance level representations and end-to-end approaches in the context of sub-challenges. Specifically, we investigate the significance of using both utterance level acoustic and derived linguistic features. We further employ data augmentation using utterance emphasis (see section 3.15.3) and random utterance segmentation (section 3.15.3), as a strategy to cope with class imbalance. For obtaining linguistic features we first obtain the text for each of the utterances using a pretrained English ASPire model. We then train a Recurrent Neural Network language model on the obtained text at the phone level and use the representation at the hidden state as the embedding of the utterance. Apart from this, we explore the applications of various Convolutional Neural

Network models and chart their performance. It has to be noted that even though acoustic and phonetic embeddings use identical inputs, they differ in the higher level features learned internally. Therefore we believe that they complement each other producing a superior fusion result.

3.15 Framework

In this section, we present different features and classifiers used for all three sub-challenges. We used two different classification models: 1) Bidirectional LSTM using low-level features which uses temporal information , and 2) Random Forest classifier or SVM Classifier using high-level features, which are utterance based, combined with utterance level embeddings.

3.15.1 Temporal classification

Low level features

For acoustic feature extraction we divided each utterance (length is 8) into 25 segments with a 10 frame shift. For each frame we extract 13 mel-frequency cepstral coefficients and their deltas and double-deltas obtaining a feature vector of 39 dimensions. We further extract the log pitch ($f0$) and strengths of excitation (5 dim) **yoshimura2001mixed** In addition, we also obtain 40 dimensional filter banks and 23 dimensional PLP based features. Filter banks have been obtained using the open source toolkit Kaldi Povey et al., 2011 with 'dithering' enabled as it was shown to be robust in other experiments. We have also extracted several features using Opensmile toolkit Eyben, Wöllmer, and Schuller, 2010 and performed singular value decomposition with the intention of obtaining an acoustic representation. This procedure also results in a dense low dimensional representation. This representation was later used in combination with the high level features we obtained in the spirit of early fusion.

Classifier

Using all previously mentioned features, we train a 2 layer bidirectional LSTM network with 512 units in each cell. This is followed by 2 fully connected layers each with 512 units. The final softmax layer dimensions were dependent on the sub challenge. The network is trained by minimizing the expected divergence between the classes using cylindrical SGD Smith, 2017.

3.15.2 Utterance-based classification

Recently, end-to-end approaches have shown impressive results on many speech based tasks Trigeorgis et al., 2016. Specifically combinations of CNN and fully connected layers with a global pooling layer have obtained human level recognition rates on speaker and language recognition tasks. The global pooling layer functions as averaging sequential inputs therefore aggregating frame level representations to utterance level. This is advantageous for end-to-end learning.

Extracting high level acoustic representations using Modified SoundNet

SoundNet Aytar, Vondrick, and Torralba, 2016 is a convolutional network operates on raw waveforms and is trained to predict the objects and scenes in video streams at certain points. After the network is trained, the activations of its intermediate layers can be considered a representation of the audio suitable for classification. It has to be noted that SoundNet is a fully convolutional network, in which the frame rate decreases with each layer. Since we need to predict the emotions with reasonable recall, we cannot extract features from the higher layers of SoundNet directly.

The original SoundNet network has seven hidden convolutional layers interspersed with maxpooling layers. Each convolutional layer essentially doubles the number of feature maps and halves the frame rate. The network is trained to minimize the KL divergence from

the ground truth distributions to the predicted distributions. In the original SoundNet architecture, the higher layers have been subsampled too much to be used directly for feature extraction. In order to fully exploit the information in the higher layers, we train a fully connected variant of SoundNet (see Fig. 3.20). Instead of using convolutional layers all the way up, we switch to fully connected layers after the 5th layer. We have also changed the input sampling rate to 16 to match the provided data.

Linguistic features

An informal analysis of the recordings indicated that the content being spoken plays a non trivial role in the valence of the utterance. A simple manifestation of this is the distribution of filled pauses and hesitations in the provided data across the classes. In the Self Assessed Affect dataset, examples belonging to the class 'low' have higher number of such irregularities compared to the other two classes. Note that these features are not extracted for the Crying dataset. Therefore we hypothesize that using an off the shelf phoneme decoder to recognize such events might be beneficial. For this we first obtain the text at the phoneme level for each of the utterances using a pretrained English ASPire model from the toolbox Kaldi Povey et al., 2011. We then train a Recurrent Neural Network language model on the obtained text at the phoneme level and use the representation at the hidden state as the embedding of the utterance. The architecture is depicted in Fig. 3.18.

Classifier

We obtain the prediction scores from our models using either a Random Forest Classifier or a one-vs-rest classifier implemented using a binary SVM classifier depending on the performance. It is a known fact that SVM models perform better on sparse data than does trees in general. Therefore depending on the data augmentation techniques, we choose the classifier.

3.15.3 Data Manipulation & Enhancement

In this section we present various data engineering approaches that make the data more suitable for our models. Specifically, we explore approaches that aim to (a) obliterate the imbalance in class, (b) extract derived features which might help in distinguishing between the classes, (c) downsizing and normalizing on the duration of clips, etc.

Class balancing by data restriction

In order to address the class imbalance present in the original data, we reduce the number of samples used for the classes that are dominant in the dataset. We hypothesize that the skewness of the original data causes low recall for classes that are in minority. Therefore, we study the effects of attempting to artificially balancing the classes by using less samples of dominant classes.

Class balancing by data augmentation

The objective function we minimize in this approach is the expected divergence between the classes. An analysis of the original data points to the imbalance between the classes: For example, in Self Assessed Affect subchallenge, there are almost 3 times less number of examples for the 'low' class compared to the other classes in the training set. To alleviate this, we look at approaches to augment the existing data. Since our model operates on the sequence of frames, we hypothesize that segmenting the audio data into chunks Agrima et al., 2017 exposes the model to different distributional properties. We obtain 4 times the original data for the class with less number of examples in Self Assessed Affect challenge by chopping the original signal between (0-2), (0-4), (0-6) and (0-8) seconds.

Deriving Speaker Identity

Speaker normalization and adaptation have been widely documented as significant for a speech recognition system. As the original data did not have speakers tagged per utterance, we have tried to do speaker recognition using length normalized i Vector. i-Vectors are low-dimensional representation of GMM supervectors in a single subspace which have been formulated to include all characteristics of speaker and inter-session variability. Mathematically, given an observation set X_s, the adapted mean super-vector m_s is modeled as,

$$m_s = m_0 + \mathbf{T}w_s + \theta \tag{3.29}$$

where m_0 is the Universal Background Model (UBM) supervector, and θ is the residual term which accounts for the variability not captured by T. Following Garcia-Romero and Epsy-Wilson Garcia-Romero and Espy-Wilson, 2011, we perform a within class covariance normalization followed by length normalization of i vectors. These have been shown to 'gaussianize' the distribution and improve the performance of PLDA. iVectors have been extracted after log energy based voice activity detection on the utterances. This system was built within framework of Kaldi toolkitPovey et al., 2011.

Improving contrastiveness of features

We have tried to improve the contrastive nature among the classes artificially. An informal analysis of the recordings from Self assessed affect subchallenge led to the observation that the utterances with high valence were also relatively at a higher speed compared to the utterances with lower rate. Therefore we increased the rate of speech for the high valence utterances by 10 percent while simultaneously decreasing the rate of speech for low valence utterances by 10 percent. We performed similar perturbations with respect to pitch: boosting the pitch of the samples from 'high' class and lowering the pitch for the samples from 'low'. The samples for 'medium' class have not been subjected to any modification.

3.15.4 Early Fusion - Combining different representations

We have experimented with a feature level fusion of Soundnet layer 5 and ResNet50 He et al., 2015 features extracted from the audio files. Resnet has been trained on around 1.28 images from the Imagenet dataset and has a top 5 error of 3.57% beating all other CNN image classifiers. We aim to systematically study the strategies of combining representations from multiple feature extractors.

Normalizing length of audio files

The audio files in the Atypical Affect Assessment have variational lengths. To bring all audio files to the same scale of duration, we clip all the audio files to a maximum of 3s which is the average length of duration of all audio files. Though this leads to some information loss, it could probably help in enhancing the presence of weaker class files.

3.16 Datasets

3.16.1 Self-Assessed Affect Recognition

The dataset used in this sub-challenge is the Ulm State-of-Mind in Speech (USoMS). It contains recordings of 100 students. The labels were obtained from the subjects themselves obtaining 3 classes: low, medium, and high.The class distributionfor combined train and dev sets are: 716 high, 698 medium, and 174 low. This highlights skewness in the data distribution.

3.16.2 Atypical Affect Recognition

The dataset comprised of a total of 10677 audio files out of which there are 3342 training, 4186 validation files and the remaining test files. There are four target classes that pertain to the four emotions - neutral, happy, sad and angry. The distribution of classes is again

skewed with 5209, 1708, 516 and 175 being the total numbers of neutral, happy, sad and angry labels on the train and validation sets.

3.16.3 CRYING

This dataset is obtained from the Cry Recognition In Early Development (CRIED) database. It consists of 5588 vocalizations of 20 infants sampled at 44.1kHz in mpeg format. The objective is to identify three mood-related types of infant vocalization - neutral/positive, fussing and crying. The class distribution is as follows: 2292 cases of neutral/positive mood, 368 files of class fussing and the remaining 178 belonging to the class crying. The dataset is clean of vegetative sounds such as breathing sounds, smacking sounds, hiccups and so on. Further details about the datasets can be obtained from Schuller et al., 2018.

3.17 Experiments

In the following we present the preliminary results obtained using the systems we investigated on the Self Assessed Affect sub challenge. We further present the results of UAR for blind tests for all the three sub-challenges.

3.17.1 Class balancing by data restriction(System CBR)

We systematically try to reduce the data points from the classes with higher number of examples. The results from this experiment are depicted in Table 3.6.

3.17.2 Speaker identity based experiments(System SI)

Since the classifiers we use are discriminative in nature, we experiment with two ways of incorporating speakers or subject specific information:

Table 3.5 UAR for data filtering experiments

Data split			UAR[]
1-4 6*100% Low	3*100% High	90% Medium	56.8
		70% Medium	55.0
		40% Medium	52.1
2-4	3*100% Medium	90% High	**[detect-weight]59.1**
		70% High	56.8
		40% High	51.5
1-4	All Data	57.2	

- (1) We add the identity of the speaker as an extra dimension thus forcing the model to build speaker specific models. For example, in case of decision trees, this forces the model to split at the identity of speaker.
- (2) Normalizing with respect to the speaker, following the procedure typically used in speech recognition.

The results from these experiments have been depicted in table 3.13.

3.17.3 Improving contrastiveness of features(System CTR)

We have explored two ways of artificially increasing the contrastiveness of the features, based on observations on the original data. Since the different classes appear to have a different distribution of artifacts such as hesitation, we have tried to use signal processing techniques to further separate the classes. Specifically, we have used festival toolkit A. Black et al., 1998 to decompose the signal into its spectrum, pitch and then apply class specific modifications to the utterances in the train set. The waveform was reconstructed using the vocoding framework within festvox voice building tools. We have used WSOLA Verhelst

Table 3.6 UAR for class balancing by data restriction

Data split			UAR[]
1-4	6*100% Low	3*100% High	90% Medium	56.8
			70% Medium	55.0
			40% Medium	52.1
2-4		3*100% Medium	90% High	**[detect-weight]59.1**
			70% High	56.8
			40% High	51.5
1-4		All Data	57.2	

Table 3.7 UAR for Speaker identity based experiments

		Normalization	
3-4	UAR[]	used	not used
2*Speaker ID	used	62.2	54.0
	not used	61.1	**[detect-weight]64.7**

and Roelands, 1993 to accomplish duration based manipulations. The results from these experiments are shown in the table 3.8.

3.17.4 Blind Test Results and Discussion

The evaluation results on blind test set for the three sub-challenges is mentioned in the table 3.15. Based on the preliminary experiments, system **SI** appears to achieve a significant boost over the baseline before fusion. This seems plausible due to the nature of task at hand: emotions and intent have been known to be speaker specific. System **CTR** surprisingly does not have the expected gain in performance. We hypothesize that even though the premise of improving the class statistics by enhancing contrastiveness is valid, the manner in which we have performed the manipulation might be flaky. For example, given manipulating pitch might not be the best way to improve contrastiveness when the classes are separated by

valence. However, we do see improvements with the Atypical affect subchallenge. Specifically, the recall for the class angry seems to improve with very little augmentation. Another observation with respect to system **CBR** is that the 'neutral' class seems to be very sensitive to any subsampling.

3.18 Sleepiness Detection

3.18.1 Ordinal Data

A significant amount of data generated by our world, from natural forces to human behavior, is effectively continuous. As a result, humans' tendency to bin continuous data **tee2018brain** has given rise to enormous amounts of ordinal data for applications ranging from healthcare to recommender systems **Marateb2014ManipulatingMS**; **Melville2017** Thus, while humans tend to assign hard labels, the underlying data generally lies on a continuous spectrum. In order to perform effectively, statistical models must be able to capture the underlying data distribution rather than the humans' potentially subjective, and consequently noisy, discrete values. In a limited data setting where using sheer data size to generalize models is not an option, alternative techniques are required to make full use of the available data.

Leveraging the ordinal nature of a dataset as opposed to treating the classes as categorical is one effective approach for extracting more information from a limited set of samples. Many ordinal regression techniques have been proposed throughout the long-standing history of the field and have been traditionally applied to simpler tasks and non-deep models **chu2005ordinal**; **rennie2005ordistic** For complex data that generally require deeper architectures, the large number of parameters in these ordinal techniques can tend to result in overfitting. Thus, simpler approaches are required in order to effectively integrate ordinal techniques into deep networks.

While treating continuous values as ordinals has good bearings intuitively, it is hard to train deep models that can effectively work with such data since standard classification techniques in deep learning are categorical. Hence, they cannot for example take into account the fact that class 2 is closer to class 3 as opposed to class 8. In order to effectively capture this information in a model, one approach is to construct an output distribution that reflects the relationship between classes. Soft labeling is one such technique that has been empirically shown to be effective with noisy ordinal data **zhang2019fsim** Our proposed approach builds on this idea of leveraging ordinal relations to generalize from limited noisy data, namely via learning the relative distances between the encoded representations of different data samples.

3.18.2 Ordinal Regression

Each audio sample in the dataset is labeled with a number based on the KSS scale **shahid2012kss** Since numbers on this scale follow a clear ranking, approaches in ordinal regression can be applied to this task. Namely, instead of penalizing all incorrect labels equally as in traditional multi-class classification, we can leverage the intuition that an incorrectly predicted class $\hat{y}$ that is numerically closer to the actual class y should be penalized less than a farther $\hat{y}$. Two primary ordinal regression techniques that have been applied to statistical models include ordistic loss, which represents the output distribution as a mixture of Gaussians, and a thresholding-based approach which learns the decision boundary between adjacent classes **rennie2005ordistic** Since both approaches involve many parameters, utlizing them in a deep architecture can lead to overfitting.

Soft labels have been shown to not only work effectively with neural models, but also help with convergence and training on noisy data **Hinton2015DistillingTK**; **zhang2019fsim** While not originally created for ordinal tasks, empirical results suggest that soft labelling can be effectively applied to ordinal regression problems **zhang2019fsim** In this paper, we show why soft labelling is particularly effective for ordinal tasks and propose a general

deep approach that learns ordinal relationships through soft labels and relative distance constraints.

3.18.3 Deep Metric Learning

Deep metric learning (DML) encompasses approaches that capture the similarity between datapoints via deep architectures. One such technique is the triplet loss function **schroff2015facenet** which constrains models to map input data from the same class to similar locations in an embedding space and data from different classes to separate locations. Specifically, the loss function for a triple (x_a, x_p, x_n) with respective classes $y_a = y_p \neq y_n$ is given by

$$\|f(x_a) - f(x_p)\|_2^2 - \|f(x_a) - f(x_n)\|_2^2 + \alpha,$$

where $\|\cdot\|$ is the Euclidean norm, $f(x)$ is the encoded representation of x, and α is a hyper-parameter representing the margin between same-class and different-class pairs.

Previous works have shown the effectiveness of triplet loss and Siamese architectures in limited data settings **koch2015siamese** Siamese networks perform well in such cases since they keep the number of model parameters low through weight sharing and effectively increase the dataset size through accepting multiple inputs at a time. Additionally, by encouraging input representations to cluster spatially by their class labels, these approaches can implicitly accentuate features useful for downstream classification tasks.

Like many other DML techniques, triplet loss is designed for categorical data, and consequently does not leverage any properties of ordinal data. We propose an augmented loss function, which we refer to as ordinal triplet loss in later sections, that captures the ordered nature of a collection of data through accounting for the absolute difference between class labels in its relative distance constraints.

3.18.4 Proposed Approach

Our proposed OTL approach is mainly comprised of two parts: soft labelling and an ordinal triplet loss function. Previous works have demonstrated the superiority of soft labels over hard labels for tasks with noisy data **zhang2019fsim** In Section 3.1., we show that soft labels are especially suited for ordinal tasks via a statistical interpretation. The ordinal triplet loss function serves to encourage the model to learn representations specific to the ordinal task at hand by adding a loss constraint to a hidden layer. We discuss the formulation of the loss function in Section 3.2 and how to integrate it into a deep architecture in Section 3.3.

3.18.5 Soft Labels

Results from Zhang et al **zhang2019fsim** suggest that soft labels are well-suited for tasks with noisy, complex data. We reformulate their approach through a statistical lens in order to evince its particular effectiveness for ordinal tasks.

In a K-class ordinal task, we can uniformly scale a class label $k \in \{0, 1, \ldots, K-1\}$ to the interval $[0, 1]$, i.e. mapping class k to $k/(K-1)$, without losing generality. Additionally, through associating a datapoint in original class k with a pair $(k/(K-1), 1-k/(K-1))$ that sums to 1, we can reinterpret the class as a combination of binary labels. In other words, we can interpret the datapoint as being a combination of $k/(K-1)^{th}$ of a class-0 datapoint and $1-k/(K-1)^{th}$ of a class-1 datapoint. Assuming that the binary classes are generated from a Bernoulli distribution, we can express the likelihood of a set of data $\{x_1, x_2, \ldots, x_B\}$ with respective classes $\{y_1, y_2, \ldots, y_B\}$ as

$$\prod_{i=1}^{B} f(x_i)^{\frac{y_i}{K-1}} (1 - f(x_i))^{1-\frac{y_i}{K-1}},$$

where $f(x_i)$ is the model output for datapoint x_i. We can thus maximize this likelihood by training the model using the class pairs via cross-entropy loss. During test time, we invert the class-to-soft-label function to retrieve class predictions, namely mapping a pair $(\hat{p}, 1-\hat{p})$ to $\lceil \hat{p}(K-1) \rfloor$, where $\lceil \cdot \rfloor$ is the nearest integer function.

Training the model in this matter naturally penalizes class predictions more the farther they are from the true class, thus capturing the ordinal nature of the data. In fact, due to the curvature of the log likelihood function, loss penalties approximately increase exponentially with respect to distance to the middle class, capturing the central tendency bias inherent in datasets using the Likert scale. It is worth noting that this soft label formulation works with ordered data in general, including continuous data.

3.18.6 Ordinal Triplet Loss

Ordinal triplet loss augments the traditional triplet loss function **schroff2015facenet** by capturing ordinal relations, thus further utilizing properties in a limited corpus. Namely, the function adds a constraint ensuring that datapoints with farther class labels have larger distances between them in their embedded space. Each input triplet is comprised of an anchor sample x_a, another sample x_s, and a sample x_d constrained to have a class farther from x_a than x_s. In other words, their respective class labels satisfy

$$|y_a - y_d| > |y_a - y_s| + \alpha,$$

where $\alpha \in N$ is a hyperparameter. Since x_s does not need to have the same class as x_a, the resulting set of possible triplets is noticeably larger than that of the traditional triplet loss formulation. When appropriate techniques described in Section 3.4 are applied to select which triplets to train, this expanded set of triplets can help the model generalize better. The ordinal triplet loss for a triplet (x_a, x_s, x_d) is given by

$$\sigma(\|f(x_a) - f(x_d)\| - \|f(x_a) - f(x_s)\|),$$

where $f(x)$ is the encoded representation of x, $\|\cdot\|$ is the Euclidean norm, and σ is the logistic function, given by $\sigma(x) = \log(1+e^{-x})$. Conceptually, the loss function penalizes cases where the model maps the x's to representations where x_a is closer to x_d than x_s. The logistic function serves to make the loss function differentiable. Like the soft label approach, ordinal triplet loss can be applied to continuous data as well.

3.18.7 Network Architecture

We use an architecture similar to that of Zhang et al **zhang2019fsim** to train our model, replacing their loss functions with ordinal triplet loss. Namely, the model receives triplet inputs and jointly optimizes the ordinal triplet loss function, which uses all three inputs, and the soft label cross-entropy loss, which uses only the anchor samples. Each iteration, the model embeds all inputs using an encoder f before applying ordinal triplet loss, and passes the anchor sample embeddings through an MLP g before applying the soft label cross-entropy loss. We add a batch norm layer between f and g to help with convergence. The loss function for a batch $\{(x_1, y_1), (x_2, y_2), \ldots, (x_B, y_B)\}$ is given by

$$\frac{1}{B}\left(\sum_{i=1}^{B} l_s(x_a^{(i)}, y_a^{(i)}) + \beta \sum_{i=1}^{B} l_t(x_a^{(i)}, x_s^{(i)}, x_d^{(i)})\right),$$

where l_t is the ordinal triplet loss function, l_s is the soft label cross-entropy loss function, and β is a hyperparameter describing how much to weigh the ordinal triplet loss.

Conceptually, f serves to separate embeddings in a manner that captures ordinal relations in order to help g in the downstream classification task. As with other Siamese architectures **koch2015siamese** the weight sharing between elements in each triplet and the increased number of possible inputs via grouping samples into tuples aims to help with training effectively on limited amounts of complex data.

3.18.8 Implementation Details

Since the number of possible triplets is cubic with respect to the number of data samples, training using the traditional epoch formulation is impractical. Thus, we choose datapoints using an ordinal version of the triplet loss semi-hard sampling approach **schroff2015facenet** Namely, given an (x_a, x_s) pair, we select the x_d with the minimum $\|f(x_a) - f(x_d)\|$ that satisfies

$$\|f(x_a) - f(x_d)\| > \|f(x_a) - f(x_s)\|,$$

as well as the class label constraint $|y_a - y_d| > |y_a - y_s| + \alpha$.

3.19 Experiments

We describe in the following sections the experiments we conducted to achieve our best model. Our experiments generally proceeded in four parts: selecting features to train our models, modifying them to improve convergence, experimenting with soft labelling, and finally testing our proposed ordinal triplet loss formulation. All experiments used the Adam optimizer and a learning rate scheduler which decreased the rate by a factor of 0.1 after 10 epochs of no improvement.

3.19.1 Feature Selection

Table 1 describes the experiments we conducted to select the best features to use for our model. Features tested include the ComParE baseline features, SoundNet features, MFCCs, and raw waveforms. Of the ComParE baseline features, we observed that ComParE, BoAW-2000, and auDeep-fused yielded the best performances for both neural and statistical models. SoundNet features are extracted from the pretrained network with the same name Aytar, Vondrick, and Torralba, 2016. We used the MFCCs to train a multi-layer LSTM augmented with an attention mechanism. The raw waveforms were used to train a deep network comprised of two convolutional layers followed by a multi-layer LSTM. For the SoundNet and baseline features, we used MLPs structured such that each subsequent layer in the network has approximately half the number of units as the previous one. SVM results for ComParE, BoAW-2000, and auDeep-fused are based on those reported in the challenge paper **schuller2019interspeech** We observed that of the tested features, the three listed baseline features yielded the best results, as bolded in the table.

3.19.2 Data Modification

Table 2 on the next page describes the experiments we conducted to modify the input data. Namely, we tested upsampling and weighting the classification loss by class label frequencies as potential approaches to reconcile the skewed data distribution. We also tested applying PCA on the input features before feeding them into the model as a potential approach to reduce the high dimensionality of the features. For all the experiments in this section, we used MLPs with the halving property described in the previous section. We observed that these data modification approaches did not consistently improve the model, and thus did not use them in subsequent experiments.

3.19.3 Impact of Soft Labels

Table 3 describes the results from using the soft labelling formulation. All experiments in this section also used MLPs with the halving property described earlier. We observe that models trained on soft labels perform noticeably better than models trained on hard labels for two of the three feature types.

3.19.4 Impact of Ordinal Triplet Loss

Table 4 below summarizes our results using ordinal triplet loss. We train all models in this formulation using the Adam optimizer with learning rate 10^{-7}, the joint loss described in Section 3.3, batch sizes of 64, and early stopping with a patience of 10. For our models trained via ordinal triplet loss, f is an MLP with input dimensions halved for each subsequent layer, and g is comprised of two fully connected layers. We observe that utilizing ordinal triplet loss yields noticeable improvement in model performance with respect to the BoAW-2000 feature set.

3.19.5 Analysis of Results

Figure 1 plots the t-SNE visualization of the training data in our model's embedding space. Lighter points represent data samples with higher class labels. The model is able to successfully learn a space that captures desirable ordinal relations, generally mapping data with closer class labels to closer locations in the embedding space.

3.20 *Proposed Approach*

3.20.1 Generative Models of poly species acoustics

Let us consider that an audio corpus A consists of acoustic recordings from different species $\{s_1, s_2..., s_n\}$, where each s_i might comprise of multiple instances in a particular species. It is desirable to define a set of attributes that can describe the generative process behind A. For this, let an audio utterance $u \subset$ A be described by a set of attributes **A** where the individual attributes could be the identity of species, style of utterance, content, etc. Then the prior distribution of A can be represented by a parametric function g such that g maximizes the likelihood of A over the set of its attributes:

$$P_\omega(X) = g_\omega(A) \tag{3.30}$$

We posit that to explain the generative process behind A, it is sufficient to disentangle the causal factors of variations in A and then compose them to reconstruct u. To illustrate this, let us consider a toy-example where we build a machine learning model capable of generating Pythagorean triplets. Pythagorean triplets are a triplet of numbers that follow Pythagoras Theorem such as $\{3, 4, 5\}$ and $\{5, 12, 13\}$. In this task, the attribute-set consists of the relationship between the first two-elements of the triplet. If the model is able to discover this attribute, it can generalize for any given numbers.

For synthesizing audio from multiple species, the generative process needs to be able

to disentangle the appropriate individual attributes from the observed data A_{obs} and also compose them. Based on this, we are interested in working with a set of models that provide flexibility to both decompose / disentangle the causal factors of variation in the input data and also have the ability to combine them. To accomplish this, we use latent stochastic variable models.

3.20.2 VQVAE

We hypothesize that if we use background knowledge about the data distribution while designing the priors, we can help the encoder effectively disentangle the latent causal factors of variation in the data. This presents us with an opportunity to control what gets disentangled in the latent space by appropriately choosing prior distribution. In the current context, a desirable requirement from encoder is to generate task agnostic yet shared representations that help discriminate the intermediate classes. Therefore, we engineer our prior space to account for the segmental information in the utterance by representing the prior as a discrete latent variable bank, similar to the filterbanks used for feature extraction from speech. Each discrete latent variable has a different set of parameters. We have chosen the number of classes to be 64.

3.21 Experiments

In the following we present the preliminary results obtained using the systems we investigated on the Styrian Dialect sub challenge. We further present the results of UAR for blind tests for all the three sub-challenges. We have used our submission from previous year as one of our baselines.

3.21.1 Baseline System - SoundNet

SoundNet Aytar, Vondrick, and Torralba, 2016 is a convolutional network operates on raw waveforms and is trained to predict the objects and scenes in video streams at certain points. After the network is trained, the activations of its intermediate layers can be considered a representation of the audio suitable for classification. It has to be noted that SoundNet is a fully convolutional network, in which the frame rate decreases with each layer. Since we need to predict the emotions with reasonable recall, we cannot extract features from the higher layers of SoundNet directly.

The original SoundNet network has seven hidden convolutional layers interspersed with maxpooling layers. Each convolutional layer essentially doubles the number of feature maps and halves the frame rate. The network is trained to minimize the KL divergence from the ground truth distributions to the predicted distributions. In the original SoundNet architecture, the higher layers have been subsampled too much to be used directly for feature extraction. In order to fully exploit the information in the higher layers, we train a fully connected variant of SoundNet (see Fig. 3.20). Instead of using convolutional layers all the way up, we switch to fully connected layers after the 5th layer. We have also changed the input sampling rate to 16 to match the provided data.

3.21.2 Temporal classification System

Low level features

For acoustic feature extraction we divided each utterance (length is 8) into 25 segments with a 10 frame shift. For each frame we extract 13 mel-frequency cepstral coefficients and their deltas and double-deltas obtaining a feature vector of 39 dimensions. We further extract the log pitch ($f0$) and strengths of excitation (5 dim) **yoshimura2001mixed** In addition, we also obtain 40 dimensional filter banks and 23 dimensional PLP based features. Filter banks have been obtained using the open source toolkit Kaldi Povey et al., 2011 with

'dithering' enabled as it was shown to be robust in other experiments. We have also extracted several features using Opensmile toolkit Eyben, Wöllmer, and Schuller, 2010 and performed singular value decomposition with the intention of obtaining an acoustic representation. This procedure also results in a dense low dimensional representation. This representation was later used in combination with the high level features we obtained in the spirit of early fusion.

Classifier

Using all previously mentioned features, we train a 2 layer bidirectional LSTM network with 512 units in each cell. This is followed by 2 fully connected layers each with 512 units. The final softmax layer dimensions were dependent on the sub challenge. The network is trained by minimizing the expected divergence between the classes using cylindrical SGD Smith, 2017.

3.21.3 VQVAE based System

The architecture of our proposed model continues from **unsupervised_representation_learning_wave** with some modifications. As our encoder, we use the same encoder mentioned in **unsupervised_represent** that downsamples the input audio by 64. We have used WaveNetAaron Van Den Oord et al., 2016 as our decoder. Following Strubell et al., 2017, we have shared the parameters of all the residual layers with common dilation factors. We use Mixture of Logistics loss to train the model and the number of logistics was set to 10. Audio signal was power normalized and squashed to the range (-1,1). To make the training faster, we have used chunks of 4000 time steps. Quantizer acts as a bottleneck and performs a similarity matching to generate the appropriate code from a parameterized learnable codebook. We define the latent space $e \in R^{k \times d}$ contains k d-dim continuous vector. The similarity measure is implemented using minimum distance in the embedding space. We have used 128 dimensions to perform the comparison.

3.21.4 Class balancing by data augmentation

We systematically try to increase the data points from the classes with lesser number of examples. Since our utterance level feature extractor realizes different feature representation for audios of different length, we have experimented with augmenting the original data with additional data created by randomly joining audio rom the same class. For this, we have combined all the audio files and split them into longer chunks. We have made 5 second and 3 second chunks in this fashion and augmented their features to the original set.

3.21.5 Speaker identity based experiments(System SI)

Since the classifiers we use are discriminative in nature, we experiment with two ways of incorporating speakers or subject specific information:

- (1) We add the identity of the speaker as an extra dimension thus forcing the model to build speaker specific models. For example, in case of decision trees, this forces the model to split at the identity of speaker.
- (2) Normalizing with respect to the speaker, following the procedure typically used in speech recognition.

3.21.6 Blind Test Results and Discussion

The evaluation results on blind test set for the three sub-challenges is mentioned in the table 3.15. Based on the preliminary experiments, we have utilized our generative model based approach for evaluating with the blind test set. The results from blind test are presented in the table 3.15.

Code Mixing is a phenomenon where linguistic units such as phrases, words and morphemes of one language are embedded into an utterance of another language Muysken, 2000; Gella, Bali, and Choudhury, 2014. This is quite common in multilingual societies such as in

India where English has transitioned from the status of a foreign language to that of a second language. Today such mixing has manifested itself in various types of text ranging all the way from news articles through comments/posts on social media, leading to co-existence of multiple languages in the same sentence. In the context of Text to Speech (TTS), voice deployed in such contexts has to be able to synthesize mixed text without ignoring the content from one of the languages. Typical approaches for building such mixed lingual voices require bilingual recordingsTraber et al., 1999; S. Rallabandi and A. W. Black, 2017; Chandu, S. K. Rallabandi, et al., 2017: speech data from the speaker in both native language as well as the additional language. However, obtaining such data might not always be feasible. On the other hand, social media and web 2.0 has enabled an outburst of audiovisual content at an unprecedented rate. Therefore, it might be useful to design techniques that can leverage such resources. In this paper, we present initial steps in that direction.

We investigate training strategies for building code mixed voices subject to the availability of only monolingual data in participating languages. Specifically, we concern ourselves with two scenarios: (1) Mixing in the case of a sentence which is primarily Indic but interspersed with English words. Such sentences are found as a newspaper headlines (Ex: *Microsoft ki mobile devices unit ne apni nayee smart phone Lumia 640 aur uske badee screen wali variant 640 par se parda utha liya hai.*) (2) Mixing in the case of a sentence which is primarily English but has some Indic words. Such sentences are found as navigation instructions (Ex: *Proceed for 100 meters and then take a left at Sarojini Naidu Nagar Road, heading onto the Ballary chowrasta.*) Although building voices using such a combination of multilingual corpora appears as a simple extension of multispeaker or multilingual speech synthesis, generating code mixed content is a deceptively non trivial task since there is a mismatch between training and the testing scenarios: Even though the model has access to data from both the participating languages during training, code mixed content it is exposed to at test time - as seen from the example sentences - is a novel composition of linguistic units from both the languages. To assist the model in dealing with such mismatch, we incorporate latent

stochastic variables into the training procedure.

Models with latent random variables (referred to as latent stochastic variable models hereafter) provide flexibility to jointly train the latent representations as well as the downstream network. They are expected to both discover and disentangle causal factors of variation present in the distribution of original data, so as to generalize at inference time. However, while training latent stochastic variable models, optimizing the exact log likelihood can be intractable. To address this, a recognition network is employed to approximate the posterior probability using reparameterization Kingma and Welling, 2013b. We make an observation that articulatory information about speech production presents a discrete set of independent constraints. For instance, manner and place of articulation are two articulatory dimensions characterized by discrete sets(labial vs dental, etc). Based on this, we condition the recognition network in latent stochastic variable models to conform to articulatory prior space by using a bank of discrete prior distributions. We show that such priors help encode language independent information thereby facilitating synthesis of code mixed content.

3.21.7 Synthesis of Code Mixed Text

Synthesis of code mixed text using monolingual data Elluru et al., 2013; Chandu, S. K. Rallabandi, et al., 2017 has been addressed primarily at the linguistic level: by either mapping the words/phones of the foreign language with the closest sounding phones of the native language or by using transliteration Sitaram, S. K. Rallabandi, et al., 2015; Sitaram and A. W. Black, 2016. However, these methods have been shown to generate foreign accents Tomokiyo, A. W. Black, and Lenzo, 2005; Campbell, 2001; Badino, Barolo, and Quazza, 2004. In our work, we borrow the central idea from the works - the requirement of a common linguistic space - and apply this as a constraint on the representations learnt in our latent stochastic variable model. In Yao Qian, Xu, and Soong, 2011, the authors follow a two step procedure to address the issue with accented speech. They first warp the source

speakers' speech parameter trajectories (in L1) towards the target speaker and then 'tile' them with the data (in L2) to form a pseudo training corpus which is subsequently used to train a bilingual speech synthesis system. Similar practices can be found in the literature for voice adaptation Kain and Macon, 1998; Kurimo et al., 2010; Oura et al., 2010; Yamagishi et al., 2009; Latorre, Iwano, and Furui, 2005 and voice conversion Sundermann et al., 2006. Although we do not explicitly aim to transfer acoustic parameters, our decoder is engineered to work with a global speaker embedding that learns speaker specific information. Therefore, our approach can be seen as analogous to these works. Our work is closest to Yuewen Cao et al., 2019; Xue et al., 2019 in that we use monolingual recordings. However, we explicitly work in the latent prior space while Yuewen Cao et al., 2019 operate at the level of encoding individual languages and Xue et al., 2019 begin with an average voice and refine it using phoneme informed attention.

3.21.8 *VACONDA*[1] - *Va*riational *A*ttention based *CON*trolled *D*isentanglemen using *A*rticulatory priors

We make an observation that dealing with speech presents a characteristic advantage - speech has both continuous as well as discrete priors. The generative process of speech assumes a Gaussian prior distribution which is continuous in nature. However, the language which is also present in the utterance can be approximated to be sampled from a discrete prior distribution. Exact manifestation of this in linguistics can be at different levels: phonemes, words, syllables, sub word units, etc. From the analysis presented in previous subsections, we posit that it helps encoder effectively disentangle the latent causal factors of variation if we use background knowledge about the data distribution while designing the priors. In other words, incorporating appropriated priors provides us with an opportunity to control what gets disentangled (or) decomposed (or) factorized in the latent space. In our context,

[1]Phonetically similar to its namesake 'Wakanda' from Marvel Comics

an appropriate requirement from the encoder is to generate language agnostic yet phonetic representations such that a speaker dependent decoder can synthesize code mixed content. Therefore, we engineer our prior space to account for phonetic information in the utterance by representing the prior as a discrete latent variable bank, similar to filterbanks used for feature extraction from speech. Each discrete latent variable has a different set of states reflecting one of the articulatory dimensions. The specific design of our latent space is highlighted in the table 3.16. Voice building procedure with these priors is depicted in figure 3.21. We have used the articulatory dimensions according to the definitions in Indic voice building process of A. W. Black, 2006. Although some of them might be redundant, for this initial study we have retained all the articulatory dimensions. Without loss of generality, we assume that the individual latent articulatory dimensions are independent of each other. The divergence between the true prior and approximate prior now becomes:

$$D_{KL}(q_\phi(z_{enc}|p)||p(z_{code})) = \sum_{i=1}^{N}[E_{q_\phi(z^i_{enc}|p)}[logq_\phi(z^i_{enc}|p)] - E_{q_\phi(z^i_{enc}|p)}[logp(z^i_{code})]]$$

where N is the number of articulatory dimensions and i denotes the index of individual articulatory dimensions. z_{code} denotes the parameterized codebook and z_{enc} denotes the representation output by the encoder.

3.22 Experiments

3.22.1 Data

We have used speech and text data from three Indian languages Hindi, Telugu and Marathi released as a part of resources for Indian languages Baby, n.d. to build our synthesis systems. From our baseline voice building process, we found male speaker from Hindi to be the most reliable voice in terms of quality. Therefore, all of our systems use English recordings from Mono segment of this speaker as English set - as a scaffolding. For other two languages, we use only monolingual data from the speakers. In other words, to generate code mixed

Telugu sentence, the systems have access to English content but from a different speaker. As baseline for comparison, we have built a CLUSTERGEN voice using monolingual recordings employing phone mapping. Evaluation was performed in the form of listening tests with 20 native students following the convention of Blizzard Challenge evaluations using Parlikar, 2012a with naturalness as criterion in terms of Mean Opinion Score (MOS) on a scale of 1(least natural) to 5(highly natural). All the listening tests involved test sentences generated using the Multilingual test set (ML) from Prahallad et al., 2014. The evaluation results are depicted in table 3.17.

3.22.2 Implementation Details

We have built two systems employing variational attention: VQTacotron with vanilla vector quantization and VACONDA - with articulatory prior on the latent space. The architecture of our models continues from Baljiker, S. K. Rallabandi, and A. Black, 2018, with some modifications. We have used WaveNetAaron Van Den Oord et al., 2016 as our decoder. Following Strubell et al., 2017, we have shared the parameters of all the residual layers with common dilation factors. We use Mixture of Logistics loss to train the model and the number of logistics was set to 10. Speech signal was power normalized and squashed to the range (-1,1). To make the training faster, we have used chunks of 8000 time steps. Our quantizer performs vector quantization to generate the appropriate code from a parameterized codebook. We define the latent space $e \in R^{k \times d}$ contains k d-dim continuous vector. Quantization is implemented using minimum distance in the embedding space. We have used 128 dimensions to perform the comparison in system VQTacotron. The number of classes was chosen to be 64, approximating 64 universal phonemes. For system VACONDA, we use a linear mapping to first project the 128 dimensional vector to 160 dimensions. We then perform comparison with respect to individual articulatory dimensions each of which is 16 in size. The speaker embedding is shared between the decoder of our acoustic model and WaveNet.

We have noticed the lengths of utterances in the Indic datasets being too big to train attention from scratch. Therefore we have initialized attention using alignments performed within Festvox using HMM aligner. All the models were built at phone level since that was observed to be the most stable configuration even though our phones do not cover all the variants (ex. we do not have explicit phones for geminates). We have used quantization penalty and commitment loss terms as mentioned in Chorowski et al., 2019. In addition, we have also normalized each latent embedding vector to be on a unit sphere.

3.22.3 Observations

An informal analysis on the outputs from the proposed systems revealed that the characteristics of the English speaker were retained in certain areas within the utterance, resulting in a slightly stylized version[2]. We want to investigate this further and hope to uncover techniques that can provide more control. While most of the systems using CLUSTERGEN S. Rallabandi and A. W. Black, 2017 make errors in the prosodic features such as irregular duration shifts at the boundaries between languages, the proposed approaches have smooth transitions at the boundaries. However, we have observed marked differences in the pronunciations by the proposed approaches. For instance, the phone 'S' from the word 'Stanford' when heard in isolation is indistinguishable from other fricative sounds. Since we specifically deal with articulatory priors in VACONDA, a reasonable assumption to make is that this issue will be bypassed by the model. However, this characteristic is common across voices built using both VQTacotron as well as VACONDA.

[2]The samples can be found here http://www.cs.cmu.edu/~srallaba/IS2019_CodeMixedTTS/.

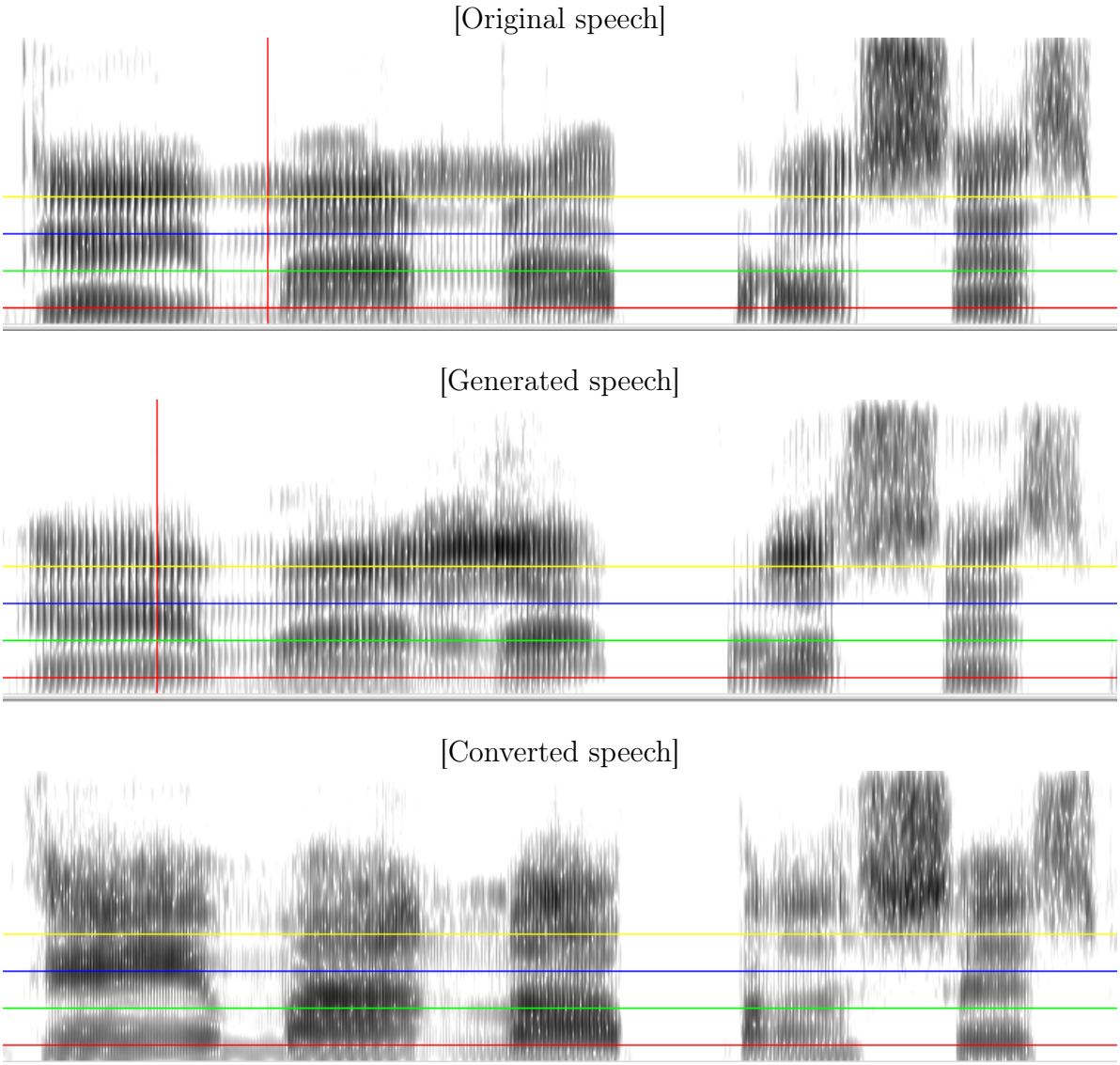


Figure 3.2 Spectrograms of original, generated, and converted speech. The source speaker is female while the target speaker is male.

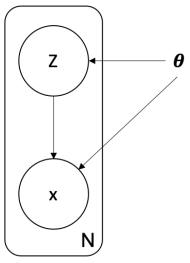


Figure 3.3 Latent Variable Model - Variational Autoencoder

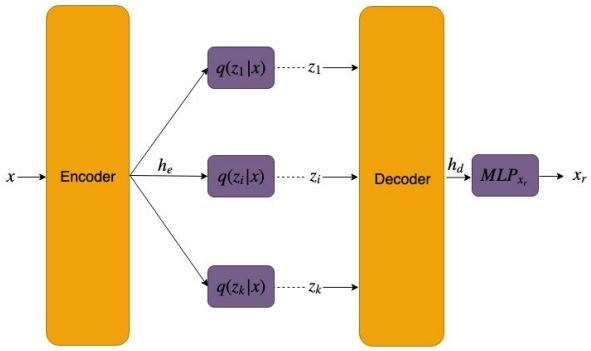


Figure 3.4 Multi-node VAE model. Dashed lines represent sampling using reparametrization. Encoder and Decoder are Bi-LSTM networks. Purple blocks are fully connected layers.

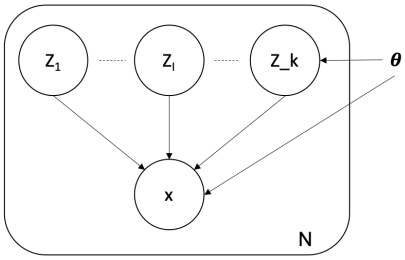


Figure 3.5 Latent Variable Model - Multinode Variational Autoencoder

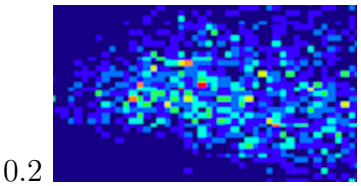


Figure 3.6 Wilderness

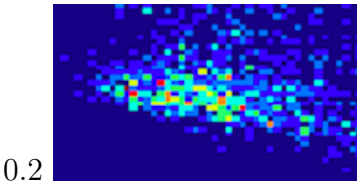


Figure 3.7 Hub4

Figure 3.8 Input Data Distributions for (a) Wilderness (b) Hub4. The red dots show the high density regions in each distribution.

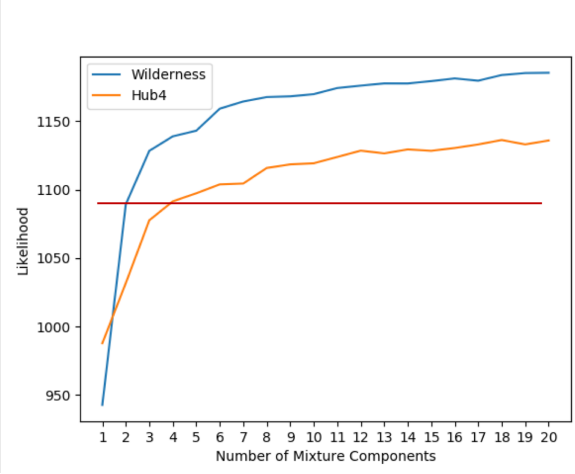
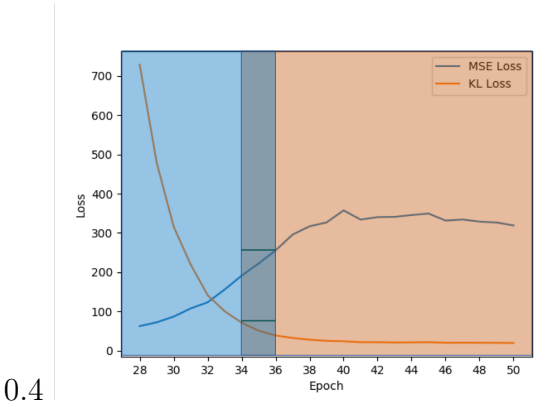
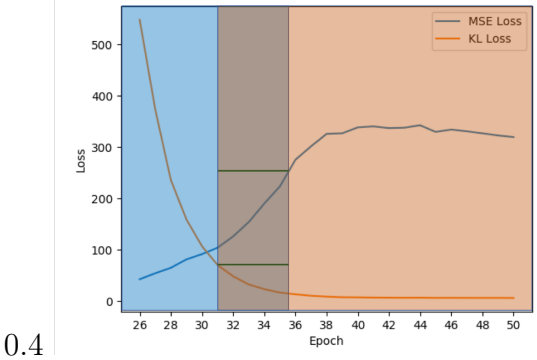


Figure 3.9 Gaussian Mixture Fit for Wilderness and Hub4



0.4

Figure 3.10 1-Node VAE: Wilderness



0.4

Figure 3.11 3-Node VAE: Wilderness

Figure 3.12 Training Loss (a) 1-Node VAE: Wilderness (b) 3-Node VAE: Wilderness. The left shaded blue region and the right shaded orange region show the required MSE loss and KL loss threshold respectively to obtain good speech. Overlap region represents the model parameters where speech separation occurs.

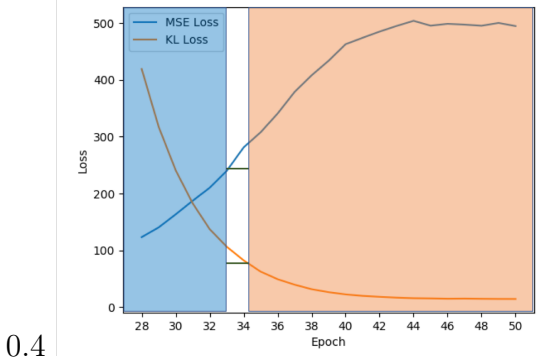


Figure 3.13 1-Node VAE: Hub4

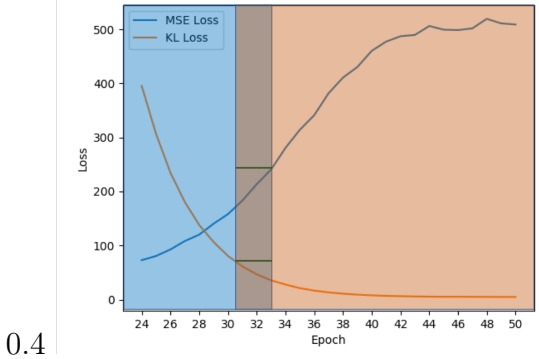


Figure 3.14 3-Node VAE: Hub4

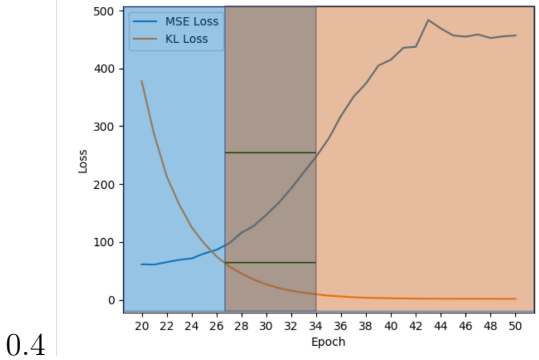


Figure 3.15 8-Node VAE: Hub4

Figure 3.16 Training Loss (a) 1-Node VAE: Hub4 (b) 3-Node VAE: Hub4 (c) 8-Node VAE: Hub4. The left shaded blue region and the right shaded orange region show the required MSE loss and KL loss threshold respectively to obtain good speech. Overlap region represents the model parameters where speech separation occurs.

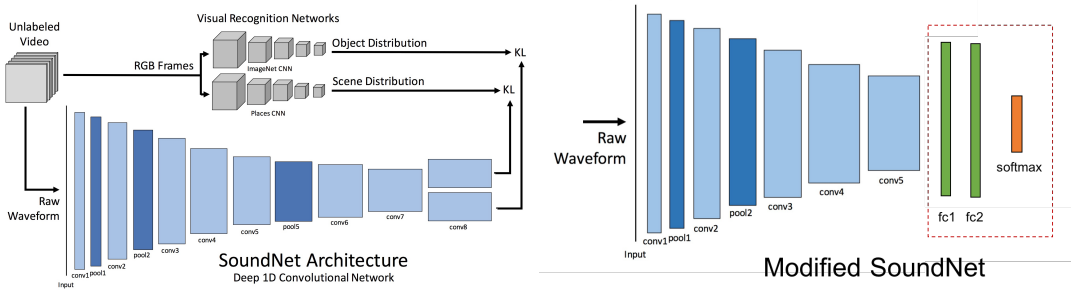


Figure 3.17 Original SoundNet architecture Aytar, Vondrick, and Torralba, 2016 on top and modified SoundNet architecture at the bottom. The modified version uses 2 layers of 512 fully connected (fc) units and a softmax layer of 3 units.

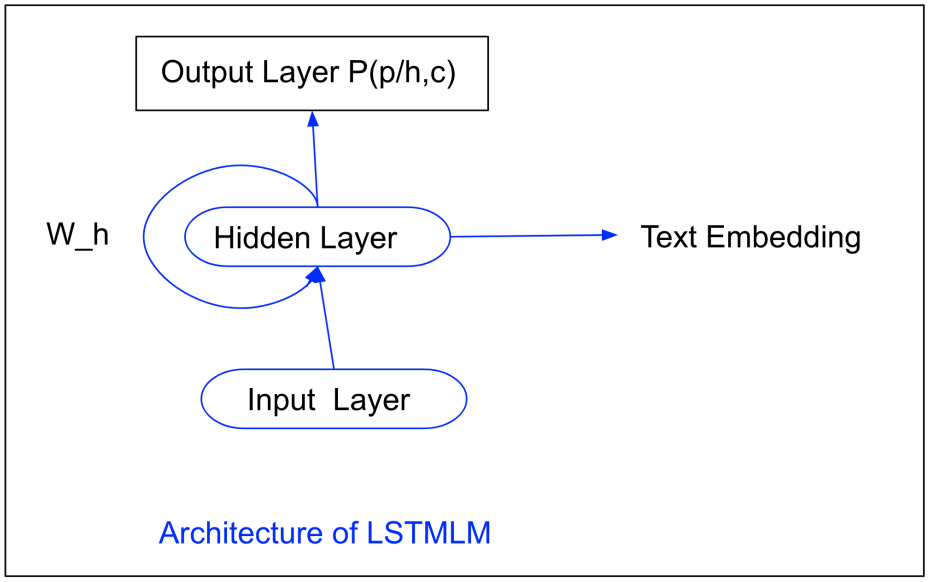


Figure 3.18 Architecture for extracting textual embedding. The hidden layer obtained after passing all phonemes of an utterance, is used as embedding of the same utterance. The phonetic decoding is then obtained using a pre-trained acoustic model.

Table 3.8 UAR for Emphasis and Data Augmentation Experiments

Augmentation []	UAR []
100	54.4
200	58.3
300	57.2
400	[detect-weight]58.8

Table 3.9 UAR Blind test summary

Sub-challenge	UAR
Self Assessed Affect	48.3
Atypical Affect	34.2
CRYING	71.406

Table 3.10 Performance on Different Features

	Model	Spearman (Devel)
SoundNet	MLP	0.030
ComParE	SVM	0.251
	MLP	**0.300**
BoAW-2000	SVM	0.269
	MLP	**0.313**
auDeep-fused	SVM	0.261
	MLP	**0.329**
MFCC	Attention LSTM	0.018
Raw Waveform	CNN LSTM	0.031

Table 3.11 Data Modifications

	Features	Spearman (Devel)
Upsampling	ComParE	0.271
	BoAW-2000	0.308
	auDeep-fused	0.303
PCA	ComParE	0.279
	BoAW-2000	0.325
	auDeep-fused	0.254
Weighted Loss	ComParE	0.279
	BoAW-2000	0.301
	auDeep-fused	0.243

Table 3.12 Soft Labels

Features	Spearman (Devel)
ComParE	**0.311**
BoAW-2000	**0.333**
auDeep-fused	0.322

Table 3.13 Ordinal Triplet Loss

Features	Spearman (Devel)
ComParE	0.308
BoAW-2000	**0.343**
auDeep-fused	0.323

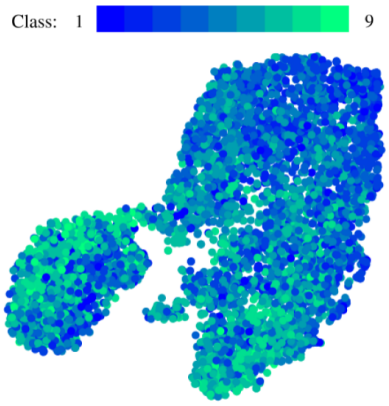


Figure 3.19 t-SNE Visualization of Embedding Space

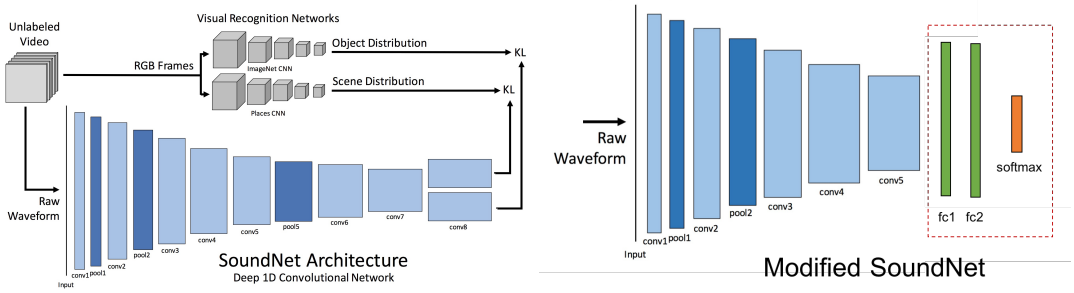


Figure 3.20 Original SoundNet architecture Aytar, Vondrick, and Torralba, 2016 on top and modified SoundNet architecture at the bottom. The modified version uses 2 layers of 512 fully connected (fc) units and a softmax layer of 3 units.

Table 3.14 UAR on Val set. Each model was trained for 100 epochs

Architecture	**Features**	**UAR**
Baseline	AUDEEP	27
Temporal Classification	Low level	32.2
Baseline	SoundNet	36
Proposed	VQVAE Features	48.2

Table 3.15 UAR Blind test summary

Sub-challenge	Metric
Styrian Dialects	47.25
Baby Sounds	57.2
Orca	86.6

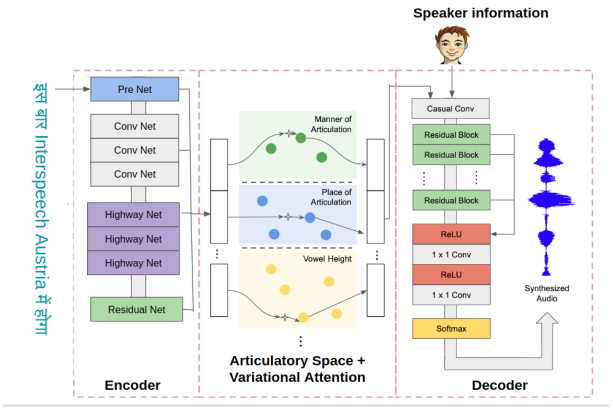


Figure 3.21 Illustration of our procedure for generating a code mixed utterance. Text from different languages is converted into a common representation space by Tacotron encoder. The encoded representation is hashed to a latent code based on a discrete articulatory prior bank. The code is passed to the decoder, followed by a WaveNet using speaker embeddings as global conditioning that generates audio.

Table 3.16 Articulatory Features

[] **Feature name**	**Possible Classes**	**Cardinality**
[] vowel or consonant	+ - 0	3
vowel length	s l d a 0	5
vowel height	1 2 3 0 -	5
vowel frontness	1 2 3 0 -	5
lip rounding	+ - 0	3
consonant type	s f a n l r 0	7
place of articulation	l a p b d v g 0	8
consonant voicing	+ - 0	3
[]		

Table 3.17 MOS Scores for Naturalness in prosodic modeling based experiments

Config	Clustergen	VQTacotron	VACONDA
Hi-Eng (Male)	3.9	**4.31**	4.28
Tel-Eng(Female)	3.6	3.9	**4.1**
Mar-Eng(Male)	3.7	4.0	**4.0**
Mar-Eng(Female)	3.4	3.9	**4.0**

Chapter Four

De-Entanglement of Structure - Case study with Emphatic Speech Synthesis

sectionProblem Motivation and Introduction Humans exhibit both coarse as well as fine grained explicit control over how they speak an utterance. This targeted control on speech - often manifested in the form of prosodic constructions - allows us to effectively convey our intent in a conversation. Examples of controlled speech generation include simple prosodic manipulations such as implying specific meaning, highlighting or expressing interest in something as well as various communication strategies such as contradiction, contrast, complaints or grudging admiration(Ward, 2019). Further, such manipulation in prosody has been shown effective in applications such as Infant Behavior Programs (Morningstar et al., 2019), improving language acquisition(Carvalho et al., 2019) and promoting rapport(Acosta and Ward, 2011). It seems natural to employ generative models of speech(Y. Wang, Skerry-Ryan, Stanton, et al., 2017; Gibiansky et al., 2017; Ping, Peng, and J. Chen, 2018; Biadsy et al., 2019) to assist in such scenarios (Wood et al., 2018). However, although there has been tremendous progress in the neural generative models for speech in the context of vocoder fidelity(Prenger, Valle, and Catanzaro, 2018; Aäron Van Den Oord et al., 2016), the notion of controllability in such models is not yet fully evolved. While there have been works towards models aimed at controlling prosody(Hsu et al., 2018; Y. Wang, Skerry-Ryan, Xiao, et al., 2017), the ex-

erted control is still global or coarse grained in terms of styles of speech(Skerry-Ryan et al., 2018; Y. Wang et al., 2018), etc. In this work, we propose an approach that allows both global as well as local control over the prosodic variation in the generated speech.

Typically TTS is formulated as a conditional generative modeling problem. In our approach, we propose to instead formulate it as a conditional variational auto-encoder and incorporate automatically derivable information from speech data into the model architecture. This is motivated by the understanding that the utterances themselves do not always contain all the information needed to comprehend the appropriate prosody information. The missing information is either part of background knowledge about the world - implicit to humans but not annotated in the data - or is provided by accompanying context of the utterance. Formulating the task using variational inference allows us to efficiently capture the distribution of prosody thereby avoiding the averaging effect observed in a typical TTS system due to prosody marginalization. To illustrate this, consider an example sentence: '*You do not have a pet shark*'. Most prosodic constructions for this sentence involve sarcasm since it is not commonplace to have sharks as pets - world knowledge. Similarly consider the sentence: '*I dont want to be a nun*'. The linguistic unit subject to realization of prosodic stress in this sentence depends on the context information. Finally, consider the example of a TTS system deployed in a screenreader to assist visually impaired students comprehend math equations. Human voice talent would almost certainly place appropriate prosodic cues that help in comprehension of $x^{(y+z)}$ as opposed to $(x^y + z)$. Our formulation allows the model to leverage prosodic information available from the speech signal and capture prosodic distribution.

To accomplish local as well as global prosody control, we incorporate inductive biases into the model architecture in the form of fundamental frequency(F_0). Specifically, we quantize F_0 into multiple bins and constrain the latent space to disentangle these quantized values from acoustics at the level of phonemes. Our model is explained in detail in section 4. During inference, the prosody distribution can be utilized to control and generate variability in the

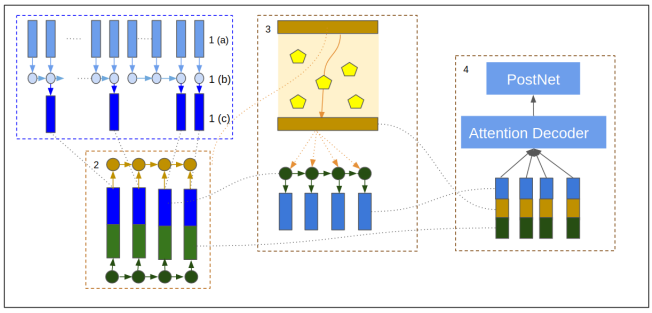


Figure 4.1 *Architecture of EDITH. Circles denote LSTM cells, rectangles represent vectors and pentagons represent global latent vectors. (Best viewed in color)*

output speech. In short, our contributions from this work are: (1) We present EDITH, a hierarchical model that disentangles prosodic features in the form of F_0 enabling explicit global as well as local control. (2) We show that EDITH captures reliable representation of local prosody by generating speech with desired variations at the chosen linguistic level.

sectionEmphasis by Disentangling Tonal Heuristics(EDITH)

EDITH learns the joint distribution between pairs of temporal sequences $\{\mathbf{x}, \mathbf{y}\}$ where $\mathbf{x}$ denotes the features and $\mathbf{y}$ denotes the acoustic parameters. Let Ti and To denote the lengths of input and output sequences respectively. Input features $\mathbf{x}$ consist of both linguistic features denoted as $x^{1:Ti}_{linguistic}$ as well as features extracted from the acoustic signal denoted as $x^{1:To}_{acoustic}$. The output features $\mathbf{y}$ consist of linear feature representation $y^{1:To}_{linear}$ as well as mel features $y^{1:To}_{mel}$. It has to be noted that $x^{1:To}_{acoustic} = y^{1:To}_{mel}$. To efficiently model varying prosody and prevent the averaging effect, we incorporate a variational layer. Therefore, EDITH is a conditional variational auto-encoder. During inference, we discard the encoder part of our model. Our model can be summarized by the following set of equations:

$$\begin{aligned} encoded &= \mathbf{H}^{Encoder}(x_{linguistic}, x_{acoustic}) \\ z_g, z_l &= \mathbf{VI}(encoded) \\ \hat{y}_{mel} &= \mathbf{H}^{Decoder}(x_{linguistic}, z_g, z_l) \\ \hat{y}_{linear} &= \mathbf{H}^{postnet}(\hat{y}_{mel}) \end{aligned} \tag{4.1}$$

Design of our encoder is inpired by the encoder from (Wan et al., 2019). We use clockwork hierarchical LSTM to encode $x_{linguistic}^{1:Ti}$ and $x_{acoustic}^{1:To}$. However, our models are clocked at the rate of phones as opposed to syllables. In addition, we do not incorporate any features from word or sentence levels in our encoder to keep the architecture compact. Our variational layer is derived from (Oord, Vinyals, et al., 2017b) and is employed to generate global and local latent variables z_g, z_l respectively. Our decoder is similar to a typical attention based acoustic decoder(Y. Wang, Skerry-Ryan, Stanton, et al., 2017) and includes a postnet. While similar in formulation, EDITH has an important difference from (Wan et al., 2019) in that our local latent variables follow the rate of input as opposed to output as in (Wan et al., 2019). This allows us to exercise more control over the generated prosodic variations.

Optimization and Learning: $x_{acoustic}$ is passed through phone rate LSTM. This is shown as block 1 in figure 4.1. $x_{linguistic}$ is passed through phone LSTM. The representations are concatenated and passed through EDITH Encoder. This is shown as block 2. Outputs from the encoder are passed through the variational layer where vector quantization is performed to pick the most suitable global latent prosodic vector. Conditioned on encoder outputs and the global latent prosodic vector, we predict Ti local prosodic vectors corresponding to predicted local prosodic features. We constrain the local latent variables to correspond to quantized F_0 by modeling their prediction as a classification task. These local latent variables thus capture the local variations in prosody while global latent variable is reserved for capturing sentence level variations. Ground truth quantized values for classification are obtained by selecting the maximum bin within the duration of phoneme. This is shown as block 3 in the figure. We then employ dot product attention in our decoder. y_{mel} is generated by decoder conditioned on local, global latent variables and the encoded $x_{linguistic}$. A postnet is employed to generate y_{linear} conditioned on y_{mel}. EDITH is optimized to minimize two *L1* losses one each for y_{mel} and y_{linear} and one classification loss for local latent variables. Additionally, to train the vector quantization layer, we minimize encoder commitment loss for z_g and vector quantization loss following Oord, Vinyals, et al., 2017b

for both z_g and z_l. This can be expressed as below:

$$\begin{aligned} L = \lambda_{linear} \sum_{t=0}^{To} \|y^t_{linear} - \hat{y}^t_{linear}\| \\ + \lambda_{mel} \sum_{t=0}^{To} \|y^t_{mel} - \hat{y}^t_{mel}\| \\ + \lambda_{qF_0} \sum_{t=0}^{Ti} Div(qF_0, q\hat{F}_0) + \lambda_e L_e + L_{VQ} \end{aligned} \tag{4.2}$$

sectionModel Interpretation This approach can be interpreted as VQVAE(Oord, Vinyals, et al., 2017b). It can also be seen as GST(Y. Wang et al., 2018) based encoding but our approach has two differences:(1) We do not use a different encoder for spectral information and (2) We explicitly constrain the latent classes to correspond to the quantized F_0s. We divide the model into individual blocks or modules. Therefore, it can be seen as an extension to Neural Module NetworksAndreas et al., 2016. In Wan et al., 2019, authors introduce clockwork hierarchical VAE to predict F_0, duration and C_0. Our approach of incorporating F_0 information at the output of encoder in the form of additional task can be seen similar to this work. However, we use quantized F_0s, do not employ clockwork structure in our model and do not explicitly model duration or C_0.

sectionExperimental Setup

Data: We have used data from LJSpeech dataset(Ito et al., 2017) to build our systems. We have used all of the 13100 sentences. The text was normalized manually to convert non standard forms (for ex. 1993) to written forms (nineteen ninety three).

Baselines: Our acoustic model is based on TacotronY. Wang, Skerry-Ryan, Stanton, et al., 2017 Seq2Seq speech synthesis system is built using PyTorch(Paszke et al., 2019). We have not performed masking of padded frames as is typically done in Seq2Seq models. We found that not masking helps model better predict end of sentence as mentioned in Y. Wang, Skerry-Ryan, Stanton, et al., 2017. Since adjacent frames seem to be correlated, our decoder predicts 5 frames per timestep. Our model has three deviations from the original

implementation: (1) Phones are used as the input instead of characters. (2) CBHG module in the encoder and postnet has been replaced with with three LSTM layers. (3) We use all the predicted frames at a time step as input to the decoder(as opposed to only the last time step) while predicting the next frames. We have used a batch size of 64 to train the baseline model. To enable control of prosody, we employ quantized F_0 values as additional inputs to this baseline model. For this, we first extract F_0 values for the dataset and quantize them into multiple bins each spanning 25 Hz without any overlap. These quantized F_0 values are embedded and added as additional inputs to the baseline model. In other words, this is a conditional generative model with phones and quantized F_0s as inputs. Additionally, we also build a model that uses word level prosodic features extracted using AuToBI(Rosenberg, 2010). We refer to this system as **AuToBI**.

EDITH Hyperparameters: The encoders of both $x_{acoustic}$ and $x_{linguistic}$ are realized using bidirectional LSTMs. We have used 256 as the hidden dimensions for both these encoders. Both our global and local latent variables are of 256 dimensions. We employ 10 global latent classes. The network to predict local latent variables is implemented using bidirectional LSTMs that takes 512 dimensional input and outputs 256 dimensional vectors. Encoder weight λ_e was linearly increased to 0.2 till 10K timesteps and remained constant after that. For quantization of F_0, we have followed the same procedure as in Baseline. 25 Hz was chosen as the size of bin. This effectively resulted in a total of 14 bins and thus 14 local latent classes. After every update step, we normalize the local latent variables by the norm. Since these classes correspond to ordinal data in terms of quantized F_0s, we believe that normalizing places the vectors on a unit circle.

SubUtterance Models: Long utterances present in audiobooks are rich in prosodic variations but also lead to computational overhead in terms of processing speed. Therefore, we have built systems that have access to only part of the utterance by selecting aligned segments of text and acoustics within a full sentence. We note that such an approach is already used for vocoding: Typical vocoders the authors are aware of are trained using

Table 4.1 *Results from Preference and MOS Tests for Emphasis generation. The entries for the preference portion(columns 2 through 6)indicate preference values obtained by the systems in the first column against every other system in the subsequent columns.*

Config	FUB	FUE	SUB	SUE	AUToBI	MOS
FUB	-	92	396	363	**441**	4.0
FUE(ours)	**345**	-	**424**	**378**	**477**	4.0
SUB	91	86	-	235	**278**	3.4
SUE(ours)	64	86	243	-	227	3.6
AUToBI	47	19	219	256	-	3.9

aligned chunks of acoustic vectors and corresponding speech samples as opposed to full utterances. Encouraged by this, we build sub utterance based models for both baseline as well as proposed approach. To distinguish from the full sentence models, we refer to these systems as Sub Utterance Baseline(SUB) and Sub Utterance EDITH(SUE) while referring to the full sentence models as Full Utterance Baseline(FUB) and Full Utterance EDITH(FUE) respectively.

Evaluation: Evaluation was performed in the form of listening tests using (Parlikar, 2012b). We have conducted two types of listening tests: (1) Rating the naturalness in terms of Mean Opinion Score (MOS) on a scale of 1(least natural) to 5(highly natural) and (2) ABX Preference test on Emphasis where the users need to mention their preference towards either of the systems or state that they prefer neither. For the preference evaluation we have manually curated 50 sentences where the meaning was implied based on prosody. Participants were shown the entire sentence and its implication in parenthesis. An example sentence from our testset is '*It looks like a starfish* (but it really is not).' Every system was used to generate this test set[1]. For baseline and proposed approaches, the phonemes to be

[1]in the mentioned example, the systems generated just the part '*It looks like a starfish*' and not the part in parenthesis

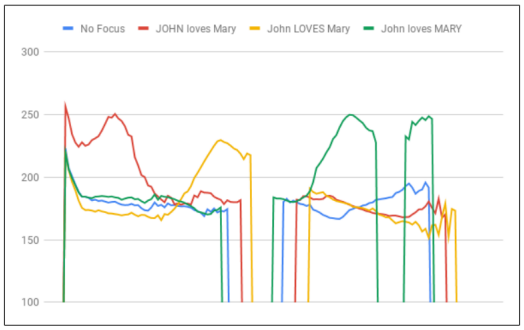


Figure 4.2 *Plot of Fundamental Frequency(F_0) trajectories obtained from generated waves using proposed approach FUE. Variants of the sentence 'John loves Mary' are generated with emphasis on individual words(captialized). The blue trajectory corresponds to F_0 when no emphasis was applied to any word. The plot highlights that the proposed approach allows explicit local control at the desired level in the generated speech. We have submitted the generated wavefiles as supplementary material.*

emphasized are rendered with embedding vector corresponding to bin 12 while others are rendered with bin 8 The participants are to mention their preference to the system that faithfully generates prosody in line with the information in parenthesis. We had 25 listeners and each participant rated 20 random sentences giving us a total of 500 ratings per pair of systems.

Discussion: The preference evaluation results for the proposed approaches are presented in table 4.1. We have excluded the *No Preference* values from this table for brevity. However, they can be estimated based on the values in the table. The full utterance based systems seem to outperform sub utternace as well as AuToBI based systems consistently. Within the full sentence systems, our proposed approach(FUE) outperforms the baseline conditional generative model(FUB). A sample output generated by conditioning the local latent variables to emphasize individual linguistic units(words) from our approach can be examined in figure 4.2. An informal listening test in the scenarios where full sentence models were not preferred revealed an interesting finding: All these scenarios were when the emphasized word was the first in the sentence. We hypothesize that this might be due to the canonical word order(**SVO**) in English. One approach to handle this could be to incorporate a suitable weighting to consider this effect and we plan to investigate this further. The sub utterance

based approaches seem to match the performance of AUToBI systems while clearly under performing their full utterance counterparts. Informal listening evaluations revealed that the sub utterance models seem to have repetition of phoneme units within the generated sentence. We attribute this to the errors in alignment and phoneme boundary estimation and plan to investigate approaches to circumvent this behavior in future work.

sectionConclusion

In this case study, we have proposed an approach to obtain local and fine grained control over prosody in neural generative models for speech. For this we quantize fundamental frequency, which is highly correlated with prosody information, into multiple bins. We infer this information employing hierarchical global and local latent variables in the model architecture. We show that our approach generates appropriate emphasis at word level and significantly outperforms AuToBI in terms of flexibility.

Chapter Five

De-Entanglement by Divergence

5.1 Visual Question Answering

Visual Question Answering (VQA) involves answering a natural language query about an image. Questions can be arbitrary and they encompass many sub-problems in computer vision: (1) Object recognition (2) Object detection (3) Attribute classification (4) Scene classification (5) Counting. VQA is characterized by wide ranging applications from helping visually impaired people through human machine interaction. It has the potential to serve as an effective media content retrieval framework. A primary form of implementing a VQA system would be to use a bucketing approach: by learning image and text features and fusing them to get an answer. In recent years, there have been several extensions to the trivial approach mentioned above Fukui et al., 2016; Lu, J. Yang, et al., 2016; Zichao Yang et al., 2016; Lu, X. Lin, et al., 2015 claim to learn good representations of abstract concepts needed to answer questions. However, it has been shown Agrawal et al., 2017a that most of the approaches capture surface level correlations and fail to handle unseen novel combinations during test time.

In this work, we investigate approaches to improve compositionality in VQA, where we explicitly focus on learning compositionality between concepts and objects. Language and vision are inherently composite in nature. For example different questions share substructure

viz *Where is the dog?* and *Where is the cat?* Similarly images share abstract concepts and attributes viz *green pillow* and *green light*. Hence it is vital not only to focus on understanding the information present across both these modalities, but also to model the abstract relationships so as to capture the unseen compositions of seen concepts at test time. Achieving this would then allow the model to generalize better by learning an inference procedure, resulting in true success on this task.

In this work, we propose ***JUPITER*** - **JU**stification via **P**ointwise combination of **I**mage and **T**ext based on **E**xpected **R**ewards, is built on top of the Neural Module Networks Hu et al., 2017. This is motivated from our hypothesis that generating captions can provide additional information to improve VQA. Additionally, JUPITER uses Reward Augmented Maximum Likelihood Norouzi et al., 2016, which is improves caption generation.

5.2 Related Work

Visual Question Answering: Kazemi and Elqursh, 2017 provided a strong baseline for VQA using a simple CNN-LSTM architecture, and achieved 64.6% on the VQA 1.0 Openended QA challenge. This further proved that the dataset is biased. Agrawal, Batra, et al., 2017 introduced grounding to prevent the model from memorizing this bias. Similarly, Li et al., 2018 used a zero-shot training approach to improve the generalizabilty of the model, and prevent the model to learn the bias. However, recently Agrawal et al., 2017b showed that most models degrade in performance when tested on unseen samples. In this work, we aim to tackle this lack of generalizability.

Neural Module Networks: To the best of our knowledge, the work by authors in Hu et al., 2017 and Andreas et al., 2015 is the only work so far that explicitly uses a divide and conquer approach for compositionality. Natural language questions are best answered when broken down into their subparts. The authors use a similar intution and propose a mod-

ular architecture. This approach first parses the natural language question into linguistic components. Second, each component is assigned to a sub-module that solves a single task. Lastly, these modules are then composed into an appropriate layout that predicts an answer for each training example. Such a dynamic network not only helps learning object-object relationships well via compositionally, but also improves the reasoning abilities of the model.

Multitask Learning: There have been number of works that explore multitask learning as an approach to joint learning of vision and language tasks. In one such work Johnson, Gupta, and Fei-Fei, 2018, authors learn related regions of the image by simultaneously training three different semantic tasks - scene graph generation, object detection, and image captioning. A multi-task learning architecture was also proposed by W. Zhao et al., 2018 for image captioning where they enable sharing of a CNN encoder and an LSTM decoder between object classification task and the syntax generation tasks. Ruder, 2017; Y. Lin et al., 2018 show mutlitask learning reduces overfitting in limited-resource settings, and can learn representations to improve downstream (part-of-speech tagging and name-entity recognition) tasks. Our purpose of joint training in multitask learning is to provide regularization on the learned features for VQA, with an added benefit of achieving better performance on the auxiliary task (of generating captions).

Incorporating additional knowledge: In Chandu, Pyreddy, et al., 2018 authors show that incorporating captions helps resolve some ambiguities in visual question answering. In Aditya, Y. Yang, and Baral, 2018 authors first obtain captions and then use them for improving VQA via the framework of predicate logic. In Q. Wu et al., 2016 authors learn attributes from an image using an image labeling and then query using an external knowledge base.

5.3 JUPITER - Justification by Pointwise combination of Image and Text based on Expected Rewards

The key motivation of this approach [depicted in Figure: 5.1] was to manipulating the loss function to account for captions. We hyothesize that explicitly accounting for captions in the loss function will affect the downstream VQA predictions. Figure 5.1 shows the framework architecture and functioning.

Model Description

Our model uses Neural Module Networks (NMN), along with multiple proposed extensions. More specifically, we use the following extensions:

- *Multitask Learning*: We modify the decoder to perform multiple tasks namely, caption generation and VQA. We use the attention grid generated by *'Find'* module in the NMN, the encoded question layout, and the input image to generate captions in an auto regressive. Our hypothesis is that using this conditioning, we can force the model to generate attention grid that is suitable to both downstream tasks, in turn improving VQA performance.

- *Conditional Generation*: As opposed to multitask learning approach, in this extension we explicitly provide the generated captions as input to VQA decoder. More specifically, we train the model to first generate a relevant image caption using previously defined setup. Next we condition the answer decoder on the generated caption. The intuition is that providing the model with information more explicitly will help to predict answers based on this information.

- *Re-weighting*: In this extension, we re-weight the answer hypothesis using the generated caption. We hypothesize that this will help the model to disambiguate between answer logits that have maximum entropy.

- *M-Hybrid and C-Hybrid*: In order to harvest complimentary benefits from our primary extensions, we also implemented two hybrid systems. M-Hybrid extension combined multitask learning and re-weighting approach, and the C-Hybrid extension combined conditional generation and re-weighting approach.

- *Reinforcement Learning*: This extension uses Reward Augmented Maximum Likelihood (RAML) as opposed to Maximum Likelihood (MLE) for generating captions. The intuition for this extension was to enable the agent to generate captions that will help the model to answer the given question. More specifically, the agent at each caption generation step can perform one of the two tasks: (1) Generate next word for the captions or (2) Answer the question based on caption generated so far. The agent is rewarded based on VQA accuracy. Since training with REINFORCE is known to be unstable, we use a baseline wherein we generate answers based on the final hidden state of a deocder trained using MLE.

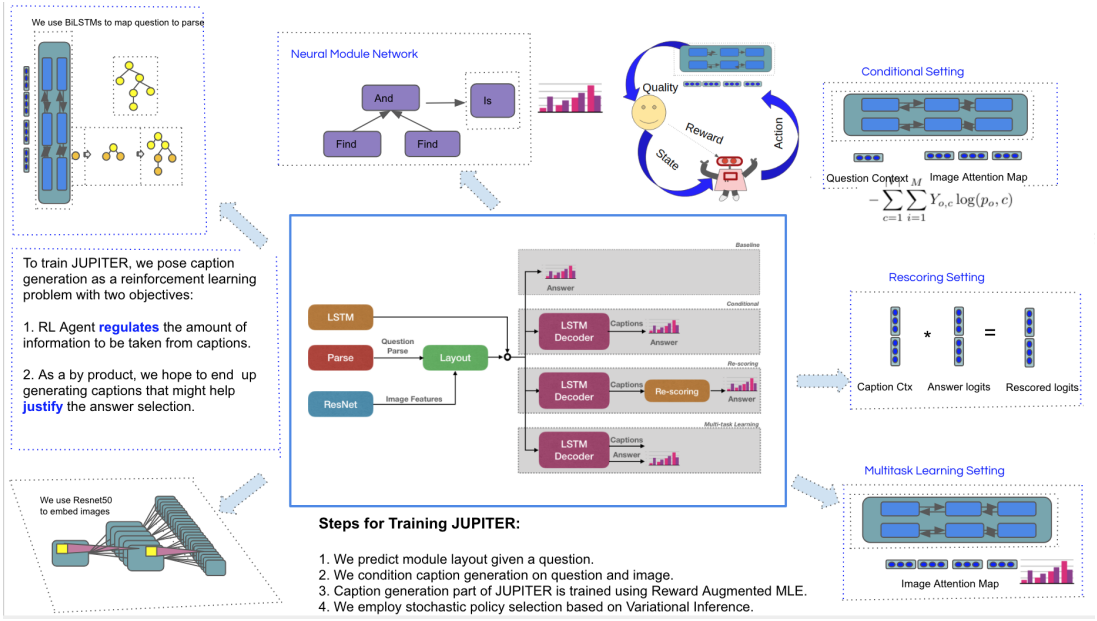


Figure 5.1 Justification by Pointwise combination of Image and Text based on Expected Rewards

Learning

We denote input question as Q and input image as I. L^* denotes the gold layout for Q and C^* is gold caption for I. We denote L as the layout generated by NMN for Q. C is caption generated from JUPITER. We denote answer classes by y and the correct answer class by y^*. T is the training data samples of type (I, Q, y^*). Next, we describe the objective function for each extension in detail.

- *Multitask Learning*: We use a two-part objective function for multitask learning. The first part is generating captions from the input and the second is generating answer logits from the input and the generated NMN layout.

$$L(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y|I,Q,L) + logP_\theta(C|I,Q) \tag{5.1}$$

- *Conditional Generation*: This extension uses a similar objective function. However, we generate answer logits from the input, generated NMN layout as well as the generated captions.

$$L(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y|I,Q,L,C) + logP_\theta(C|I,Q) \tag{5.2}$$

- *Re-weighting*: This extension uses a similar objective as conditioned generation. Further, for re-weighting we define new answer logits y'.

$$y' = C_T y \tag{5.3}$$

where, C_T is the final hidden state of generated caption, and y is the previous answer logits. The updated objective function is:

$$L(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y'|I,Q,L,C) + logP_\theta(C|I,Q) \tag{5.4}$$

- *Reinforcement Learning*: The agent transitions between generating next word in the caption and generating final answer. The agent receives minibatch VQA accuracy as

its reward. The Baseline we use to stabilize the training and the expected reward of our agent respectively are expressed as

$$L_{baseline}(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y|I,Q,L,C) \tag{5.5}$$

We use cross-entropy loss to train the model. We jointly train our captions module in JUPITER alongside NMN, which learns a question layout L.

5.4 Dataset and Input Modalities

VQA dataset by Antol et al., 2015 has 265016 images, 614163 questions. The dataset consists of 82,783 training, 40,504 validation, and 40,775 test images. Each image has 3 questions on average and 10 ground truth answers. Questions as well as answers are open ended, accounting for a more real-world scenario. The questions are rich in a way, as the require the model to have complex reasoning and understanding abilities.

5.5 Results and Discussion

In this section, we discuss the results from our proposed approaches viz. JUPITER, VENUS and MARS, and compare them against our baselines. Table 5.1 consolidates the results of our experiments. To better understand the performance of these models, we report the performance across different answer categories namely, Number, Yes/No and Other. The overall best baseline model for VQA is NMN by Hu et al., 2017.

sectionResults: Baseline Models The input to our baseline models is the image and the question. We do not use any external knowledge. Our results show that the baseline models have highest accuracy on the Yes/No questions. However, the Number type questions often require deeper understanding of the image, and so our baselines have lowest performance on them. Humans tend to have low agreement for Yes/No questions. We attribute this

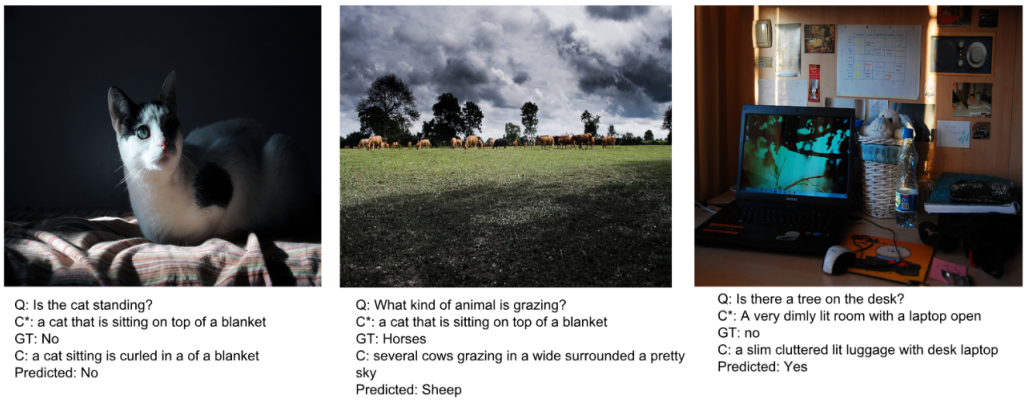


Figure 5.2 Qualitative Analysis from JUPITER: left image depicts a scenario where generating caption helped the model in selection of the right answer. Image in the center depicts a scenario where captions end up confusing the model. Image in the right most highlights an interesting scenario where the generated caption seems irrelevant.

to question ambiguity or missing information in the image.It has to be noted that our implementation of the NMN baseline achieves better scores compared to the open source original implementation. This can be attributed to the presence of additional modules in our implementation, specifically OR, COUNT, FILTER, and EXIST modules.

sectionResults: Proposed Models Looking at the objective evaluation results from table 5.1, it is clear that incorporating captions leads to improvements across the approaches. This result empirically validates our hypothesis related to captions: Captions help VQA. To understand the extent of this, we have also performed ablation analysis wherein we have used just captions to answer the question ignoring the input image. Surprisingly, systems built in this fashion seem to perform better than our baselines. This leads to an interesting observation: *Captions seem to contain supplementary and in some cases complementary information to the images themselves.* However, we acknowledge that proving such hypothesis would require additional experimentation. For instance, it would be interesting to perform similar ablation analyses employing computationally more powerful frameworks such as attention as baselines or adding more visual information such as ground truth bounding boxes. It is also interesting to note that the proposed approaches achieve better scores compared

Model	System	Input	Number	Yes/No	Other
Human	Best	Image + Question	83.39	95.77	72.67
Human	Worst	Image + Question	65.28	46.52	78.02
NMN	Baseline (Replicated)	Image + Question	23.31	63.93	26.65
NMN	Baseline (Our implementation)	Image + Question	26.35	64.49	31.55
RNN	Baseline	Image + Question	19.34	57.82	17.77
VED	Baseline	Image + Question	17.76	58.00	10.43
RNN	MARS	Image + Question + Caption*	23.09	57.88	18.22
RNN	MARS	Question + Caption*	21.72	57.95	21.59
VED	VENUS	Image + Question + Caption*	19.25	57.83	10.10
VED	VENUS	Question + Caption*	18.13	58.10	10.33
NMN	M-Hybrid	Image + Question + Caption	26.31	64.27	30.43
NMN	C-Hybrid	Image + Question + Caption	27.48	65.8	32.2
NMN	JUPITER	Image + Question + Caption	32.82	67.95	33.15

Table 5.1 Results from human, baselines and proposed approaches. * denotes systems that employ Gold captions

against the *worst* human performance in Yes/No category.

Our approach JUPITER outperforms all other approaches across all the categories. In addition, within the models employing module networks, the system employing reinforcement learning outperforms other approaches. This is in line with our hypothesis related to Reward Augmented Maximum Likelihood and raises interesting questions related to *comparison between supervised approaches such as Maximum Likelihood and their reward based reinforcement counterparts*. It would be interesting to perform a much larger scale evaluation comprehensively comparing the effectiveness of these approaches in the context of downstream tasks. In figure 5.2, we present some scenarios that highlight the way captions get utilized for answering question about the corresponding images.

Chapter Six

Conclusion

In this thesis, I present an argument for De-Entanglement: a property that has potential to isolate the factors of variation in the data distribution. I am interested in knowing if explicitly isolating relevant factors using such an approach is helpful with respect to downstream tasks. I first highlight three different approaches to accomplish 'De-Entanglement'. I then present one case study per approach to investigate the importance of such an approach. I conclude by arguing that while this serves as a neat framework to build systems, such an approach might not always be applicable or necessary.

Bibliography

Acosta, Jaime C and Nigel G Ward (2011). "Achieving rapport with turn-by-turn, user-responsive emotional coloring". In: *Speech Communication* 53.9-10, pp. 1137–1148.

Aditya, Somak, Yezhou Yang, and Chitta Baral (2018). "Explicit Reasoning over End-to-End Neural Architectures for Visual Question Answering". In: *arXiv preprint arXiv:1803.08896*.

Agrawal, Aishwarya, Dhruv Batra, et al. (2017). "Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering". In: *CoRR* abs/1712.00377. arXiv: 1712.00377. URL: http://arxiv.org/abs/1712.00377.

Agrawal, Aishwarya et al. (2017a). "C-vqa: A compositional split of the visual question answering (vqa) v1. 0 dataset". In: *arXiv preprint arXiv:1704.08243*.

— (2017b). "C-VQA: A Compositional Split of the Visual Question Answering (VQA) v1.0 Dataset". In: *CoRR* abs/1704.08243. arXiv: 1704.08243. URL: http://arxiv.org/abs/1704.08243.

Agrima, Abdellah et al. (2017). "Detection of Negative Emotion Using Acoustic Cues and Machine Learning Algorithms in Moroccan Dialect". In: *International Conference on Soft Computing and Pattern Recognition*. Springer, pp. 100–110.

Andreas, Jacob et al. (2015). "Deep compositional question answering with neural module networks. arXiv preprint". In: *arXiv preprint arXiv:1511.02799* 2.

— (2016). "Neural module networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 39–48.

Ansari, Abdul Fatir and Harold Soh (2018). "Hyperprior Induced Unsupervised Disentanglement of Latent Representations". In: *arXiv preprint arXiv:1809.04497*.

Antol, Stanislaw et al. (2015). "VQA: Visual Question Answering". In: *CoRR* abs/1505.00468. arXiv: 1505.00468. URL: http://arxiv.org/abs/1505.00468.

Aytar, Yusuf, Carl Vondrick, and Antonio Torralba (2016). "Soundnet: Learning sound representations from unlabeled video". In: *Advances in Neural Information Processing Systems*, pp. 892–900.

Baby, Arun. "Resources for Indian languages". In: *CBBLR workshop, International Conference on Text, Speech and Dialogue,2016*.

Badino, Leonardo, Claudia Barolo, and Silvia Quazza (2004). "Language independent phoneme mapping for foreign TTS". In: *Fifth ISCA Workshop on Speech Synthesis*.

Badino, Leonardo, Claudia Canevari, et al. (2014). "An auto-encoder based approach to unsupervised learning of subword units". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7634–7638.

Baljiker, Pallavi, Sai Krishna Rallabandi, and Alan Black (2018). "An Investigation of Convolution Attention Based Models for Multilingual Speech Synthesis of Indian Languages". In: *Proceedings of Interspeech*.

Biadsy, Fadi et al. (2019). "Parrotron: An End-to-End Speech-to-Speech Conversion Model and its Applications to Hearing-Impaired Speech and Speech Separation". In: *arXiv preprint arXiv:1904.04169*.

Black, Alan W (2006). "CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling." In: *INTERSPEECH*.

Black, Alan et al. (1998). *The festival speech synthesis system*.

Bowman, Samuel R et al. (2015). "Generating sentences from a continuous space". In: *arXiv preprint arXiv:1511.06349*.

Burgess, Christopher P et al. (2018a). "Understanding disentangling in Beta VAE". In: *arXiv preprint arXiv:1804.03599*.

Burgess, Christopher P et al. (2018b). "Understanding disentangling in beta-VAE". In: *arXiv preprint arXiv:1804.03599*.

Campbell, Nick (2001). "Talking foreign". In: *Proc. Eurospeech*, pp. 337–340.

Cao, Yuewen et al. (2019). "End-to-end Code-switched TTS with Mix of Monolingual Recordings". In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Carvalho, Alex de et al. (2019). "Prosody and function words cue the acquisition of word meanings in 18-month-old infants". In: *Psychological science* 30.3, pp. 319–332.

Chandu, Khyathi Raghavi, Mary Arpita Pyreddy, et al. (2018). "Textually Enriched Neural Module Networks for Visual Question Answering". In: *arXiv preprint arXiv:1809.08697*.

Chandu, Khyathi Raghavi, Sai Krishna Rallabandi, et al. (2017). "Speech Synthesis for Mixed-Language Navigation Instructions". In: *Proc. Interspeech 2017*, pp. 57–61.

Chen, Tian Qi et al. (2018). "Isolating sources of disentanglement in variational autoencoders". In: *Advances in Neural Information Processing Systems*, pp. 2615–2625.

Chen, Xi et al. (2016). "Variational lossy autoencoder". In: *arXiv preprint arXiv:1611.02731*.

Chorowski, Jan et al. (2019). "Unsupervised speech representation learning using WaveNet autoencoders". In: *arXiv preprint arXiv:1901.08810*.

Deng, Yuntian et al. (2018). "Latent alignment and variational attention". In: *Advances in Neural Information Processing Systems*.

Ebbers, Janek et al. (2017). "Hidden Markov Model Variational Autoencoder for Acoustic Unit Discovery." In: *INTERSPEECH*, pp. 488–492.

Elluru, Naresh Kumar et al. (2013). "Is word-to-phone mapping better than phone-phone mapping for handling English words?" In: *ACL (2)*, pp. 196–200.

Esmaeili, Babak et al. (2018). "Structured disentangled representations". In: *stat* 1050, p. 12.

Ewan, Dunbar et al. (2019). "ZeroSpeech 2019: TTS without T". In: *Interspeech 2019*, pp. 3442–3446.

Eyben, Florian, Martin Wöllmer, and Björn Schuller (2010). "Opensmile: the munich versatile and fast open-source audio feature extractor". In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM, pp. 1459–1462.

Fukui, Akira et al. (2016). "Multimodal compact bilinear pooling for visual question answering and visual grounding". In: *arXiv preprint arXiv:1606.01847*.

Garcia-Romero, Daniel and Carol Y Espy-Wilson (2011). "Analysis of i-vector length normalization in speaker recognition systems". In: *Twelfth Annual Conference of the International Speech Communication Association*.

Gauntlett, David (2014). "The LEGO System as a tool for thinking, creativity, and changing the world". In: *LEGO Studies: Examining the Building Blocks of a Transmedial Phenomenon*, pp. 1–16.

Gella, Spandana, Kalika Bali, and Monojit Choudhury (2014). *"ye word kis lang ka hai bhai?" Testing the Limits of Word level Language Identification*.

Gibiansky, Andrew et al. (2017). "Deep voice 2: Multi-speaker neural text-to-speech". In: *Advances in neural information processing systems*, pp. 2962–2970.

He, Kaiming et al. (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

Higgins, Irina et al. (2016). "beta-vae: Learning basic visual concepts with a constrained variational framework". In:

Hsu, Wei-Ning et al. (2018). "Hierarchical generative modeling for controllable speech synthesis". In: *arXiv preprint arXiv:1810.07217*.

Hu, Ronghang et al. (2017). "Learning to Reason: End-to-End Module Networks for Visual Question Answering". In: *CoRR* abs/1704.05526. arXiv: 1704.05526. URL: http://arxiv.org/abs/1704.05526.

Huijbregts, Marijn, Mitchell McLaren, and David Van Leeuwen (2011). "Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection". In: *2011 IEEE*

international conference on Acoustics, speech and signal processing (ICASSP). IEEE, pp. 4436–4439.

Ito, Keith et al. (2017). *The LJSpeech dataset*.

Jansen, Aren, Samuel Thomas, and Hynek Hermansky (2013). "Weak top-down constraints for unsupervised acoustic model training". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 8091–8095.

Jia, Ye et al. (2018). "Transfer learning from speaker verification to multispeaker text-to-speech synthesis". In: *Advances in Neural Information Processing Systems*, pp. 4480–4490.

Johnson, Justin, Agrim Gupta, and Li Fei-Fei (2018). "Image Generation from Scene Graphs". In: *CoRR* abs/1804.01622. arXiv: 1804.01622. URL: http://arxiv.org/abs/1804.01622.

Kain, Alexander and Mike Macon (1998). "Personalizing a speech synthesizer by voice adaptation". In: *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

Kazemi, Vahid and Ali Elqursh (2017). "Show, Ask, Attend, and Answer: A Strong Baseline For Visual Question Answering". In: *CoRR* abs/1704.03162. arXiv: 1704.03162. URL: http://arxiv.org/abs/1704.03162.

Keller, SC et al. (2014). "A single low-energy, iron-poor supernova as the source of metals in the star SMSS J031300. 36- 670839.3". In: *Nature* 506.7489, p. 463.

Kingma, Diederik P and Max Welling (2013a). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

— (2013b). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*.

Kurimo, Mikko et al. (2010). "Personalising speech-to-speech translation in the EMIME project". In: *Proceedings of the ACL 2010 System Demonstrations*. Association for Computational Linguistics, pp. 48–53.

Latorre, Javier, Koji Iwano, and Sadaoki Furui (2005). "Polyglot synthesis using a mixture of monolingual corpora". In: *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*. Vol. 1. IEEE, pp. I–1.

Li, Yuanpeng et al. (2018). "Zero-Shot Transfer VQA Dataset". In: *arXiv preprint arXiv:1811.00692*.

Lin, Ying et al. (2018). "A multi-lingual multi-task architecture for low-resource sequence labeling". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 799–809.

Locatello, Francesco et al. (2018). "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *arXiv preprint arXiv:1811.12359*.

Lu, Jiasen, Xiao Lin, et al. (2015). *Deeper lstm and normalized cnn visual question answering model*.

Lu, Jiasen, Jianwei Yang, et al. (2016). "Hierarchical question-image co-attention for visual question answering". In: *Advances In Neural Information Processing Systems*, pp. 289–297.

Makhzani, Alireza and Brendan J Frey (2017). "PixelGAN autoencoders". In: *Advances in Neural Information Processing Systems*, pp. 1975–1985.

Morningstar, Michele et al. (2019). "Changes in parental prosody mediate effect of parent-training intervention on infant language production." In: *Journal of consulting and clinical psychology* 87.3, p. 313.

Muysken, Pieter (2000). *Bilingual speech: A typology of code-mixing*. Vol. 11. Cambridge University Press.

Norouzi, Mohammad et al. (2016). "Reward augmented maximum likelihood for neural structured prediction". In: *Advances In Neural Information Processing Systems*.

Oord, Aaron van den, Oriol Vinyals, et al. (2017a). "Neural discrete representation learning". In: *Advances in Neural Information Processing Systems*, pp. 6306–6315.

— (2017b). "Neural discrete representation learning". In: *Advances in Neural Information Processing Systems*, pp. 6306–6315.

Oura, Keiichiro et al. (2010). "Unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis". In: *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2010*. IEEE, pp. 4594–4597.

Parlikar, Alok (2012a). "TestVox: web-based framework for subjective evaluation of speech synthesis". In: *Opensource Software*.

— (2012b). "TestVox: web-based framework for subjective evaluation of speech synthesis". In: *Opensource Software*.

Paszke, Adam et al. (2019). "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems*, pp. 8024–8035.

Ping, Wei, Kainan Peng, and Jitong Chen (2018). "Clarinet: Parallel wave generation in end-to-end text-to-speech". In: *arXiv preprint arXiv:1807.07281*.

Povey, Daniel et al. (2011). "The Kaldi speech recognition toolkit". In: *IEEE 2011 workshop on automatic speech recognition and understanding*. EPFL-CONF-192584. IEEE Signal Processing Society.

Prahallad, Kishore et al. (2014). "The Blizzard Challenge 2014". In: *Proc. Blizzard Challenge workshop*. Vol. 2014.

Prenger, Ryan, Rafael Valle, and Bryan Catanzaro (2018). "WaveGlow: A Flow-based Generative Network for Speech Synthesis". In: *arXiv preprint arXiv:1811.00002*.

Qian, Yao, Ji Xu, and Frank K Soong (2011). "A frame mapping based HMM approach to cross-lingual voice transformation". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011*. IEEE, pp. 5120–5123.

Rallabandi, Sai Krishna and Alan Black (2019). "Variational Attention using Articulatory Priors for generating Code Mixed Speech using Monolingual Corpora". In: *in proceedings of Interspeech*.

Rallabandi, SaiKrishna and Alan W Black (2017). "On building mixed lingual speech synthesis systems". In: *Proceedings of Interspeech 2017*, pp. 52–56.

Ravanelli, Mirco and Yoshua Bengio (2018). "Interpretable convolutional filters with SincNet". In: *arXiv preprint arXiv:1811.09725*.

Rosenberg, Andrew (2010). "Autobi-a tool for automatic tobi annotation". In: *Eleventh Annual Conference of the International Speech Communication Association*.

Rousseau, David and Sotirios Tsaftaris (2019). "Data augmentation techniques for deep learning". In: *Tutorial Session, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Ruder, Sebastian (2017). "An overview of multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1706.05098*.

Sakti, Sakriani, Eka Kelana, et al. (2008). "Development of Indonesian large vocabulary continuous speech recognition system within A-STAR project". In: *Proceedings of the Workshop on Technologies and Corpora for Asia-Pacific Speech Translation (TCAST)*.

Sakti, Sakriani, R Maia, et al. (2008). "Development of HMM-based Indonesian speech synthesis". In: *Proc. Oriental COCOSDA*, pp. 215–219.

Salimans, Tim et al. (2017). "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications". In: *arXiv preprint arXiv:1701.05517*.

Schuller, Björn, Stefan Steidl, Anton Batliner, Felix Burkhardt, et al. (2010). "The INTERSPEECH 2010 paralinguistic challenge". In: *Proc. INTERSPEECH 2010, Makuhari, Japan*, pp. 2794–2797.

Schuller, Björn, Stefan Steidl, Anton Batliner, Simone Hantke, et al. (2015). "The INTERSPEECH 2015 computational paralinguistics challenge: nativeness, parkinson's & eating condition". In: *Sixteenth Annual Conference of the International Speech Communication Association*.

Schuller, Björn, Stefan Steidl, Anton Batliner, Peter Marschik, et al. (2018). "The INTERSPEECH 2018 Computational Paralinguistics Challenge: Atypical & Self-Assessed Affect, Crying & Heart Beats". In: *INTERSPEECH 2018 – 18th Annual Conference of*

the International Speech Communication Association, September 2-6, Hyderabad, India, Proceedings.

Schuller, Björn, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, et al. (2013). "The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism". In: *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France*.

Schuller, Björn et al. (2017). "The interspeech 2017 computational paralinguistics challenge: Addressee, cold & snoring". In: *Computational Paralinguistics Challenge (ComParE), Interspeech 2017*, pp. 3442–3446.

— (2018). "The INTERSPEECH 2018 Computational Paralinguistics Challenge Atypical and Self-Assessed Affect, Crying and Heart Beats". In: *Computational Paralinguistics Challenge (ComParE), Interspeech 2017*, pp. 3442–3446.

Shen, Jonathan et al. (2017). "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions". In: *arXiv preprint arXiv:1712.05884*.

Silver, David et al. (2017). "Mastering the game of go without human knowledge". In: *Nature* 550.7676, p. 354.

Sitaram, Sunayana and Alan W Black (2016). "Speech Synthesis of Code-Mixed Text." In: *LREC*.

Sitaram, Sunayana, Sukhada Palkar, et al. (2013). "Bootstrapping text-to-speech for speech processing in languages without an orthography". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 7992–7996.

Sitaram, Sunayana, Sai Krishna Rallabandi, et al. (2015). "Experiments with cross-lingual systems for synthesis of code-mixed text". In: *9th ISCA Speech Synthesis Workshop*, pp. 76–81.

Skerry-Ryan, RJ et al. (2018). "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron". In: *arXiv preprint arXiv:1803.09047*.

Skolem, Th (1955). "Peano's axioms and models of arithmetic". In: *Studies in Logic and the Foundations of Mathematics*. Vol. 16. Elsevier, pp. 1–14.

Smith, Leslie N (2017). "Cyclical learning rates for training neural networks". In: *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, pp. 464–472.

Strubell, Emma et al. (2017). "Fast and accurate entity recognition with iterated dilated convolutions". In:

Sundermann, David et al. (2006). "Text-independent voice conversion based on unit selection". In: *IEEE International Conference on Acoustics, Speech and Signal Processing,* vol. 1. IEEE, pp. I–I.

Tishby, Naftali, Fernando C Pereira, and William Bialek (2000). "The information bottleneck method". In: *arXiv preprint physics/0004057*.

Tomokiyo, Laura Mayfield, Alan W Black, and Kevin A Lenzo (2005). "Foreign accents in synthetic speech: development and evaluation." In: *Interspeech*, pp. 1469–1472.

Traber, Christof et al. (1999). "From multilingual to polyglot speech synthesis". In: *Sixth European Conference on Speech Communication and Technology*.

Trigeorgis, George et al. (2016). "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network". In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, pp. 5200–5204.

Van Den Oord, Aaron et al. (2016). "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499*.

Van Den Oord, Aäron et al. (2016). "Wavenet: A generative model for raw audio". In: *CoRR abs/1609.03499*.

Verhelst, Werner and Marc Roelands (1993). "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech". In: *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*. Vol. 2. IEEE, pp. 554–557.

Vyas, Nidhi et al. (2019). "Learning Disentangled Representation in Latent Stochastic Models: A Case Study with Image Captioning". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Wan, Vincent et al. (2019). "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network". In: *arXiv preprint arXiv:1905.07195*.

Wang, Yuxuan et al. (2018). "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis". In: *arXiv preprint arXiv:1803.09017*.

Wang, Yuxuan, RJ Skerry-Ryan, Daisy Stanton, et al. (2017). "Tacotron: Towards end-to-end speech synthesis". In: *arXiv preprint arXiv:1703.10135*.

Wang, Yuxuan, RJ Skerry-Ryan, Ying Xiao, et al. (2017). "Uncovering latent style factors for expressive speech synthesis". In: *arXiv preprint arXiv:1711.00520*.

Ward, Nigel G (2019). *Prosodic patterns in English conversation*. Cambridge University Press.

Wood, Sarah G et al. (2018). "Does use of text-to-speech and related read-aloud tools improve reading comprehension for students with reading disabilities? A meta-analysis". In: *Journal of learning disabilities* 51.1, pp. 73–84.

Wu, Qi et al. (2016). "Ask me anything: Free-form visual question answering based on knowledge from external sources". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4622–4630.

Xue, Liumeng et al. (2019). "Building a mixed-lingual neural TTS system with only monolingual data". In: *arXiv preprint arXiv:1904.06063*.

Yamagishi, Junichi et al. (2009). "Robust speaker-adaptive HMM-based text-to-speech synthesis". In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.6, pp. 1208–1230.

Yang, Zichao et al. (2016). "Stacked attention networks for image question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 21–29.

Zhao, Tiancheng, Ran Zhao, and Maxine Eskenazi (2017). "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders". In: *arXiv preprint arXiv:1703.10960*.

Zhao, Wei et al. (2018). "A Multi-task Learning Approach for Image Captioning." In: *IJCAI*, pp. 1205–1211.

Zhou, Chunting and Graham Neubig (2017). "Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction". In: *arXiv preprint arXiv:1704.01691*.