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 X

Chapter One

Introduction

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(**gauntlett2014lego**). 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(**stellar_evolution**). 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(**peano_axioms**). But we as humans have been conditioned to solve this task by cumulative addition of individual digits with appropriate carryover and not necessarily following (**peano_axioms**). Similarly consider the inner workings of AlphaZero(**alpha_zero_withouthumans**). 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?

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 (**isolating_sources_betavae**; **understanding_disentanglement_betavae**; **structured_disentang**
hyperprior_disentanglement). Note that although obtaining such degenerate representations is considered typical, it is not the only manifestation: it also manifests as continuous representations(**ravanelli2018interpretablesyncnet**) and other abstract phenomena(e.g. grounding). I argue that explicitly controlling what and how much gets de-entangled (**understanding_disentanglement_betavae**) is better than implicit disentanglement as is followed today(**locatello2018challenging**). 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(**alexnet**). 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(**tutorial_dataaugmentation**) across modalities and tasks. It can be shown that the pooling layer acts as information bottleneck(**tishby2000information**) module in such architectures. I point out(**variational_attention_** 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 (**variational_attention_rsk**; **vyas2019learning**) 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 (**isolating_sources_betavae**; **understanding_disentanglement_betavae**; **structured_disentan** **hyperprior_disentanglement**). Note that although obtaining such degenerate representations is considered typical, it is not the only manifestation: it also manifests as continuous representations(**ravanelli2018interpretablesyncnet**) and other abstract phenomena(e.g. grounding). I argue that explicitly controlling what and how much gets de-entangled (**understanding_disentanglement_betavae**) is better than implicit disentanglement as is followed today(**locatello2018challenging**). 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

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

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 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(**tacotron_original**) 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(**latentalignment_variationalattention**) 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(**latentalignment_variationalattention**). 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 (**pixelgan_autoencoder**). 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 (**lossy_vae**). 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, (**beta_analysis**) 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 instance, 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 by Priors - 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

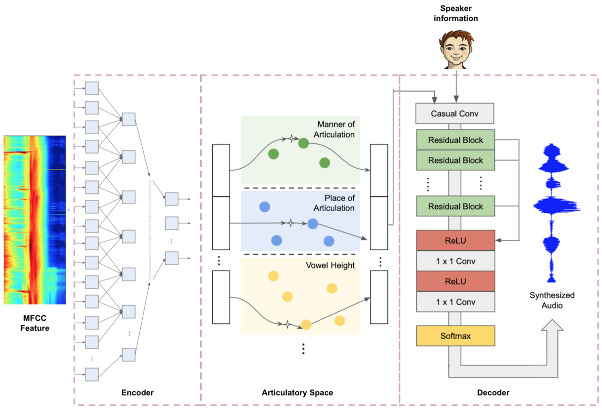


Figure 3.1 Illustration of our procedure for automatically discovering acoustic units from a speech utterance. We pass the speech utterance through a downsampling encoder. The encoded representation is hashed to a latent code based on a discrete articulatory prior bank. The code is passed to the decoder,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

discovering the appropriate acoustic phonetic units.

In ZeroSpeech Challenge(**zerospeech2019**) 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 (**shen2017natural**). 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 here-after). 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 (**vae**). When deployed in encoder-decoder models, this approach is often subject to an optimization challenge referred to as KL-collapse (**bowman_continuous**), wherein the generator (usually an RNN) marginalizes the learnt latent representation. Typical approaches to dealing this issue involve annealing the KL divergence loss (**bowman_continuous**; **zhou2017multi**), weakening the generator (**zhao2017learning**) 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 (**vq-vae**; **chorowski2019unsupervised**), 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 $\mathbf{U}$ 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 (**subword_diarization**), authors presented an approach to modify the speaker diarization system to detect speaker-dependent acoustic units. (**unsupervised_AMtraining_ArenJans**
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_autoencoder**) 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. (**hmm-vae_bhiksha**) extended the structured VAE to incorporate the Hidden Markov Models as latent model. (**vq-vae**; **chorowski2019unsupervised**) 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 (**shen2017natural**). 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 (**salimans2017pixelcnn**++), 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 (**beta_vae**) 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 (**lossy_vae**), 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 (**sakti2008development1**; **sakti2008development2**; **zerospeech2019**). 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 (**sitaram2013bootstrapping**), 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 (**tacotron_transferlearning2multispeaker**) 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 ${}_j d(e_j, z_e(x))$, 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.

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**
[]		

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 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.

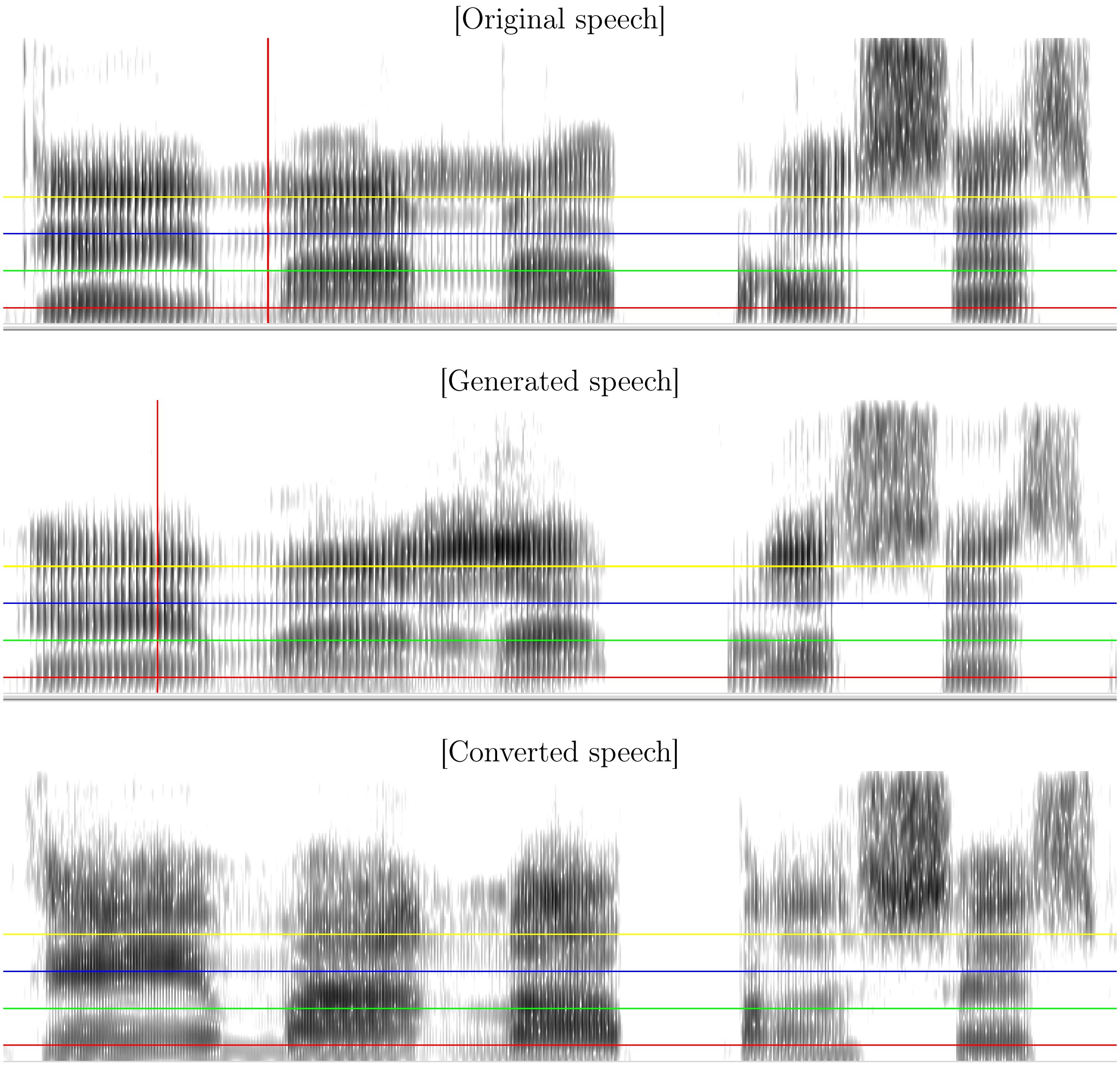


Figure 3.2 Spectrograms of original, generated, and converted speech. The source speaker is female while the target speaker is male.

Chapter Four

De-Entanglement by additional Divergences - 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(**nigel_ward_prosodic_patterns**). Further, such manipulation in prosody has been shown effective in applications such as Infant Behavior Programs (**parental_prosody_changes_mediate_infant_language_production**), improving language acquisition(**prosody_functionwords_acquisition**) and promoting rapport(**rapport_dialogs** It seems natural to employ generative models of speech(**tacotron_original**; **deepvoice2**; **clarinet**; **parrotron**) to assist in such scenarios (**does_tts_help_comprehension**). However, although there has been tremendous progress in the neural generative models for speech in the context of vocoder fidelity(**waveglow**; **wavenet_original**), the notion of controllability in such models is not yet fully evolved. While there have been works towards models aimed at controlling prosody(**tacotron_hierarchical**; **tacotron_stylefacoruncovering**), the exerted control is still global or coarse grained in terms of styles of speech(**tacotron_prosodycontrol**; **tacotron_styletokens**), 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

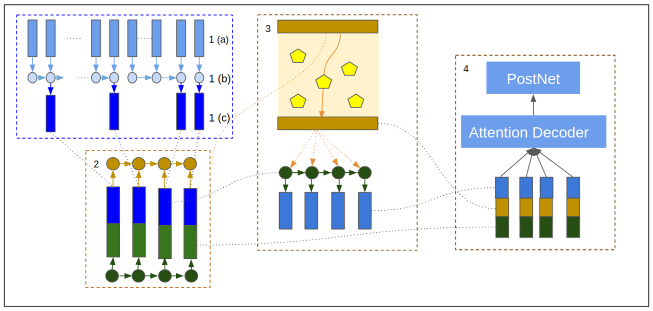


Figure 4.1 *Architecture of EDITH. Circles denote LSTM cells, rectangles represent vectors and pentagons represent global latent vectors. (Best viewed in color)*

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 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 (**chive**). 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 (**vqvae**) 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(**tacotron_original**) and includes a postnet. While similar in formulation, EDITH has an important difference from (**chive**) in that our local latent variables follow the rate of input as opposed to output as in (**chive**). 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 **vqvae** for both z_g and z_l. This can be expressed as below:

$$\begin{aligned} L = \lambda_{linear} \sum_{t=0}^{To} \| y_{linear}^t - \hat{y}_{linear}^t \| \\ + \lambda_{mel} \sum_{t=0}^{To} \| y_{mel}^t - \hat{y}_{mel}^t \| \\ + \lambda_{qF_0} \sum_{t=0}^{Ti} Div(qF_0, q\hat{F_0}) + \lambda_e L_e + L_{VQ} \end{aligned} \tag{4.2}$$

sectionModel Interpretation This approach can be interpreted as VQVAE(**vqvae**). It can also be seen as GST(**tacotron_styletokens**) 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 Networks**neural_module_networks** In **chive** 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(**ljspeech_dataset**) 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 Tacotron**tacotron_original** Seq2Seq speech synthesis system is built using PyTorch(**pytorch**). 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 **tacotron_original** 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(**rosenberg2010autobi**). 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.

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

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 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 (**testvox_parlikar**). 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

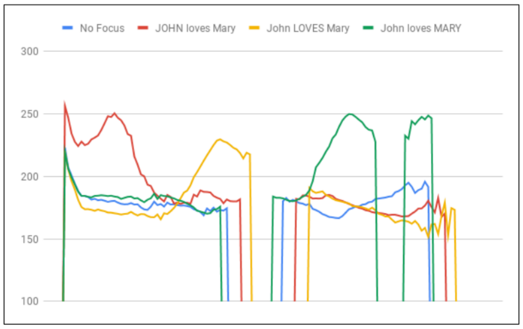


Figure 4.2 *Plot of Fundamental Frequency(F_0) trajectories obtained from generated waves using proposed approach FUE. Variants of the sentence 'John loves Mary' are generated with emphasis on individual words(captialized). The blue trajectory corresponds to F_0 when no emphasis was applied to any word. 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.*

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 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 utterance 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

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

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

sectionVisual 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 **fukui2016multimodal lu2016hierarchical yang2016stacked lu2015deeper** claim to learn good representations of abstract concepts needed to answer questions. However, it has been shown **agrawal2017c** 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 **HuARDS17** This is motivated from our hypothesis that generating captions can provide additional information to improve VQA. Additionally, JUPITER uses Reward Augmented Maximum Likelihood **RAML** which is improves caption generation.

sectionRelated Work **Visual Question Answering**: **KazemiE17** 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. **AishAgrawal17** introduced grounding to prevent the model from memorizing this bias. Similarly, **li2018zero** used a zero-shot training approach to improve the generalizabilty of the model, and prevent the model to learn the bias. However, recently **AgrawalKBP17** 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 **HuARDS17** and **deepmodulenets** 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 modular 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 **JustinJohnson2018** 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 **zhao2018multi** for image captioning where they enable sharing of a CNN encoder and an LSTM decoder between object classification task and the syntax generation tasks. **ruder2017overview**; **lin2018multi** 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 **chandu2018textually** authors show that incorporating captions helps resolve some ambiguities in visual question answering. In **aditya2018explicit** authors first obtain captions and then use them for improving VQA via the framework of predicate logic. In **wu2016ask** authors learn attributes from an image using an image labeling and then query using an external knowledge base.

sectionJUPITER - 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:

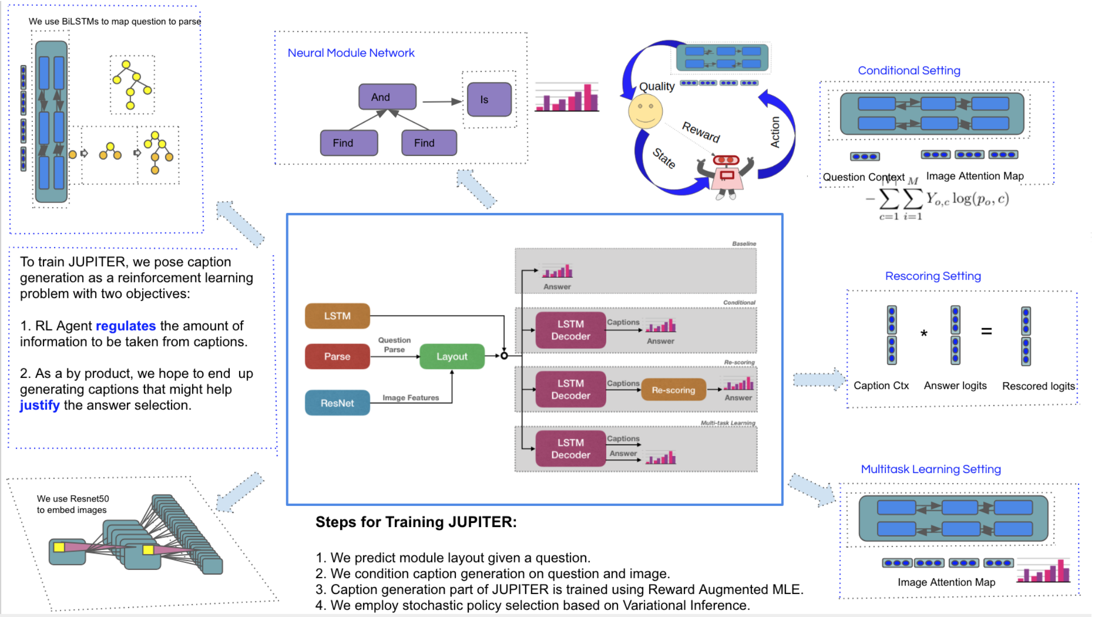
- *Multitask Learning*: We modify the decoder to perform multiple tasks namely, caption generation and VQA. We use the attention grid generated by *'Find'* module in the NMN, the encoded question layout, and the input image to generate captions in an auto regressive. Our hypothesis is that using this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



To train JUPITER, we pose caption generation as a reinforcement learning problem with two objectives:

1. RL Agent **regulates** the amount of information to be taken from captions.

2. As a by product, we hope to end up generating captions that might help **justify** the answer selection.

Steps for Training JUPITER:

1. We predict module layout given a question.
2. We condition caption generation on question and image.
3. Caption generation part of JUPITER is trained using Reward Augmented MLE.
4. We employ stochastic policy selection based on Variational Inference.

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

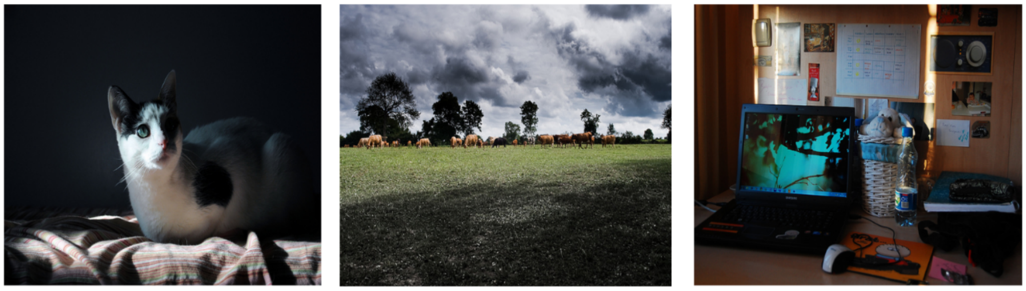
$$L_{baseline}(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y|I,Q,L,C) \tag{5.5}$$

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*.

sectionDataset and Input Modalities VQA dataset by **AntolALMBZP15** 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.

5.1 Results and Discussion



Q: Is the cat standing?
C*: a cat that is sitting on top of a blanket
GT: No
C: a cat sitting is curled in a of a blanket
Predicted: No

Q: What kind of animal is grazing?
C*: a cat that is sitting on top of a blanket
GT: Horses
C: several cows grazing in a wide surrounded a pretty sky
Predicted: Sheep

Q: Is there a tree on the desk?
C*: A very dimly lit room with a laptop open
GT: no
C: a slim cluttered lit luggage with desk laptop
Predicted: Yes

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.

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 **HuARDS17**

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 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.

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

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 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.