CHiVE: Varying Prosody in Speech Synthesis with a Linguistically Driven Dynamic Hierarchical Conditional Variational Network

Vincent Wan [1] **Chun-an Chan** [1] **Tom Kenter** [1] **Jakub Vit** [2] **Rob Clark** [1]

Abstract

The prosodic aspects of speech signals produced by current text-to-speech systems are typically averaged over training material, and as such lack the variety and liveliness found in natural speech. To avoid monotony and averaged prosody contours, it is desirable to have a way of modeling the variation in the prosodic aspects of speech, so audio signals can be synthesized in multiple ways for a given text. We present a new, hierarchically structured conditional variational auto-encoder to generate prosodic features (fundamental frequency, energy and duration) suitable for use with a vocoder or a generative model like WaveNet. At inference time, an embedding representing the prosody of a sentence may be sampled from the variational layer to allow for prosodic variation. To efficiently capture the hierarchical nature of the linguistic input (words, syllables and phones), both the encoder and decoder parts of the auto-encoder are hierarchical, in line with the linguistic structure, with layers being clocked dynamically at the respective rates. We show in our experiments that our dynamic hierarchical network outperforms a non-hierarchical state-of-the-art baseline, and, additionally, that prosody transfer across sentences is possible by employing the prosody embedding of one sentence to generate the speech signal of another.

1. Introduction

Most current text-to-speech (TTS) prosody modeling paradigms implicitly assume a one-to-one mapping between text and prosody and fail to recognize the one-to-many nature of the task, i.e., there is a large number of ways in

[1] TTS Research, Google UK, London [2] University of West Bohemia, work carried out whilst at Google. Correspondence to: Tom Kenter <tomkenter@google.com>.
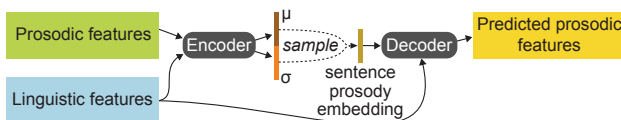
Figure 1. High-level overview of the CHiVE model at training time.

which a given sequence of words can be spoken. This leads to an averaging effect and as such a lack of variation and variety in generated synthetic speech. This work aims to model prosody in a way that avoids this problem and works towards allowing mechanisms to control the variation in the generated prosody in a linguistically motivated way.

Prosody prediction in TTS is usually concerned with providing segment durations and fundamental frequency (F_0) contours for the utterance being synthesized. These targets are used to guide unit selection (Fujii et al., 2003), form the input features to a vocoder (Yoshimura et al., 1999), or drive a WaveNet-like model (Oord et al., 2016). The way in which prosody models are traditionally built has a number of inherent problems. Firstly, as stated above, there is an underlying assumption that there is a unique mapping from a given text or linguistic specification to a prosodic realization. Secondly, duration is modeled independently of F_0, e.g., (Zen et al., 2016). Lastly, modeling techniques are generally frame- or phone-based and as such, lack the ability to ensure linguistically valid prosodic contours for an utterance as a whole.

In this paper we propose a new model to explicitly address the problems described above. We present a model called a Clockwork Hierarchical Variational autoEncoder (CHiVE), that produces F_0 and duration and, additionally, c_0 (the 0th cepstrum coefficient) as energy, which is a strong prosodic correlate. In our work, these predicted prosodic features are used to drive a WaveNet model (Oord et al., 2016) to produce the final speech signal.

The CHiVE model is trained as a conditional variational auto-encoder (Kingma & Welling, 2014; Sohn et al., 2015). A high-level graphical overview is provided in Figure 1. The encoder has both linguistic features (a matrix represented by a blue rectangle) and prosodic features (green

rectangle) as input, and encodes these into an embedding using a variational bottleneck layer. The linguistic features represent aspects of the input such as part-of-speech for words, syllable information and phone-level information. The prosodic features represent acoustic information about the input, in terms of pitch, duration and energy. The variational layer predicts two vectors, which are interpreted as means and variances (the brown and orange vectors in Figure 1 respectively), used to parameterize an isotropic Gaussian distribution, from which an embedding is sampled. The decoder has linguistic features as input (the same ones the encoder has), and is additionally fed an embedding sampled from the variational layer. The sampled embedding, in this way, is designed to encode prosodic aspects of the input sentence. We refer to it as the *sentence prosody embedding*.

To capture the linguistic hierarchy incorporated in the input features – provided at the level of words, syllables, phones and frames – both the encoder and the decoder are hierarchical, with dynamically clocked layers following the layers in the linguistic structure of an utterance. A unique feature of the CHiVE mdoel is that the structure of the network is dynamic, i.e., the unrolled recurrent structure of the model is different for every input sentence, depending on the linguistic structure of that sentence. The model is described in full in §3. We show in our experiments that our dynamic hierarchical network outperforms a non-hierarchical baseline.

The main value of CHiVE lies in the distribution learned by the variational layer, which represents a space of valid prosodic specifications. At inference time, two different scenarios can be employed to make use of this space.

The first scenario is where we discard the encoder, and choose or sample a sentence prosody embedding from the Gaussian prior, trained at the variational layer. The zero vector can be chosen, which, by design, is the mean of the sentence prosody embedding space, and in our experience tends to represent prosody which exhibits the averaging effect as seen in models such as (Zen et al., 2016).

The second scenario is similar to the setup at training time, where a sentence is encoded by the encoder. Yet, instead of trying to reproduce the prosodic features of the input sentence as in the auto-encoder setting during training, we can use the predicted sentence prosody embedding to generate speech for a different sentence, thereby transferring the prosody of one sentence to another. In this setting, the decoder gets the linguistic features of the new sentence as input, while it is conditioned on a sentence prosody embedding of a reference sentence. This way, the speech for the textual content of the new sentence will be synthesized with the prosody of the reference sentence.

In short, our contributions are:

- We present CHiVE, a linguistically driven dynamic hierarchical conditional variational auto-encoder, the

first of its kind to have a dynamic network layout that depends on the linguistic structure of its input.

- We show that the CHiVE model yields a meaningful prosodic space, by showing that prosodic features sampled from it yield synthesized audio superior to the audio based on features sampled from a non-hierarchical variational baseline.

- We illustrate prosody transfer, where the prosody of a reference sentence, as captured by the variational layer in the CHiVE model, can be used to synthesize speech for another sentence.

2. Related Work

We discuss work related to three aspects of the CHiVE model we present: modelling of prosody, hierarchcial models and variational models.

2.1. Modeling prosody

Early parametric approaches to TTS using hidden Markov models (HMMs) adopted a "bucket of Gaussians" approach to duration modeling (Yoshimura et al., 1998) which easily fits within the HMM architecture. This could be considered a step backwards from earlier approaches, for example: (Black et al., 1998; Van Santen, 1994; Goubanova & King, 2008) Additionaly, in these models F_0 is considered as just another acoustic feature, i.e., a local feature of a context-dependent phone, rather than a supra-segmental property of the utterance; earlier work including (Fujisaki & Hirose, 1984; Black & Hunt, 1996; Vainio et al., 1999) was ignored. This trend continued with the move to deep neural networks (Zen et al., 2013; Zen & Sak, 2015). There are a few exceptions. In (Mixdorff & Jokisch, 2001) a feed forward neural network is presented to predict prosody using a Fujisaki-style model, but they only apply it to re-synthesis rather than TTS; In (Tokuda et al., 2016), an integrated model is proposed using a hidden semi-Markov model within a neural network framework. In contrast to the early work described above work, our approach models the primary acoustic prosodic features used to realize prosody jointly, and across the full sequence.

An alternative approach to modeling prosody explicitly is taken by end-to-end approaches to TTS, for example, (Ping et al., 2017; Wang et al., 2017) where there is no explicit prosody model at all, and any modeling of prosody is performed implicitly within the model. To a large extent these end-to-end based approaches appear to be able to generate natural prosody and expression, however, in their current forms it is not clear how to obtain varied prosodic patterns. Recent work with style tokens (Wang et al., 2018) attempts to address this point, but the style that can be controlled so far is quite abstract. Style tokens may guide utterance level

prosodic features such as emotion, pitch range, or speaking-rate, but the representations learned by the tokens do not necessarily represent clear useful styles. In short, the task style tokens are designed for – capturing a particular style across many utterances – is different from the one we address here: varying the intonational aspects of prosody per utterance.

2.2. Hierarchical models

Hierarchical models are popular in natural language processing, e.g., to encode sentence embeddings from the sequence of words in them (Serban et al., 2017; Kenter & de Rijke, 2017), or to construct sentence representation from a sequence of words, which are in turn treated as sequences of characters, or bytes (Jozefowicz et al., 2016; Kenter et al., 2018). In TTS, hierarchical networks have been explored to a limited extent: (Ronanki et al., 2017) use hierarchical information by having recurrent connections at frame and phone levels. In other work, (Ribeiro et al., 2016; Yin et al., 2016) try to integrate hierarchical information, but in the context of feed-forward neural networks instead of recurrent neural networks (RNNs).

Most similar to our approach are the clockwork models such as (Koutnik et al., 2014), from which we derive the name. Different from that work, the clockwork model we present is run as a conditional variational auto-encoder, rather than as a standard encoder-decoder model. Lastly, (Liu et al., 2015) have a set of fixed clocking speeds that are independent of the input sequence and act within sub-portions of a layer of RNN cells. In contrast to that work, we adopt dynamic clock rates across layers by utilizing sequence-to-sequence encoding techniques.

2.3. Variational models

There are few recent attempts to use VAE-style models in speech synthesis. For example (Akuzawa et al., 2018) propose VAE-Loop where the latent space is used to model expression of speech, and (Henter et al., 2018) use a number of VAE related techniques to show that categories of acted emotion can be modeled in an unsupervised fashion. (Hsu et al., 2018) present an extended Tacotron approach combining a variational approach with a basic hierarchical structure to better address separation of latent feature spaces. Again these techniques are concerned with the abstract nature of the utterance prosody such as speaking style rather than being concerned with the details of prosodic meaning.

3. The CHiVE model

The CHiVE model learns a mapping between pairs of $\{x, y\}$, where x is a sequence of input features, and y a sequence of prosodic parameters. The features describing the linguistic aspects of the input are denoted as $x^{linguistic}$.

They are provided in a hierarchical fashion – a tree structure – with features at sentence level, word level, syllable level, phone level and frame level. We denote the features pertaining to the acoustic prosodic information (F_0, c_0 and duration) as $x^{prosodic}$, consisting of $x^{F_0 c_0}$ and $x^{duration}$. The prosodic features are presented at multiple levels as well. Throughout this paper, let $p \in [0, \ldots, P]$ enumerate phones, and $t \in [0, \ldots, T]$ enumerate acoustic frames in sequences x and y. Duration values, $x_p^{duration}$, are provided at phone level, while F_0 and c_0 values, $x_t^{F_0 c_0}$, are provided at frame level. We note that $y = x^{prosodic}$, i.e., as the CHiVE model is a (conditional) auto-encoder, its task is to reconstruct the input features $x^{prosodic}$.

Predicting the two output sequences, $\hat{y}^{duration}$ and $\hat{y}^{F_0 c_0}$, is modeled as a regression task. The predictions are computed from the input features by two hierarchical recurrent neural networks (RNNs), with a variational layer in between:

$$\hat{\mu}, \hat{\sigma} = \mathcal{H}^{encoder}(x^{prosodic}, x^{linguistic})$$
$$\hat{s} \sim \mathcal{N}(\hat{\mu}, \hat{\sigma})$$
$$\hat{y}^{duration}, \hat{y}^{F_0 c_0} = \mathcal{H}^{decoder}(\hat{s}, x^{linguistic}).$$

The hierarchical encoder and decoder, and the way they handle features, are described in more detail in §3.1 and §3.3, respectively. The sentence prosody embedding $\hat{s}$ is sampled from an isotropic multi-dimensional Gaussian distribution, parametrized by predicted mean vector $\hat{\mu}$ and variance vector $\hat{\sigma}$.

The network is optimized to minimize two L_2 losses, plus a KL divergence term for the variational layer:

$$\mathcal{L} = \lambda_1 \sum_p ||y_p^{duration} - \hat{y}_p^{duration}||^2 \quad (1)$$
$$+ \lambda_2 \sum_t ||y_t^{F_0 c_0} - \hat{y}_t^{F_0 c_0}||^2$$
$$+ \lambda_3 D_{KL}[\mathcal{N}(\hat{\mu}, \hat{\sigma}) \, || \, \mathcal{N}(0, 1)],$$

where the first L_2 loss is computed over the durations values per phone, the second L_2 loss is computed over the F_0 and c_0 values per frame, and $\hat{\mu}$ and $\hat{\sigma}$ are the means and variances as predicted by the encoder, respectively. The λ terms are used to weight the three losses.

The CHiVE model consists of three parts: an encoder, a variational layer, and a decoder. We will describe the design of each of these in turn in this section. Additional details about the hyper-parameters of the network and the features used are discussed in §4.1.1.

3.1. CHiVE encoder

The CHiVE encoder is a hierarchical RNN, composed of three RNNs: a frame-rate RNN and a phone-rate RNN, that both produce syllable-rate output, and a syllable-rate RNN
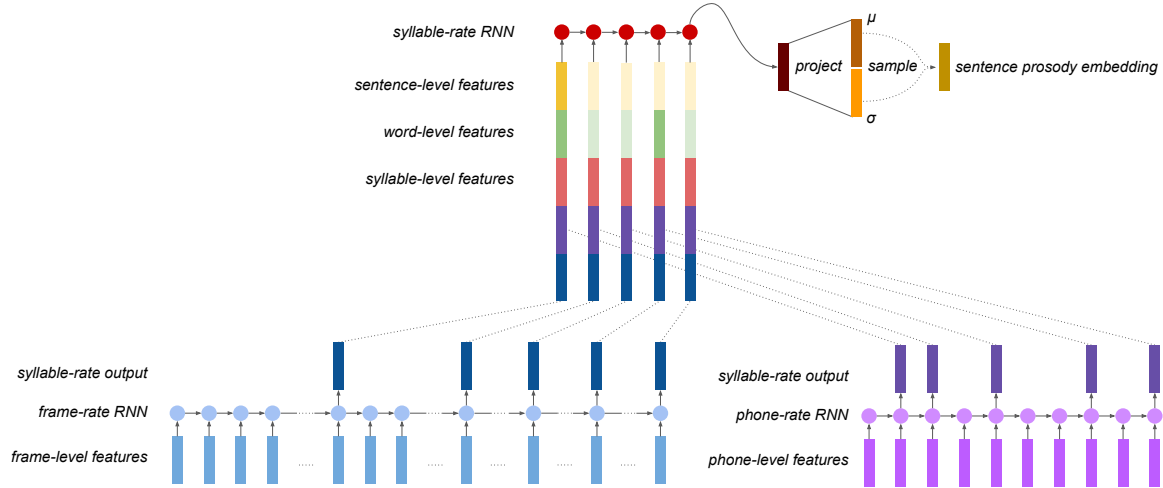
Figure 2. CHiVE encoder and variational layer. Circles represent blocks of RNN cells, rectangles represent vectors. Broadcasting (vector duplication) is indicated by displaying vectors in a lighter shade of the same colour. (Best viewed in colour).

that takes in the output of the other two RNNs, in addition to sentence-, word- and syllable-level linguistic input features. Figure 2 gives a graphical depiction of the CHiVE encoder.

The frame-rate RNN, depicted in blue, reads frame-level prosodic features (F_0 and c_0), and outputs its internal activation at the end of every syllable. There are usually many frames in a syllable – our experiments use a 5 millisecond frame shift – most of which are left out of the figure for clarity. The phone-rate RNN runs asynchronously in parallel, and like the frame-rate RNN emits its hidden state at every syllable boundary. In the figure the first syllable consists of two phones and the second syllable of one phone.

The outputs of the two RNNs are concatenated and joined with additional syllable-, word- and sentence-level input features. The word-level and sentence-level input features are broadcast as appropriate, which is indicated in Figure 2 by them having a slightly lighter shade. As we can see from the figure, there is only one sentence-level feature vector, broadcast to every time step. Similarly, there are two words, the feature vectors of which are broadcast across their respective syllables. Finally, the last hidden activation of the syllable-rate RNN is taken to represent the entire input and passed on to a variational layer.

The hierarchical layout of the model is designed to reflect the hierarchical nature of the linguistic information that the features represent. We should note two things concerning the clock rates of the RNNs. Firstly, the number of input feature vectors for the RNN running at frame-rate is different from the number of input feature vectors for the one running at phone-rate. Nonetheless both RNNs produce the same number of output vectors: one for every syllable. Secondly, the number of input feature vectors at all levels (frames, phones, syllables and words) is different across

sentences. The CHiVE model handles this appropriately, by letting each sub-RNN run at the right clock speed, determined by the number of features provided at that level by the input structure. As such, CHiVE dynamically follows the linguistic structure of the input sentence, which is the main feature setting it apart from other models.

3.2. Variational layer

The variational layer takes the last hidden activation of the syllable-rate RNN (depicted in dark red in Figure 2) as input, passes it through a fully connected layer, and splits the resulting vector into two vectors, representing the mean and variance of an isotropic multi-variate Gaussian (displayed in brown and orange, respectively, in the figure). Finally, a vector is sampled from this Gaussian (displayed in dark yellow), which is used as input to the decoder.

3.3. CHiVE decoder

The CHiVE decoder is similar to the encoder, but the hierarchy is reversed. The decoder goes from sentence-level to the levels of syllables, phones and frames. Figure 3 gives a graphical overview of the decoder.

The sentence prosody embedding output by the variational layer in the previous step (displayed in dark yellow again in the figure) is concatenated with sentence-level (yellow), word-level (green) and syllable-level features (pink), all broadcast as appropriate, and used as input to a syllable-rate RNN that produces one output per syllable. To reiterate, as we are auto-encoding, the linguistic feature vectors here are the same as those input to the encoder.

The next level of the decoder is a phone-rate RNN (in purple), that takes as input each output of the syllable-rate RNN,
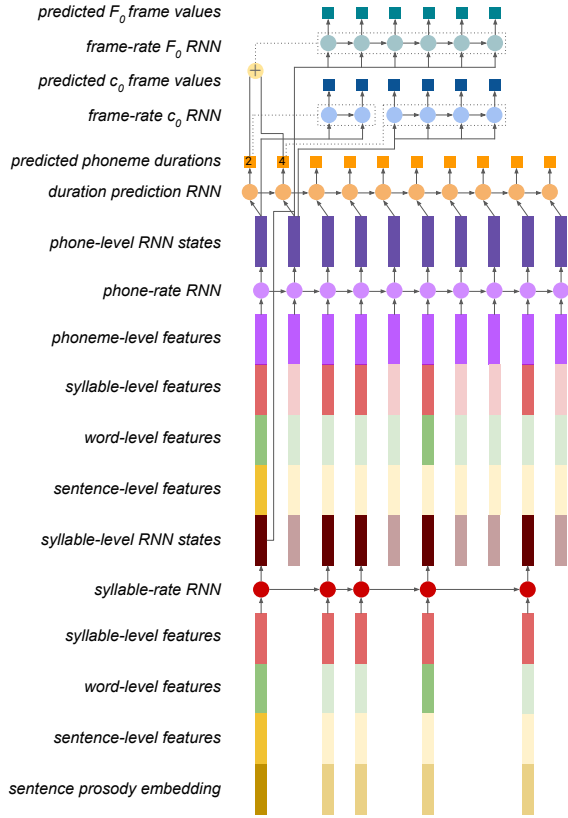
Figure 3. CHiVE decoder. Circles represent blocks of RNN cells, rectangles represent vectors. Broadcasting (vector duplication) is indicated by displaying vectors in a lighter shade of the same colour. (Best viewed in colour).

concatenated with sentence-level features, word-level features, syllable-level features (the same ones that fed into the syllable-level RNN, simply read again at this stage) and phone-level features to produce a sequence of phone representations. The output activations of the phone-rate RNN (in dark purple) are used as input to three different RNNs, that together model the prosodic features, F_0, c_0 and duration, as highlighted in §1.

Firstly, an RNN predicts phone duration (displayed in orange), expressed as the number of frames per phone. In Figure 3 the predicted values for the first two phones are 2 and 4, respectively (these numbers are low to keep the figure clear; in reality these numbers are much higher). Secondly, to predict the c_0 values for each frame, another RNN (in blue) is run for as many steps as predicted by the phone duration RNN, where the corresponding phone-rate RNN output activation is repeatedly fed as input at every time step. In the figure, the c_0 RNNs for the first two phones are displayed, which unroll for 2 time steps and 4 time steps for the first and second phone, respectively. The last state of one RNN is used as initial state for the next. In reality, the c_0

RNN is run for every phone but for reasons of clarity this is not shown in the figure. Finally, the F_0 values are predicted by another frame-rate RNN (in turquoise), for every frame in each syllable. To accomplish this, the durations for all phones in a syllable are summed, and an RNN predicting F_0 values is unrolled for as many steps. So, instead of unrolling this RNN twice – once for 2 steps and once for 4 steps as is the case for the c_0 prediction – the RNN is run once for each syllable, the unroll length of which is the sum of the durations of its constituent phones – 6, for the first syllable in our example in Figure 3. The input at every time step is the syllable-rate RNN activation (dark red), concatenated with the phone-rate RNN activation (dark purple) corresponding to the end of the syllable. In Figure 3 this is the second phone, as the first syllable consists of two phones. Again, the F_0 RNN does run for all other syllables too, yet to keep the figure clear, only one frame-level F_0 RNN is displayed.

With respect to the duration values, it should be noted that during training the *predicted* durations are used for calculating the L_2 loss, but the *ground truth* durations are used to determine the unroll lengths of the frame-level c_0 and F_0 RNNs. This allows for easy calculation of the differences between predicted c_0 and F_0 values, which can be computed by a one-to-one matching of frames. At inference time there is no ground truth duration available, and the predicted duration values are used.

In addition to the features described above, every RNN level is provided with a timing signal, using a standard cosine coarse encoding technique (Zen et al., 2013). Different dimensions of coding signal are used for different hierarchy levels: 64 for words, 4 for syllables and phones, and 3 for frames. These timing vectors can be thought of as being part of the features vectors at every level in the model, and are therefore not explicitly depicted in Figures 2 and 3.

4. Experiments

Our main experiment compares the prosodic features sampled from the distribution yielded by our hierarchical variational model to those sampled from a non-hierarchical baseline model. In addition, we illustrate that CHiVE can be used to transfer the prosody of one sentence to another. Speech samples for the main and additional experiments, and for the prosody transfer illustrations described below are available at `https://google.github.io/chive-varying-prosody-icml-2019/`.

4.1. Sampling prosodic features

As described in §1, the aim of CHiVE is to yield a prosodic space from which suitable features can be sampled, to be used by a generative model producing speech audio. We compare our dynamic hierarchical variational network to

Table 1. Side-by-side comparison between CHiVE and a baseline non-hierarchical model. Results are found to be significant, with a normal approximation to a binomial test z-value=-5.5, and p-value=3.91×10^{-8}.

Baseline preferred	neutral	CHiVE preferred
292 (30.7%)	220 (23.2%)	**438** (46.1%)

Table 2. Results of MOS test comparing CHiVE to the baseline model. The interval shown is 95% confidence around the mean.

	MOS
Baseline based on (Zen et al., 2016)	4.01 ± 0.11
CHiVE - zero embedding	4.07 ± 0.10
CHiVE - encoded embedding	4.25 ± 0.10
Real speech	4.67 ± 0.07

a non-hierarchical variational network, where the features generated by both are used to drive the same generative model to produce speech audio.

4.1.1. EXPERIMENTAL METHODOLOGY

Baseline A non-hierarchical baseline was constructed to mirror the CHiVE model but without the dynamic structure. It was a single multi-layer model running at frame time steps, with broadcasting at different levels, but no dynamic encoding/decoding.

Test set The test set consisted of 100 sentences with 10 prosodically different renditions having been synthesized for each, both for our hierarchical CHiVE model, and for the non-hierarchical baseline. That is, for each sentence we sampled 10 sentence prosody embeddings from each model and run the associated decoder to produce prosodic features, which we then supplied to Parallel WaveNet (Oord et al., 2017) to produce speech. The WaveNet model made use of the duration and F_0 features, but not c_0. We then randomly paired audio samples of the same sentence, one from each model, which gave us 1000 test sample pairs. We performed a side-by-side AB comparison, where raters were asked to indicate which of two audio samples of a pair they thought was better. Each rater received a random selection of pairs to rate and each pair was rated once.

Training and eval set Our training data was recorded by 22 American English speakers in studio conditions, the number of male and female speakers being balanced. The data consists of 161 000 lines, from a range of domains, including jokes, poems, Wikipedia and other web data. The evaluation data for hyperparameter tuning was a set of 100 randomly selected sentences per speaker, held out from training. The input features used include information about phoneme identity, number of syllables, dependency parse tree. We closely followed previous work (Zen & Sak, 2015), with two main differences: 1) a one-hot speaker identification feature was added and 2) features are split into levels corresponding to the levels of the linguistic structure.

Hyperparameters Both the CHiVE prosody model and the non-hierarchical baseline use LSTM cells. The RNNs at each level of the hierarchy consist of two LSTM layers. The variational layer predicts the parameters of a 256-dimensional Gaussian distribution. The hidden layers of the

syllable and phone-level RNNs have dimension 32 at every layer, both in case of the the encoder and the decoder. We use a batch size of 4 for training.

4.1.2. RESULTS

Table 1 lists the results of the side-by-side comparison between our CHiVE model and the non-hierarchical baseline model. The speech generated using the CHiVE prosody model is significantly preferred over the speech generated using the baseline prosody model. We conclude that incorporating the hierarchical structure of the linguistic input, as done in CHiVE, yields better performance than processing the exact same features in a non-hierarchical fashion.

4.2. Further Evaluation

To further evaluate CHiVE's performance we perform mean opinion score (MOS) tests. Additionally, to gain insight into the training phase, we analyse the models in terms of loss.

MOS listening tests To evaluate the quality of the different methods of obtaining sentence level embeddings we performed a series of MOS listening tests. We compare using an all-zero sentence prosody embedding to an embedding made by encoding the ground truth audio. As a top-line we include the ground truth audio itself, and as a baseline we include the model from (Zen et al., 2016) as used by WaveNet (Oord et al., 2017) to represent current state of the art. We choose a test set of 200 expressive sentences including jokes, poems and trivia game prompts as prosody is an important aspect of their rendition and we want to see how well our model is able to capture this. Raters judge each sentence for naturalness.

Results are shown in Table 2 where we see that the zero embedding is judged to be slightly, but not significantly, better than the baseline. The encoded version is better still, which is unsurprising as it is using the auto-encoder properties of the model. The performance here is still lower than that of the real speech. This might be caused by the acoustic information being compressed in the variational layer, the fact that a sentence prosody embedding is sampled from the encoded distribution or highly expressive properties, such as increased energy or vocal fry, are still lost by the waveform
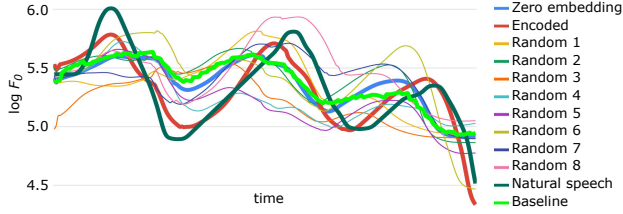
Figure 4. A selection of randomly generated $\log F_0$ contours produced by CHiVE for the sentence "That's a super choice!". Contours do not align perfectly on the time axis because the predicted phone durations differ per contour. (Best viewed in colour).

generation model as it is currently configured to make use only of the predicted F_0 and duration values. We conclude that while using the zero vector to drive prosody generation yields reasonable natural intonation, it may lack the true expressiveness that the model is capable of generating. We investigate this further below.

Variation of a single utterance   Figure 4 shows a range of sampled and reference $\log F_0$ contours for a single sentence. We first note that the natural speech contour and contour generated by CHiVE in VAE mode, labeled 'Encoded', are both very similar and expressive. This shows that the model works well as an encoder-decoder. Secondly we note that the zero-embedding and baseline contours are very similar, and less expressive than the natural speech and (auto)encoded speech. This demonstrates the averaging effect of the same text spoken in different ways—the peaks representing pitch events are broader and less distinct. Finally we see that the randomly sampled sentence prosody embeddings in general produce more expressive speech, and produce a wide range of different prosodic patterns, with prominence in different places. While we have not specifically evaluated the naturalness of these examples, anecdotal evidence suggests none of them sound unnatural.

Performance during training   We present RMSE and absolute error for our hierarchical CHiVE model and the non-hierarchical baseline used in our main experiment, described in section 4.1.1. The results presented are calculated on a held-out evaluation set. As noted earlier – when discussing Equation 1, §3 – the L_2 losses are computed over F_0, c_0 and duration. A breakdown of the overall loss is shown in Table 3. For the line marked 'enc.', the sampling step is skipped and the mean values, $\boldsymbol{\mu}$, predicted by the encoder are directly given as input to the decoder. That is, we specifically evaluate both models' ability to predict suitable durations and F_0 contour given the full information of the held-out example being evaluated.

From Table 3, we see a 21% relative reduction in the $\log F_0$ RMSE compared to the baseline when the mean values predicted by the encoder are used. We also see that the absolute

Table 3. Loss terms of the non-hierarchical baseline used in main experiment (lines labelled Bsl) and CHiVE, when an embedding sampled from the encoder is used (enc.), a zero embedding (0s), or a random embedding (rnd.). Both absolute error (Abs.) and root mean squared error (RMSE) are shown to allow comparison to other work.

	$\log F_0$	F_0	c_0	Duration
	RMSE	Abs. (Hz)	RMSE	RMSE/Abs. (ms)
Bsl enc.	0.098	11.18	0.272	0.378 / 0.019
Bsl 0s	0.193	21.52	0.274	0.656 / 0.034
Bsl rnd.	0.238	26.77	0.277	0.715 / 0.037
CHiVE enc.	0.077	8.478	0.287	0.356 / 0.019
CHiVE 0s	0.173	19.16	0.294	0.515 / 0.027
CHiVE rnd.	0.214	24.37	0.298	0.585 / 0.031

F_0 loss both when using the zero embedding and random embedding is substantially higher than when the encoded embedding is used. This is expected as these represent renditions of the utterance that are averaged or random, and as such may end up being quite different to the held-out example.

The ratio between the F_0 and c_0 in the loss was not tuned, which might explain why the distinctive trend in the F_0 seems to be reversed for the c_0 terms, albeit less pronounced. Interestingly, the loss when zero embeddings are used is lower than when random embeddings are used. This is in line with the intuition that zero embeddings represent an average, overall suitable, prosody, rather than an arbitrary point in the prosodic space.

4.3. Prosody transfer between sentences

During training the encoder/decoder portions of the CHiVE model encode/decode the same text and speech. At inference time, however, it is possible to encode one utterance to a sentence prosody embedding and then use that embedding to condition a decoder that generates prosodic features for an entirely different sentence. When CHiVE is run in this fashion, the sentence prosody embedding – displayed in dark yellow at the bottom in Figure 3 – represents the prosody of a reference sentence that was input to the encoder, while the sentence-, word-, syllable- and phone-level features – displayed in yellow, green, pink and purple respectively – represent another, new sentence, for the decoder to predict the prosodic features for. As a result, the prosody of the new sentence is guided by the prosody of the reference sentence.

To illustrate the prosody transfer capabilities of CHiVE, we show $\log F_0$ curves for two pairs of sentences. We use jokes in our examples, as they have a particular prosody pattern, clearly visible in the figures. Note, though, that this is done for illustrative purposes only, and that prosody transfer is not restricted in any way to a particular type of sentence.
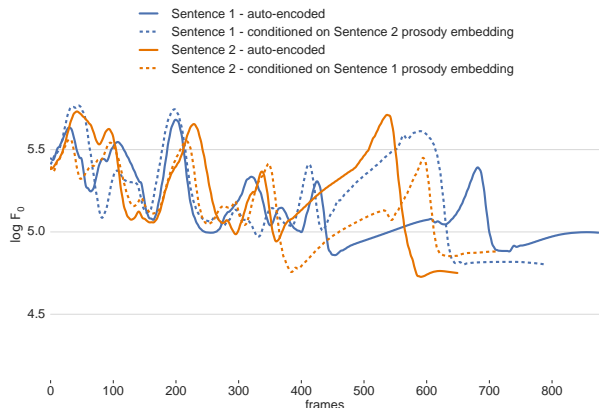
Figure 5. The $\log F_0$ curves for Sentence 1 *"What do you call a boomerang that doesn't come back? A stick!"* and Sentence 2 *"What's orange and sounds like a parrot? A carrot."* conditioned on their own sentence prosody embedding (auto-encoded) and each other's sentence prosody embedding.
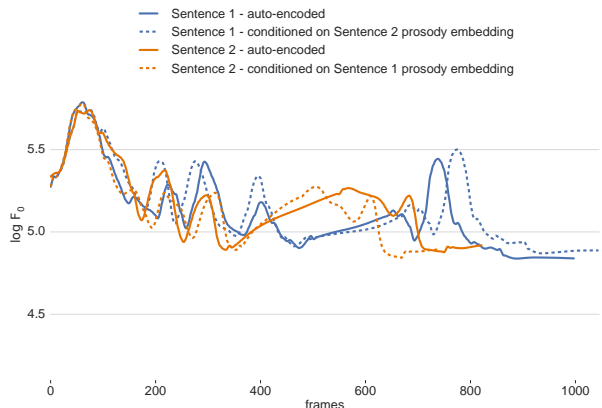
Figure 6. The $\log F_0$ curves for Sentence 1 *"What do you get when you put a vest on an alligator? An investigator."* and Sentence 2 *"What do you get from a pampered cow? Spoiled milk."* conditioned on their own sentence prosody embedding (auto-encoded) and each other's sentence prosody embedding.

Figures 5 and 6 both show the $\log F_0$ curves of two sentences. Two curves are shown for each sentence: the solid lines are the result of running CHiVE in the auto-encoder setting; the dashed lines are the result of conditioning the decoder for one sentence on the sentence prosody embedding of the other.

If no prosody transfer would take place, the pair of curves per sentence – the two blue lines, and the two orange lines in Figures 5 and 6 – would overlap. However, as can be observed from the figures, the opposite happens, where the dashed lines follow the solid lines of different colour.

In Figure 5, the dashed orange line, instead of closely following the solid orange line, is guided by the solid blue line as well. It clearly mimmicks the solid blue line near the end, where the punch line is delivered. The two curves do not perfectly overlap as there is a mismatch in terms of lengths (in words and syllables), which indicates that, even if the two sentences do not perfectly align in terms of syllable structure, meaningful prosody transfer can take place.

In Figure 6 we see a similar effect, where at the start, both the solid blue/dashed orange curves and the solid orange/dashed blue curves are closely tied. The transfer is not perfect, as can be seen, e.g., near the end, where the two blue line ends up much like each other, without, apparently, following the guidance of the reference prosody embedding, with something similar happening to the orange curves.

As these examples illustrate, the variational CHiVE model can be used to transfer prosody from one sentence to another. Additional research is needed for a more rigorous analysis, to see what aspects affect succes or failure of transfer, and if transfer can be extended more generally to, e.g., speaking style. Initial investigations with texts containing the same number of words suggests that specific prosodic patterns transfer between utterances, even when the number of syllables per word is different, and that prominence transfers to the correct syllable in the target sentence.

5. Conclusion

Our model, CHiVE, a linguistically driven dynamic hierarchical conditional variational auto-encoder model, meets the objective of yielding a prosodic space from which meaningful prosodic features can be sampled to generate a variety of valid prosodic contours. We showed that the prosody generated by the model, when using the mean of the distribution modeled in the variational layer (the zero embedding) performs as well as current state of the art. We also showed that the hierarchical structure yields speech with a higher variety in natural pitch contours for a given text when compared to a model without this hierarchy. Lastly, we illustrated that the sentence embedding produced by the encoder, that captures the prosody of one utterance, can be used to transfer the prosody to another text.

CHiVE has the ability to capture natural prosodic variations originally not inferable from the linguistic features it gets as input. This opens opportunities for speech synthesis systems to sound more natural, particularly when synthesizing longer pieces of text – e.g., when a digital assistant reads out a long answer or an entire audiobook – as in these settings repeated use of a default prosody can become tiring to listen to.

As the space captured by our hierarchical variational model is continuous – arbitrary points yield natural sounding prosody – it would be natural to want to control this space, e.g., by manually selecting a point in it rather than random sampling. Future work will concentrate on making the utterance embedding space interpretable in a meaningful way.

Acknowledgments

References

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. Expressive speech synthesis via modeling expressions with variational autoencoder. In *Proceedings of Interspeech 2018*, pp. 3067–3071, 2018.

Black, A. and Hunt, A. Generating F0 contours from ToBI labels using linear regression. In *Proceedings of ICSLP '96*, volume 3, pp. 1385–1388, 11 1996. ISBN 0-7803-3555-4.

Black, A., Taylor, P., Caley, R., and Clark, R. The Festival speech synthesis system, 1998.

Fujii, K., Kashioka, H., and Campbell, N. Target cost of F0 based on polynomial regression in concatenative speech synthesis. In *Proceedings of the 15th International Congress of Phonetic Sciences*, pp. 2577–2580, 2003.

Fujisaki, H. and Hirose, K. Analysis of voice fundamental frequency contours for declarative sentences of Japanese. *Journal of the Acoustical Society of Japan (E)*, 5(4):233–242, 1984.

Goubanova, O. and King, S. Bayesian networks for phone duration prediction. *Speech communication*, 50(4):301–311, 2008.

Henter, G. E., Lorenzo-Trueba, J., Wang, X., and Yamagishi, J. Deep encoder-decoder models for unsupervised learning of controllable speech synthesis. *arXiv preprint arXiv:1807.11470*, 2018.

Hsu, W.-N., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Cao, Y., Jia, Y., Chen, Z., Shen, J., et al. Hierarchical generative modeling for controllable speech synthesis. *arXiv preprint arXiv:1810.07217*, 2018. Forthcoming at ICLR 2019.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Kenter, T. and de Rijke, M. Attentive memory networks: Efficient machine reading for conversational search. In *Proceedings of the 1st International Workshop on Conversational Approaches to Information Retrieval (CAIR'17) at the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, 2017.

Kenter, T., Jones, L., and Hewlett, D. Byte-level machine reading across morphologically varied languages. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 5820–5827, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Koutnik, J., Greff, K., Gomez, F., and Schmidhuber, J. A clockwork RNN. In *International Conference on Machine Learning*, pp. 1863–1871, 2014.

Liu, P., Qiu, X., Chen, X., Wu, S., and Huang, X. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 2326–2335, 2015.

Mixdorff, H. and Jokisch, O. Building an integrated prosodic model of German. In *Seventh European Conference on Speech Communication and Technology*, 2001.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. WaveNet: A generative model for raw audio. *arXiv preprint*, abs/1609.03499, 2016.

Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al. Parallel WaveNet: Fast high-fidelity speech synthesis. *arXiv preprint*, abs/1711.10433, 2017.

Ping, W., Peng, K., Gibiansky, A., Arik, S. Ö., Kannan, A., Narang, S., Raiman, J., and Miller, J. Deep Voice 3: 2000-speaker neural text-to-speech. *arXiv preprint*, abs/1710.07654, 2017.

Ribeiro, M. S., Watts, O., and Yamagishi, J. Parallel and cascaded deep neural networks for text-to-speech synthesis. In *Proceedings of SSW*, pp. 107–112, 2016.

Ronanki, S., Watts, O., and King, S. A hierarchical encoder-decoder model for statistical parametric speech synthesis. In *Proceedings of Interspeech 2017*, pp. 1133–1137, 2017.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C., and Bengio, Y. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pp. 3295–3301, 2017.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28*, pp. 3483–3491, 2015.

Tokuda, K., Hashimoto, K., Oura, K., and Nankaku, Y. Temporal modeling in neural network based statistical parametric speech synthesis. In *Proceedings of ISCA Speech Synthesis Workshop*, pp. 113–118, 2016.

Vainio, M., Altosaar, T., Karjalainen, M., Aulanko, R., and Werner, S. Neural network models for Finnish prosody. In *Proceedings of ICPhS*, pp. 2347–2350, 1999.

Van Santen, J. P. Assignment of segmental duration in text-to-speech synthesis. *Computer Speech & Language*, 8(2): 95–128, 1994.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. Tacotron: Towards end-to-end speech synthesis. In *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

Wang, Y., Stanton, D., Zhang, Y., Ryan, R.-S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., and Saurous, R. A. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5180–5189. PMLR, 2018.

Yin, X., Lei, M., Qian, Y., Soong, F. K., He, L., Ling, Z.-H., and Dai, L.-R. Modeling f0 trajectories in hierarchically structured deep neural networks. *Speech Communication*, 76:82–92, 2016.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. Duration modeling for HMM-based speech synthesis. In *ICSLP*, volume 98, pp. 29–32, 1998.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Sixth European Conference on Speech Communication and Technology*, 1999.

Zen, H. and Sak, H. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 4470–4474. IEEE, 2015.

Zen, H., Senior, A., and Schuster, M. Statistical parametric speech synthesis using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 7962–7966. IEEE, 2013.

Zen, H., Agiomyrgiannakis, Y., Egberts, N., Henderson, F., and Szczepaniak, P. Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*, pp. 2273–2277, 2016.