# TOWARDS GENERATIVE LATENT VARIABLE MODELS FOR SPEECH

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

While stochastic latent variable models (LVMs) now achieve state-of-the-art performance on natural image generation, they are still inferior to deterministic models on speech. On natural images, these models have been parameterised with very deep hierarchies of latent variables, but research shows that these model constructs are not directly applicable to sequence data. In this paper, we benchmark popular temporal LVMs against state-of-the-art deterministic models on speech. We report the likelihood, which is a much used metric in the image domain but rarely, and often incomparably, reported for speech models. This is prerequisite work needed for the research community to improve LVMs on speech. We adapt Clockwork VAE, a state-of-the-art temporal LVM for video generation, to the speech domain, similar to how WaveNet adapted PixelCNN from images to speech. Despite being autoregressive only in latent space, we find that the Clockwork VAE outperforms previous LVMs and reduces the gap to deterministic models by using a hierarchy of latent variables.

## 1 INTRODUCTION

With the introduction of the variational autoencoder (VAE) (Kingma & Welling, 2014; Rezende et al., 2014) came two rapid extensions for modeling speech data (Chung et al., 2015; Fraccaro et al., 2016). Since then, little has happened on the development of temporal LVMs and their autoregressive counterparts, such as the WaveNet (Oord et al., 2016a), remain state-of-the-art. Looking to the image domain, generative LVMs have recently shown superior performance to the PixelCNN (Oord et al., 2016c;b; Salimans et al., 2017), the model that built the foundation for WaveNet. The improvements in performance have primarily been driven by altered inference models, including top-down (Sønderby et al., 2016) and bidirectional inference (Maaløe et al., 2019), deeper latent hierarchies and skip connections (Sønderby et al., 2016; Maaløe et al., 2019; Vahdat & Kautz, 2020; Child, 2021).

To innovate and compare new LVM frameworks we must ensure good baselining, similar to the many reported benchmarks on public datasets within the image domain. However, research in the speech domain has often omitted reporting a likelihood (Oord et al., 2016a; Hsu et al., 2017; Oord et al., 2018b) or has reported likelihood estimates that are incomparable due to subtle differences in the model parameterizations (Chung et al., 2015; Fraccaro et al., 2016; Hsu et al., 2017). Without a proper comparison standard, it is difficult for the field of explicit likelihood models on speech to evolve further.

This research pushes forward the state of the LVM on speech by (i) properly benchmarking previous models, (ii) introducing a high-performing hierarchical temporal LVM architecture, and (iii) analyzing the representations of the latent variables. We find that:

(I) Previous state-of-the-art LVMs achieve close to identical likelihood performance, still significantly inferior to the WaveNet. Interestingly, we also find that the WaveNet performs almost identically to a standard LSTM parameterization (Hochreiter & Schmidhuber, 1997) but surprisingly worse than the lossless compression algorithm FLAC.

(II) Similar to conclusions within image modeling (Maaløe et al., 2019; Vahdat & Kautz, 2020; Child, 2021), the LVM expressiveness increases by a deeper hierarchy of stochastic latent

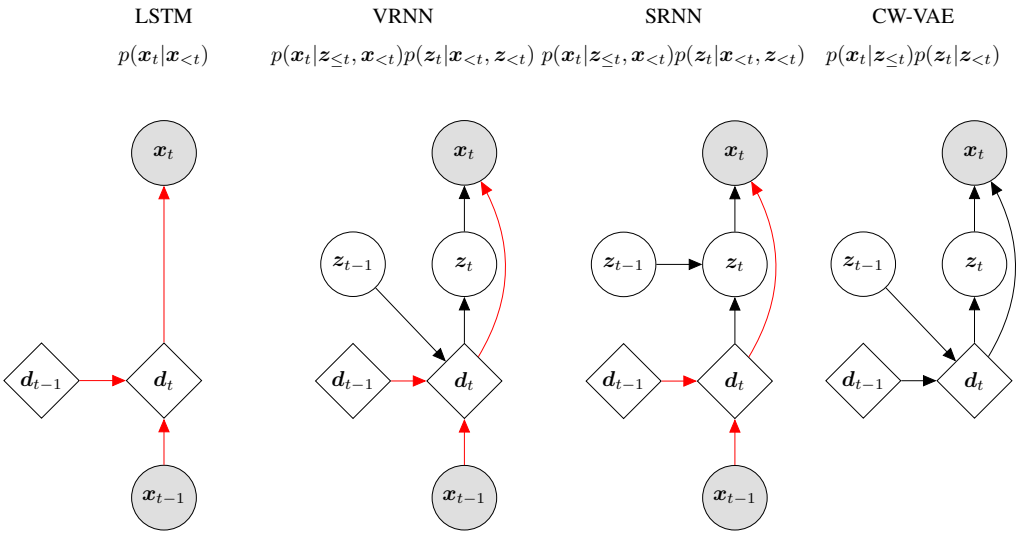



Figure 1: Generative models for a single time step of a deterministic autoregressive LSTM, the VRNN and SRNN models, and the CW-VAE with one layer of latent variables. Red arrows indicate purely deterministic paths from the output $\boldsymbol{x}_t$ to previous input $\boldsymbol{x}_{<t}$ without passing a stochastic node. Since the CW-VAE is not autoregressively dependent on the observed variable and inference injects information into only the stochastic variables, information cannot flow through such a fully deterministic path. This is possible in the VRNN and SRNN.

variables. In direct comparisons, the introduced model outperforms its deterministic counterparts. However, due to computational cost, it remains infeasible to run the model on the same setting as a state-of-the-art WaveNet model.

(III) The introduced model finds strongly clustered speech-related features in its hierarchy of latent variables building a strong case for utilizing such models for other tasks such as semi-supervised learning.

This shows that LVMs without powerful autoregressive decoders for the observed variable have potential as generative speech models when using expressive hierarchies of latent variables.

## 2 Latent Variable Models for Speech

### 2.1 Related work

LVMs formulated in context of the Variational Autoencoder (VAE) framework (Kingma & Welling, 2014; Rezende et al., 2014) continue to be of interest due to their ability to learn an approximation to the posterior distribution over the dataset. The posterior is usually on a significantly reduced dimensionality compared to the input data and lies very close to a known prior distribution. Such approximated posterior provides use cases for other tasks beyond generation such as semi-supervised learning (Kingma et al., 2014) and out-of-distribution detection (Havtorn et al., 2021). Furthermore, from image modeling research, we know that powerful LVMs can achieve state-of-the-art performance without costly autoregressive dependencies on the observed variable.

In recent years, there have been several complementary ways of improving the expressiveness of the VAE such as building more expressive priors through methods such as Normalizing Flows (Rezende & Mohamed, 2015) and building a deeper hierarchy of stochastic latent variables such as the Ladder VAE (Sønderby et al., 2016). In this research, we choose to focus on the latter due to the recent breakthroughs resulting in state-of-the-art VAEs without costly autoregressive dependencies on the observed variable (Maaløe et al., 2019; Vahdat & Kautz, 2020; Child, 2021).

To date, there two widely cited, and to our knowledge state-of-the-art, explicit likelihood generative LVMs for speech:

- The Variational Recurrent Neural Network (VRNN) (Chung et al., 2015).
- The Stochastic Recurrent Neural Network (SRNN) (Fraccaro et al., 2016).

We have indeed seen other recent LVM contributions that achieve impressive results. Amongst the most noteworthy are the FH-VAE (Hsu et al., 2017), that leverages another stochastic latent variable to capture global latent features in the speech, and the VQ-VAE (Oord et al., 2018b), that introduces a hybrid between an LVM with a quantized latent space and an autoregressive model to generate improved empirical samples. However, the FH-VAE, with its disjoint latent variables, and the VQ-VAE, with its quantized latent space autoregressive prior fitted after training the encoder/decoder, introduce significant changes to the original VAE framework to function.

Building on learnings from the LVM improvements in the image domain, we formulate a novel temporal LVM by introducing a hierarchy of stochastic latent variables by adapting a model recently introduced for video prediction:

- The Clockwork Variational Autoencoder (CW-VAE) (Saxena et al., 2021).

### 2.2 Temporal variational autoencoding

The VRNN, SRNN and CW-VAE are all autoencoders and take as input a variable-length sequence $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{T_x})$ with $\boldsymbol{x}_t \in \mathcal{X}^{D_x}$. In general, $\boldsymbol{x}$ may refer to the original observed variable or a deterministic and temporally downsampled representation of the observed variable.

First, $\boldsymbol{x}$ is encoded to a temporal stochastic latent representation $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{T_z})$ with $\boldsymbol{z}_t \in \mathcal{Z}^{D_z}$ and length $T_z \leq T_x$. This representation is then used to reconstruct the original input $\boldsymbol{x}$. The latent variable is assumed to follow some prior distribution $p(\boldsymbol{z}_t|\cdot)$. The prior distribution may depend on latent and observed variables at previous time steps, $\boldsymbol{z}_{<t}$ and $\boldsymbol{x}_{<t}$, but not $\boldsymbol{x}_t$. Here we have introduced the shorthand notation $\boldsymbol{z}_{<t} = (\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1})$

The models are trained to maximize a likelihood objective. The exact likelihood is given by

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \log \int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z}\,, \tag{1}$$

but is intractable to optimize due to the integration over the latent space. Instead, the true posterior is variationally approximated by $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ which yields the well-known evidence lower bound (ELBO) on the exact likelihood given by

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\right] = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x})\,, \tag{2}$$

with respect to $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$. We omit the $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ subscripts for the remainder of the paper. A graphical illustration of the models can be seen in figure 1.

### 2.3 Variational Recurrent Neural Network (VRNN)

The VRNN (Chung et al., 2015) is essentially a VAE per timestep $t$. Each VAE is conditioned on the hidden state of an RNN $\boldsymbol{d}_{t-1} \in \mathbb{R}^{D_d}$, with state transition $\boldsymbol{d}_t = f([\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}], \boldsymbol{d}_{t-1})$ where $[\cdot, \cdot]$ denotes concatenation. The joint distribution over observed and latent variables factorizes over time and the latent variables are autoregressive in both the observed and latent space:

$$p(\boldsymbol{x}, \boldsymbol{z}) = \prod_t p(\boldsymbol{x}_t|\boldsymbol{z}_{\leq t}, \boldsymbol{x}_{<t}) p(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t})\,. \tag{3}$$

The approximate posterior distribution similarly factorizes over time:

$$q(\boldsymbol{z}|\boldsymbol{x}) = \prod_t q(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t})\,. \tag{4}$$

The ELBO then becomes

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_t \log p(\boldsymbol{x}_t|\boldsymbol{z}_{\leq t}, \boldsymbol{x}_{<t}) - \mathrm{KL}\left(q(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}) \,||\, p(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t})\right)\right]\,. \tag{5}$$

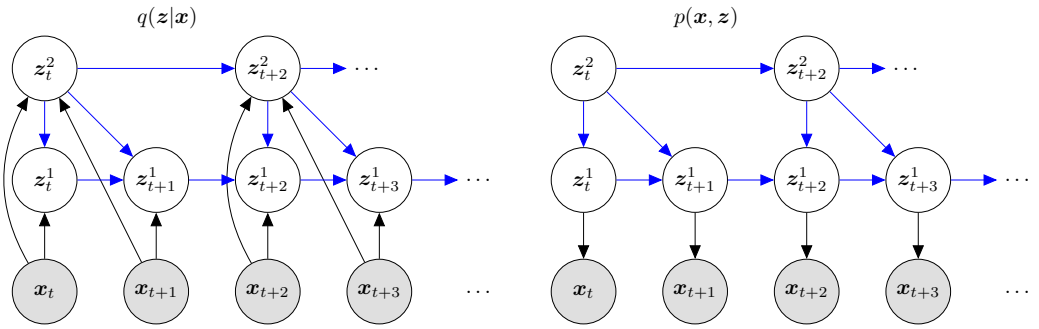



Figure 2: Inference (left) and generative (right) models for the Clockwork VAE with a hierarchy of two latent variables with $s_1 = 1$ and $s_2 = 2$. The models are unrolled over four consecutive time steps but note that the graph continues towards $t = 0$ and $t = T_x$. Blue arrows indicate parameter sharing between the inference and generative models. We omit the deterministic variable for clarity.

The VRNN uses isotropic Gaussian distributions for the prior and posterior. It uses multilayer perceptrons (MLPs), denoted by $\varphi$, to parameterize all distributions.

$$q(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}) = \mathcal{N}\left(\boldsymbol{\mu}_{z,t}, \text{diag}(\boldsymbol{\sigma}_{z,t}^2)\right) , \qquad \text{where } \left[\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}_{z,t}^2\right] = \varphi_{\text{vrnn}}^{\text{enc}}(\boldsymbol{x}_t, \boldsymbol{d}_t) , \tag{6}$$

$$p(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}) = \mathcal{N}\left(\boldsymbol{\mu}_{0,t}, \text{diag}(\boldsymbol{\sigma}_{0,t}^2)\right) , \qquad \text{where } \left[\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}^2\right] = \varphi_{\text{vrnn}}^{\text{prior}}(\boldsymbol{d}_t) , \tag{7}$$

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{\leq t}, \boldsymbol{x}_{<t}) = \mathcal{D}\left(\boldsymbol{\rho}_{x,t}\right) , \qquad \text{where } \boldsymbol{\rho}_{x,t} = \varphi_{\text{vrnn}}^{\text{dec}}(\boldsymbol{z}_t, \boldsymbol{d}_t) , \tag{8}$$

The recurrent transition function $f$ is parameterized by a Gated Recurrent Unit (GRU) (Cho et al., 2014). At timestep zero, $\boldsymbol{d}_0$ is chosen as the zero vector. $\mathcal{D}(\boldsymbol{\rho}_{x,t})$ denotes any output distribution parameterized by a set of parameters $\boldsymbol{\rho}_{x,t}$.

We note that since the decoder is dependent on $\boldsymbol{d}_t$, the transition function $f$ allows the VRNN to learn to ignore parts of or the entire latent variable and establish a purely deterministic transition from $\boldsymbol{x}_{t-1}$ to $\boldsymbol{d}_t$ similar to a regular GRU (see figure 1) which well-know weakness of VAEs with powerful decoders (Bowman et al., 2016; Sønderby et al., 2016).

### 2.4 Stochastic Recurrent Neural Network (SRNN)

The SRNN (Fraccaro et al., 2016) is similar to the VRNN but differs by separating the stochastic latent variables from the deterministic representations entirely (see figure 1). In generation, this is done by having a GRU run forwards in time over the observed variable to form a deterministic representation $\boldsymbol{d}_t$ from $\boldsymbol{x}_{<t}$. The latent variable is then sampled from the prior $p(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t})$ which is conditioned directly on the previous latent variable.

The SRNN also uses a more intricate inference network which essentially learns to solve a smoothing problem rather than a filtering problem by also running backwards in time. Specifically, in the smoothing configuration, the inference model $q(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t})$ includes an additional deterministic variable computed from $\boldsymbol{d}_t$ and $\boldsymbol{x}_t$ by a GRU running backwards in time i.e. $\boldsymbol{a}_t = f^{\text{rev}}([\boldsymbol{x}_t, \boldsymbol{d}_t], \boldsymbol{a}_{t+1})$. In the filtering configuration, this is replaced with an MLP, $\boldsymbol{a}_t = f^{\text{rev}}([\boldsymbol{x}_t, \boldsymbol{d}_t])$. The encoding distribution $q(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t})$ is then conditioned on $\boldsymbol{a}_t$, $\varphi_{\text{srnn}}^{\text{enc}}(\boldsymbol{x}_t, \boldsymbol{a}_t)$. In our experiments we run the SRNN in the smoothing configuration.

### 2.5 Clockwork Variational Autoencoder (CW-VAE)

The CW-VAE (Saxena et al., 2021) is a hierarchical latent variable model recently introduced for video generation. Contrary to the VRNN and SRNN, it is designed to make use of a hierarchy of latent variables and uses top-down inference (Sønderby et al., 2016) which enables learning a covariance between the latent variables in the hierarchy. As illustrated in figure 1, it is autoregressive in the latent space but not in the observed space. Since inference only injects information into the stochastic variables, there is no deterministic path connecting the previously observed variables to the next. This forces the information to flow through the stochastic latent variables (figure 1). Hence, the VRNN and SRNN can collapse to a deterministic model while the CW-VAE cannot.

We denote the latent variable at timestep $t$ and layer $l \in [1, L]$ by $\boldsymbol{z}_t^l$. In CW-VAE each latent layer is updated only every $s_l$ timesteps, where $s_l$ is a layer-dependent integer, or stride, defined in a way such that $s_l$ is largest for larger values of $l$. This imposes the inductive bias that latent variables exist at different temporal resolutions with $\boldsymbol{z}^l$ changing over longer time scales than $\boldsymbol{z}^{l-1}$. In speech, phonetic variation ($10 - 400\,\mathrm{ms}$), morphological and semantic features at the word level and speaker-related variation at the global scale make this a reasonable assumption.

The timesteps at which a layer updates its latent state are given by $\mathcal{T}_l \equiv \left\{ t \in [1, T] \,\middle|\, t \mod s_l = 1 \right\}$. In practice and equivalently, we represent this by having references to the unique states copied over time, $\boldsymbol{z}_t^l \equiv \boldsymbol{z}^l_{\max_\tau \{\tau \in \mathcal{T}_l | \tau \leq t\}}$.

The joint distribution factorizes over time and over the latent hierarchy.

$$p(\boldsymbol{x}, \boldsymbol{z}) = \underbrace{\left( \prod_t p\big(\boldsymbol{x}_t | \boldsymbol{z}_t^1\big) \right)}_{\text{reconstruction terms}} \underbrace{\left( \prod_{l=1}^{L} \prod_{t \in \mathcal{T}_l} p\big(\boldsymbol{z}_t^l | \boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l+1}\big) \right)}_{\text{state transitions}} . \tag{9}$$

The inference model similarly factorizes over time and over the layers of the latent hierarchy with the posterior conditioned on a span of the observed variable $\boldsymbol{x}_{t:t+s_l}$ dependent on the layer stride $s_l$.

$$q(\boldsymbol{z}|\boldsymbol{x}) = \prod_{l=1}^{L} q\big(\boldsymbol{z}^l | \boldsymbol{x}\big) = \prod_{l=1}^{L} \prod_{t \in \mathcal{T}_l} q\big(\boldsymbol{z}_t^l | \boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l+1}, \boldsymbol{x}_{t:t+s_l}\big) . \tag{10}$$

Similar to the VRNN and SRNN, an encoder, $\varphi^{\text{enc},l}_{\text{cwvae}}$, is used to parameterize the approximate posterior $q(\boldsymbol{z}|\boldsymbol{x})$ which is taken to be an isotropic Gaussian, as is the prior. This encoder may be the same for all stochastic layers $l$, be layer-specific or use a ladder-network as in the LadderVAE (Sønderby et al., 2016). A graphical representation of the inference and generative models for a two-layered CW-VAE can be seen in figure 2.

### 2.6 Modelling speech with Clockwork VAEs

In the original paper, $s_l \equiv k^{l-1}$ for some constant $k$ making it exponentially dependent on the layer index $l$ with $s_1 = 1$. This is feasible in the video domain with typical frame rates of 30 or $60\,\mathrm{Hz}$. In the speech domain however, the observed variable can have sample rates of tens of thousands of frames per second. This results in sequence lengths that are not generally feasible to model with recurrent architectures. For this reason, we chose $s_1 \gg 1$ to achieve an initial temporal downsampling and set $s_l = c^{l-1} s_1$ for $l > 1$ and some constant $c$. For instance, with $s_1 = 64$ and $c = 8$ we get $s_2 = 512$.

We design the encoder as a ladder-network similar to Sønderby et al. (2016), as this provides some benefits compared to alternatives. Specifically, a ladder-network leverages parameter sharing across the latent hierarchy and importantly processes the full observed sequence only once and shares the resulting representations for all latent variables. This yields a more computationally efficient encoder and a higher activity in latent variables towards the top of the hierarchy. Finally, we parameterize the encoder/decoder networks using 1D convolutions that operate on the raw waveform.

### 2.7 Output distribution

The choice of output distribution $\mathcal{D}(\boldsymbol{\rho}_{x,t})$ is generally data dependent and at the most fundamental level we select either continuous and discrete distributions depending on the type of $\boldsymbol{x}$, the likelihoods of which are not comparable.

In the original VRNN and SRNN, $\mathcal{D}(\boldsymbol{\rho}_{x,t})$ was taken to be an isotropic Gaussian or a mixture of isotropic Gaussians. We note that this choice generally results in an ill-posed problem when modeling audio. In this case, $\boldsymbol{x}$ denotes a waveform sampled with some bit-depth, e.g., $b = 16$ which entails that $\boldsymbol{x} \in \{0, 1, \ldots, 2^b - 1\}$ and is discrete-valued. In the two contributions, $\boldsymbol{x}$ was scaled to take values between -1 and 1, as is usual, yielding a gap between the discrete values of $\frac{1}{2^b - 1}$. This alleviates the issue since $\boldsymbol{x}$ becomes approximately continuous as $b$ becomes large and the gap goes to zero. Nonetheless, continuous distributions, and especially mixtures of continuous

distributions, are well-known to be able to yield arbitrarily high likelihoods when used to model discrete data. This choice has also been made in other papers dealing with generating speech (Hsu et al., 2017; Zhu et al., 2020).

To correctly model discrete $\boldsymbol{x}$ with a continuous distribution we must dequantize $\boldsymbol{x}$ to be continuous by e.g. adding uniform noise or using a variational approach (Ho et al., 2019). The continuous likelihood obtained via dequantization has been shown to be a lower bound on the likelihood that could have been obtained with a discrete distribution (Theis et al., 2016). Dequantization was common in the image domain (Dinh et al., 2015; Sønderby et al., 2016) until the introduction of the discretized mixture of logistics (DMoL) (Salimans et al., 2017).

The DMoL was introduced for use in autoregressive models (Salimans et al., 2017) but has become the de facto standard output distribution for generative modeling of natural images. Notably, this is also the case for latent variable models that are not autoregressive in the observed variable (Maaløe et al., 2019; Vahdat & Kautz, 2020; Child, 2021). Recently, it was applied to generative speech modelling of raw waveforms (Oord et al., 2018a). As opposed to e.g. a categorical distribution, the DMoL induces ordinality on the output space such that values that are numerically close are also considered close in the probablistic sense. This is a sensible inductive bias for images as well as audio where individual samples represent the amplitude of light and pressure.

## 3 SPEECH MODELING EXPERIMENTS

**Data** We train models on the TIMIT (Garofolo, 1993), LibriSpeech (Panayotov et al., 2015) and LibriLight datasets (Kahn et al., 2020). We provide more details on the datasets in the appendix. We represent the audio as $\mu$-law encoded PCM standardized to (discrete) values in $[-1, 1]$ with discretization gap of $\frac{1}{2^{b-1}}$. We use the original bit depth of $16\,\text{bits}$ and sample rate of $16\,000\,\text{Hz}$. We use this representation both as the input and the reconstruction target.

**Likelihood** We report likelihoods in units of bits per sample (bps) as this yields a more intuitive and interpretable version of likelihood than total likelihood in nats. It also has direct connections with information theory and compression (Shannon, 1948; Townsend et al., 2019). In units of bits per sample, lower is better. The obtained likelihoods can be seen in tables 1 and 2. We provide a formula for how to compute this in the appendix.

| $s$ | **Model** | **Configuration** | $\mathcal{L}$ **[bps]** |
|---|---|---|---|
| - | Uniform | Uninformed | 16.00 |
| - | DMoL | Optimal | 15.60 |
| - | FLAC | - | **8.582** |
| 1 | LSTM | | 11.11 |
| 1 | WaveNet | $D_C = 16$ | 11.27 |
| 1 | WaveNet | $D_C = 32$ | 11.03 |
| 1 | WaveNet | $D_C = 96$ | **10.88** |
| 1 | WaveNet | $D_C = 160$ | 10.91 |
| 64 | LSTM | | 13.34 |
| 64 | VRNN | | 12.54 |
| 64 | SRNN | | 12.42 |
| 64 | CW-VAE | $L = 1$ | 12.60 |
| 64 | CW-VAE | $L = 2$ | **12.25** |
| 256 | LSTM | | 14.20 |
| 256 | VRNN | | 13.27 |
| 256 | SRNN | | 13.14 |
| 256 | CW-VAE | $L = 1$ | 13.11 |
| 256 | CW-VAE | $L = 2$ | **12.97** |

Table 1: Model likelihoods $\mathcal{L}$ on TIMIT represented as a 16bit $\mu$-law encoded PCM for different stochastic latent variable models compared to deterministic autoregressive baselines. For the CW-VAE, $s$ refers to $s_1$ and the two-layered models have $s_2 = 8s_1$. Likelihoods are given in units of bits per sample (bps).

**Models** Architecture and training details are sketched below, while the full details are in the appendix along with additional results for some alternative model configurations. We select model configurations that can be trained on GPUs with a maximum of 12GB of RAM and train all models until convergence. We use a DMoL with 10 mixture components for the output distribution of all models and model all datasets at their full bit depth of $16\,\text{bits}$.

We supply three elementary baselines that form approximate upper and lower bounds on the likelihood. Specifically, we evaluate an uninformed discrete uniform distribution and a two-component

| $s$ | **Model** | **Configuration** | **Likelihood $\mathcal{L}$ [bps]** | | | |
|---|---|---|---|---|---|---|
| | | | dev-clean 10h/100h | dev-other 10h/100h | test-clean 10h/100h | test-other 10h/100h |
| - | Uniform | Uninformed | 16.00 | 16.00 | 16.00 | 16.00 |
| - | DMoL | Optimal | 15.66 | 15.70 | 15.62 | 15.71 |
| - | FLAC | - | **9.390** | **9.292** | **9.700** | **9.272** |
| 1 | Wavenet | $D_C = 48$ | 11.10/10.92 | 11.02/10.80 | 11.26/11.08 | 11.22/11.00 |
| 1 | Wavenet | $D_C = 96$ | **10.96/10.89** | **10.85/10.76** | **11.12/11.01** | **11.05/10.85** |
| 64 | LSTM | | 13.65/13.49 | 13.62/13.48 | 13.64/13.47 | 13.65/13.49 |
| 64 | CW-VAE | $L = 1$ | 13.15/13.14 | 13.14/13.11 | 13.18/13.15 | 13.17/13.15 |
| 64 | CW-VAE | $L = 2$ | **13.08/13.01** | **13.06/12.99** | **13.10/13.05** | **13.11/13.04** |

Table 2: Model likelihoods $\mathcal{L}$ on LibriSpeech test sets represented as $16\,\mathrm{bit}$ $\mu$-law encoded PCM. For the CW-VAE, $s$ refers to $s_1$ and the two-layered models have $s_2 = 8s_1$. Both models are trained on the $10\,\mathrm{h}$ LibriLight subset and the $100\,\mathrm{h}$ LibriSpeech train-clean-100 subset for comparison.

DMoL distribution fitted to the training set to benchmark worst case performance. We also report the compression achieved by the lossless compression algorithm, FLAC (Coalson & Castro Lopo, 2019), which constitutes an approximate lower bound for all considered models.

We configure the WaveNet baseline as in the original paper using ten layers per block and five blocks, $L = 10, B = 5$. We select the number of channels in the convolutions throughout the model, $D_c$, to allow running it on a 12GB GPU. WaveNet predicts single samples corresponding to $s = 1$.

We also provide an LSTM baseline (Hochreiter & Schmidhuber, 1997) which runs on stacked waveforms with stack size $s$. Hence, every $\boldsymbol{d}_t$ is computed from $\boldsymbol{x}_{t:t+s}$. It uses encoders and decoders similar to the VRNN and SRNN models and differs only in the design of the recurrent cell where the LSTM is fully deterministic. We report on LSTM models with hidden state size $D_d = 256$ here. Other sizes yielded as good or inferior results and can be found in the appendix.

The configuration of the VRNN and SRNN models is with waveform stacks of size $s = 64$ or $256$. Smaller stack sizes than 64 samples yield unfeasibly slow-running models since we cannot rely on fast CUDA backends for the custom recurrent cell designs. For both models we set the latent variable equal in size to the hidden units and run them with $D_z = 256$.

The CW-VAE is configured similar to the VRNN and SRNN models using $s_1 = 64$ or $256$. We set $c = 8$ for all models which yields $s_2 = 512$ and $2048$, respectively. We run the Clockwork VAE with $L = 1, 2$ and $3$ layers of latent variables. For TIMIT we use a convolutional encoder/decoder with $D_z = 96$ while for LibriSpeech we use a dense encoder/decoder with $D_z = 256$ for lower memory requirements. The size of deterministic hidden units in the model (dense or convolutional channels) are set equal to the latent variable sizes $D_z$. We also run one large two-layered CW-VAE model with convolutional encoder/decoder on TIMIT which uses $D_z = 256$.

### 3.1 Results

For temporal resolutions of $s = 1$, the deterministic autoregressive models yield the best likelihoods with WaveNet achieving $10.88\,\mathrm{bps}$ on TIMIT as seen in table 1. Somewhat surprisingly, the LSTM baseline almost matches WaveNet with a likelihood of $11.11\,\mathrm{bps}$ at $s = 1$. Unsurprisingly, due to being autoregressive also in training, the LSTM trains considerably slower than efficient dilated convolutional architecture of WaveNet.

For $s = 64$, the LSTM runs more efficiently at larger stack sizes but this comes at the cost of higher bps. Notably, the LSTM at $s = 64$ yields a considerably worse likelihood than do VRNN, SRNN and CW-VAE at the same stride being separated by close to $1\,\mathrm{bit}$. The CW-VAE outperforms both LSTM, VRNN and SRNN when configured with a hierarchy of two latent variables reaching $12.25\,\mathrm{bits}$. With a single layer of latent variables, the missing autoregression in the observed space results in a likelihood of $12.60\,\mathrm{bps}$ which is inferior to both SRNN and VRNN but notably still

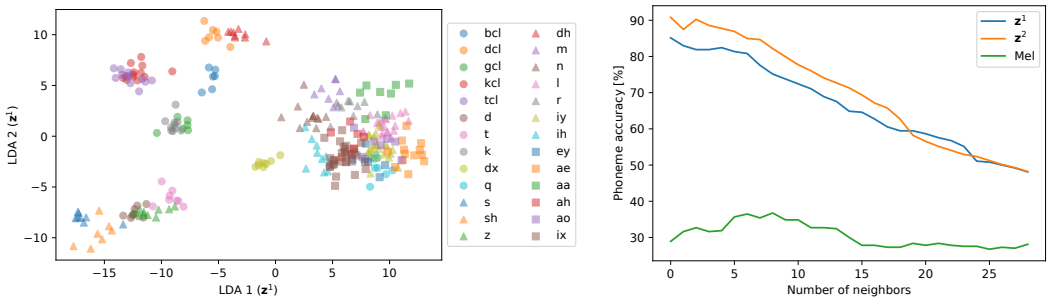



Figure 3: (left) Clustering of phonemes in a two-dimensional Linear Discriminant Analysis (LDA) subspace of a CW-VAE latent space of $\boldsymbol{z}^1$. (right) A leave-one-out classification accuracy for a $k$-nearest-neighbor classifier for different $k$ fitted to a 5D linear subspace of a CW-VAE latent space.

better than the LSTM. These observations also carry to $s = 256$, where a two-layered CW-VAE outperforms LSTM, VRNN and SRNN.

For the CW-VAE, VRNN and SRNN, decreasing the stack size $s$ improves the likelihood, as for deterministic models. This seems to indicate that LVMs may be able to outperform autoregressive models as the stack size approaches $s = 1$. However, due to computational cost of scaling up these models for $s = 1$, it was infeasible to perform this experiment.

For LibriSpeech (table 2), we see similar results with the CW-VAE improving upon the LSTM with $s = 64$ and the CW-VAE performing better with an additional layer of latent variables. WaveNet performs similar on LibriSpeech compared to TIMIT[1]. This shows that state-of-the-art likelihoods can be achieved for speech generation with autoregression purely in the latent space as long as the latent space is expressive enough. This is similar to the findings on state-of-the-art LVMs for image modeling (Maaløe et al., 2019; Vahdat & Kautz, 2020; Child, 2021).

Finally, a connection can be made between the likelihoods achieved by the considered models and the compression rates of lossless audio compression algorithms. Whereas lossy compression algorithms such as MP3 exploit the dynamic range of human hearing to achieve 70-95% reduction in bit rate (Brandenburg et al., 1998), state-of-the-art lossless compression algorithms such as FLAC achieve 50-70% (Coalson & Castro Lopo, 2019) independently of audio content, speech or otherwise. The LibriSpeech dataset achives an overall compresison to 56.2% of the original bit rate. Although both the deterministic autoregressive models and the LVMs are lossy, the objective they are trained towards minimize the amount of incurred loss which arguably makes them comparable to lossless compression algorithms. The best likelihoods achieved by the models considered above roughly correspond to a 30% reduction in bit rate which seems to indicate that there are still solid gains in likelihood to be made in speech modelling.

### 3.2 Evaluation of latent representations

A potential benefit of LVMs over purely autoregressive models is that they learn a distilled representation of the data in the form of latent variables. These should capture the high-level information in the data, and may serve as building blocks for downstream tasks. Here we evaluate the quality of this learned latent space.

**Phonemes** Phonemes are a fundamental unit of speech that relate to how certain parts of a word are pronounced. The TIMIT dataset is annotated with temporally aligned phoneme classes which allows us to analyze how they are organized in the CW-VAE latent space. Phonemes have durations in the milliseconds which for the TIMIT dataset ranges between $10$ and $400\,\mathrm{ms}$ (see figure 9 in appendix for a full overview).

We perform the evaluation with a two-layered CW-VAE with $s_1 = 64$ and $s_2 = 512$. These strides correspond to temporal resolutions of $4\,\mathrm{ms}$ and $32\,\mathrm{ms}$ both of which are on the order of phonemes.

[1]The train-clean-100 $(100\,\mathrm{h})$ and LibriLight $(10\,\mathrm{h})$ CW-VAE models are currently at epochs 70 and 400 and still converging. Final numbers will be updated for camera-ready submission.

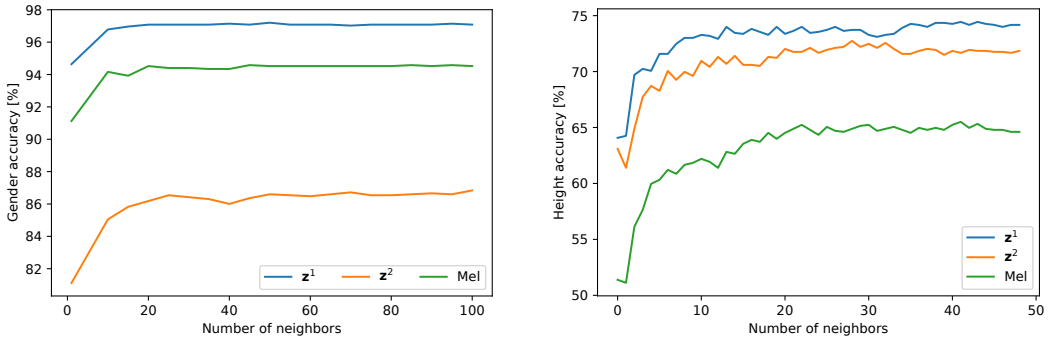



Figure 4: Leave-one-out $k$-nearest-neighbor accuracy with different $k$ for (left) the speaker's gender and (right) the height of male speakers (female speakers yield a similar result).

We therefore expect both latent variables to capture information related to phonemes. We infer the latent representations of all utterances of a single speaker from the TIMIT test set. We take $100$ Monte Carlo samples to estimate the mean representation per time step. We then compute the average latent representation over the duration of each phoneme. This approximately marginalizes out other sources of variation within the duration of the phoneme. We obtain a low-dimensional linear subspace of the latent space using a linear discriminant analysis (LDA). By considering a single speaker, we exclude inter-speaker variation in phoneme pronunciation.

In figure 3 (left) we visualize the linear subspace of $\boldsymbol{z}^1$ and the resulting phoneme clusters along with the average accuracy of a leave-one-out $k$-nearest-neighbor (KNN) classifier on the single left-out latent representation reduced with a 5-dimensional LDA. We compare to the time-averaged frequency-bin representation of a mel-spectrogram, also LDA reduced, with hop length set to 64, equal to $s_1$ for the CW-VAE, window size 256 and 80 mel bins. This is a very common preprocessing step in speech applications. We note that most phonemes are separable in the subspace and that related phonemes such as "s" and "sh" are close[2]. We show that both latent spaces yield significantly better KNN accuracies than the mel features.

**Exploration of latent variable hierarchy** In figure 4 we similarly compute a leave-one-out KNN classification on the time-averaged latent representations and mel-features for the gender and height of the speaker. We divide the height into three classes: below $175\,\mathrm{cm}$, above $185\,\mathrm{cm}$ and in-between. Compared to phonemes, these are global attributes that can affect the qualities of speech. Here we again see improved performance from using the learned latent space over mel-features. Notably, $\boldsymbol{z}^2$ is outperformed by the mel-features for gender identification which may indicate that this latent variable learns to ignore this attribute compared to $\boldsymbol{z}^1$.

## 4 Conclusion/Discussion

In this paper we have presented benchmarks for speech generation using stochastic latent variable models in comparison to deterministic autoregressive models. We have adapted the state-of-the-art video generation model, Clockwork VAE, to the speech domain, similar to how WaveNet adapted the PixelCNN. The Clockwork VAE with a hierarchy of stochastic latent variables outperformed other latent variable models that are autoregressive on both the observed and latent variables. This is in itself impressive since the Clockwork VAE is autoregressive only in the latent space. Finally, the Clockwork VAE outperformed a high-performing deterministic autoregressive model showing comparable results to the WaveNet. However, we are still to see a latent variable model outperform the WaveNet in a direct comparison. The Clockwork VAE stays too computationally expensive for this to happen. This research serves as a first step in that direction.

[2]A full description of the phonemes and phone codes used for the TIMIT dataset can be found at `https://catalog.ldc.upenn.edu/docs/LDC93S1/PHONCODE.TXT`

### REPRODUCIBILITY STATEMENT

The source code used for the work presented in this paper will be made available before the conference. This code provides all details, practical and otherwise, needed to reproduce the results in this paper including data preprocessing, model training, model likelihood and latent space evaluation. The source code also includes scripts for downloading and preparing the LibriSpeech, LibriLight and TIMIT datasets. The LibriSpeech and LibriLight datasets are open source and can be downloaded with the preparation scripts. They are also available at `https://www.openslr.org/12` and `https://github.com/facebookresearch/libri-light`, respectively. The TIMIT dataset is commercial and must be purchased and downloaded from `https://catalog.ldc.upenn.edu/LDC93S1` before running the preparation script.

The stochastic latent variable models considered in this work do not provide an exact likelihood estimate nor an exact latent space representation. For the likelihood, they provide a stochastic lower bound and some variation in the reproduced likelihoods as well as latent representations must be expected between otherwise completely identical forward passes. This variance is fairly small in practice when averaging over large datasets such as those considered in this work. We seed our experiments to reduce the randomness to a minimum, but parts of the algorithms underlying the CUDA framework are stochastic for efficiency. To retain computational feasibility, we do not run experiments with a deterministic CUDA backend.

## REFERENCES

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating Sentences from a Continuous Space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pp. 10–21, Berlin, Germany, 2016. doi: 10.18653/v1/k16-1002. URL `https://doi.org/10.18653/v1/k16-1002`.

Karlheinz Brandenburg, Ernst Eberlein, Heinz Gerhäuser, Bernhard Grill, Jürgen Herre, and Harald Popp. MPEG-2 Audio Layer III (MP3), 1998.

Rewon Child. Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021. URL `https://arxiv.org/pdf/2011.10650.pdf`.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. arXiv: 1409.1259 [cs.CL].

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A Recurrent Latent Variable Model for Sequential Data. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS)*, pp. 9, Montréal, Quebec, Canada, 2015.

Josh Coalson and Erik de Castro Lopo. Free Lossless Audio Encoding (FLAC), August 2019. URL `https://xiph.org/flac/`.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear Independent Components Estimation. In *3rd International Conference on Learning Representations Workshop*, San Diego, CA, USA, May 2015. URL `http://arxiv.org/abs/1410.8516`. arXiv: 1410.8516.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential Neural Models with Stochastic Layers. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS)*, pp. 9, Barcelona, Spain, 2016.

John S. Garofolo. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1, 1993. URL `https://catalog.ldc.upenn.edu/LDC93S1`. Web Download.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of machine learning research*, pp. 2555–2565. PMLR, June 2019. URL `http://proceedings.mlr.press/v97/hafner19a.html`.

Jakob D. Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical VAEs Know What They Don't Know. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of machine learning research*, pp. 4117–4128. PMLR, July 2021. URL https://proceedings.mlr.press/v139/havtorn21a.html.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 9, Long Beach, CA, USA, 2019. URL http://proceedings.mlr.press/v97/ho19a/ho19a.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data. In *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.

J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.E. Mazare, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, and et al. Libri-light: A benchmark for ASR with limited or no supervision. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Virtual, May 2020. doi: 10.1109/icassp40776.2020.9052942. URL http://dx.doi.org/10.1109/ICASSP40776.2020.9052942.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the the 3rd International Conference for Learning Representations (ICLR)*, San Diego, CA, USA, 2015. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, Banff, AB, Canada, 2014. URL http://arxiv.org/abs/1312.6114. arXiv: 1312.6114.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-Supervised Learning with Deep Generative Models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, Montréal, Quebec, Canada, June 2014. URL http://arxiv.org/abs/1406.5298. arXiv: 1406.5298.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6548–6558, Vancouver, Canada, February 2019. URL http://arxiv.org/abs/1902.02102.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. In *In Proceedings of the 9th ISCA Speech Synthesis Workshop*, Sunnyval, CA, USA, September 2016a. URL http://arxiv.org/abs/1609.03499.

Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with PixelCNN decoders. In *Proceedings of the 29th International Conference on Neural Information Processing Systems*, pp. 4790–4798, Barcelona, Spain, 2016b. URL https://proceedings.neurips.cc/paper/2016/hash/b1301141feffabac455e1f90a7de2054-Abstract.html.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, New York, NY, USA, August 2016c. Journal of Machine Learning. URL http://arxiv.org/abs/1601.06759.

Aäron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis.

Parallel WaveNet: Fast high-fidelity speech synthesis. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3915–3923, Stockholm, Sweden, 2018a. PMLR. URL http://proceedings.mlr.press/v80/oord18a.html. arXiv: 1711.10433.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2018b. URL http://arxiv.org/abs/1711.00937.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, Brisbane, Australia, 2015.

Adam Paszke, Gregory Chanan, Zeming Lin, Sam Gross, Edward Yang, Luca Antiga, and Zachary Devito. Automatic differentiation in PyTorch. In *In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, 2017. URL https://pytorch.org/.

Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Lille, France, 2015. URL http://arxiv.org/abs/1505.05770.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of Machine Learning Research*, volume 32, pp. 1278–1286, Beijing, China, January 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.pdf.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, April 2017. URL http://arxiv.org/abs/1701.05517.

Vaibhav Saxena, Jimmy Ba, and Danijar Hafner. Clockwork Variational Autoencoders, February 2021. URL http://arxiv.org/abs/2102.09532. arXiv: 2102.09532.

Claude E Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(July 1948):379–423, 1948. ISSN 07246811. doi: 10.1145/584091.584093. arXiv: chao-dyn/9411012 ISBN: 0252725484.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder Variational Autoencoders. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain, December 2016. URL http://arxiv.org/abs/1602.02282.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016. URL http://arxiv.org/abs/1511.01844.

James Townsend, Thomas Bird, and David Barber. Practical Lossless Compression With Latent Variables Using Bits Back Coding. In *7th International Conference on Learning Representations (ICLR)*, pp. 13, New Orleans, LA, USA, 2019.

Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. In *34th Conference on Neural Information Processing Systems (NeurIPS)*, Virtual, July 2020. URL http://arxiv.org/abs/2007.03898.

Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, seattle, WA, USA, june 13-19, 2020*, pp. 6537–6546, 2020. doi: 10.1109/CVPR42600.2020.00657. URL https://doi.org/10.1109/CVPR42600.2020.00657.

# A CONVERTING THE LIKELIHOOD TO UNITS OF BITS PER SAMPLE

Here we briefly describe how to compute a likelihood in units of bits per sample (bps). In the main text, we use $\log$ to mean $\log_e$, but here we will be explicit. In general, conversion from nats to bits (i.e., from $\log_e$ to $\log_2$) is achieved by $\log_2(x) = \log_e(x)/\log_2(e)$. Remember that $\log_2 p(\boldsymbol{x})$ factorizes as $\sum_t \log_2 p(\boldsymbol{x}_t)$. In contrast to computing bits per dimension in the image domain, it is important to remember that each example $\boldsymbol{x}^i$ must be weighted differently according the sequence length of each specific example. Thus, we compute the likelihood in bits per sample over the entire dataset as:

$$\frac{\sum_i \sum_t \log_2 p(\boldsymbol{x}_t^i)}{\sum_i T_i} \tag{11}$$

where $i$ denotes the example index, $T_i$ is the length of that example and $t$ is the time index.

# B ADDITIONAL RESULTS

We provide additional results in table 3.

# C MODEL ARCHITECTURES

**LSTM** The LSTM baseline uses an MLP encoder to embed the waveform subsegment $\boldsymbol{x}_{t:t+s-1}$ to a feature vector before feeding it to the LSTM cell. The encoder is similar to the parameterization of $\phi^{\text{enc}}_{\text{vrnn}}$ for the VRNN described above. The LSTM cell produces the hidden state $\boldsymbol{h}_t$ from $\boldsymbol{x}_{t:t+s-1}$ and passes it to a decoder. Like the encoder, the decoder is parameterized like $\phi^{\text{dec}}_{\text{vrnn}}$ of the VRNN. It outputs the waveform predictions $\boldsymbol{x}_{t+s:t+2s-1}$ from the hidden state $\boldsymbol{h}_t$.

**WaveNet** During training, WaveNet is independent from inputs outside of the receptive field. For this reason, we can train on random subsequences of training examples which reduces memory requirements. The subseqences are chosen to be of length 16000 which is larger than the receptive field.

# D DATASETS

**TIMIT** TIMIT (Garofolo, 1993) is a speech dataset which contains $16\,\text{kHz}$ recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. It amounts to 6300 total recordings splits approximately in 3.94 hours of audio for training and 1.43 hours of audio for testing. No speakers or sentences in the test set are in the training set.

TIMIT includes temporally aligned annotations of phonemes and words as well as speaker metadata such as gender, height, age, race, education level and dialect region (Garofolo, 1993).

**LibriSpeech and LibriLight** The LibriSpeech dataset (Panayotov et al., 2015) consists of readings of public domain audio books amounting to approximately $1000\,\text{h}$ of audio. The data is derived from the LibriVox project. LibriLight (Kahn et al., 2020) is a subset of LibriSpeech created as an automatic speech transcription (ASR) benchmark with limited or no supervision. We specifically train on the $100\,\text{h}$ train-clean-100 subset of LibriSpeech and the $10\,\text{h}$ subset of LibriLight. Both datasets store the audio as $16\,\text{bit}$ pulse code modulation (PCM).

# E TRAINING DETAILS

We implement all models and training scripts in PyTorch 1.9 (Paszke et al., 2017). For both datasets we use the Adam optimizer (Kingma & Ba, 2015) with default parameters as given in PyTorch. We use learning rate $3\text{e}-4$ and no learning rate schedule. We use PyTorch automatic mixed preceision (AMP) which especially for WaveNet, the LSTM baseline and the regular deterministic GRUs in

| **Model** | **Configuration** | **Likelihood $\mathcal{L}$ [bps]** |
|---|---|---|
| *Deterministic* | | |
| LSTM | $L = 1, D_h = 256, s = 256$ | 14.20 |
| LSTM | $L = 2, D_h = 256, s = 256$ | 14.17 |
| LSTM | $L = 3, D_h = 256, s = 256$ | 14.26 |
| LSTM | $L = 1, D_h = 96, s = 64$ | 13.49 |
| LSTM | $L = 2, D_h = 96, s = 64$ | 13.46 |
| LSTM | $L = 3, D_h = 96, s = 64$ | 13.40 |
| LSTM | $L = 1, D_h = 256, s = 64$ | 13.27 |
| LSTM | $L = 2, D_h = 256, s = 64$ | 13.29 |
| LSTM | $L = 3, D_h = 256, s = 64$ | 13.31 |
| LSTM | $L = 1, D_h = 512, s = 64$ | 13.37 |
| LSTM | $L = 2, D_h = 512, s = 64$ | 13.37 |
| LSTM | $L = 3, D_h = 512, s = 64$ | 13.41 |
| LSTM | $L = 1, D_h = 256, s = 16$ | 12.54 |
| LSTM | $L = 2, D_h = 256, s = 16$ | 12.54 |
| LSTM | $L = 3, D_h = 256, s = 16$ | 12.44 |
| LSTM | $L = 1, D_h = 256, s = 4$ | 11.65 |
| LSTM | $L = 1, D_h = 128, s = 1$ | 11.40 |
| LSTM | $L = 1, D_h = 256, s = 1$ | 11.11 |
| Wavenet | $D_C = 16$ | 11.27 |
| Wavenet | $D_C = 24$ | 11.14 |
| Wavenet | $D_C = 32$ | 11.03 |
| Wavenet | $D_C = 96$ | 10.88 |
| Wavenet | $D_C = 128$ | 10.98 |
| Wavenet | $D_C = 160$ | 10.91 |
| *Latent variable* | | |
| VRNN | $D_z = 96, s = 256$ | 13.51 |
| VRNN | $D_z = 256, s = 256$ | 13.27 |
| VRNN | $D_z = 96, s = 64$ | 12.93 |
| VRNN | $D_z = 256, s = 64$ | 12.54 |
| SRNN | $D_z = 96, s = 256$ | 13.28 |
| SRNN | $D_z = 256, s = 256$ | 13.14 |
| SRNN | $D_z = 96, s = 64$ | 12.87 |
| SRNN | $D_z = 256, s = 64$ | 12.42 |
| CW-VAE | $L = 1, D_z = 96, s = 256$ | 13.11 |
| CW-VAE | $L = 2, D_z = 96, s = 256$ | 12.97 |

Table 3: Model likelihoods on TIMIT represented as a 16bit $\mu$-law encoded PCM, obtained by different latent variable models and compared to autoregressive baselines. Likelihoods are given in units of bits per timestep (bpt) and obtained by normalizing the total likelihood of each sequence with the individual sequence length and then averaging over the dataset.

SRNN yield significant reductions in memory requirements. We did not observe any significant difference in final model performance compared to full ($32\,\mathrm{bit}$) precision.

We train on the 10h and 100h hour subsets of Librispeech (Panayotov et al., 2015). The 10h subset is as defined in the LibriLight dataset (Kahn et al., 2020) (a subset of Librispeech). The 100h subset corresponds to the Librispeech train-clean-100 split. In all cases we evaluate on all the test splits dev-clean, dev-other, test-clean, test-other.

Since the average sequence length in Librispeech is about 4 times longer than that of TIMIT, we do not train on the full sequences. Instead, in each epoch we randomly sample a segment of length $L_s$ frames from each training example. For WaveNet we set $L_s = 16000$ which is larger than the receptive field and train with batch size 2. For CW-VAE we set $L_s = 64000$ as the model is

autoregressive in latent space and as such can benefit from longer sequence lengths and train with batch size 16.

In testing, we evaluate on the full sequences but we need to split the test examples into segments again due to memory constraints. Hence, we do several forward passes per test example, carrying along the internal state for models that are autoregressive in training (LSTM, VRNN, SRNN, CW-VAE) while we define overlapping segments for WaveNet with the overlap given as $(\text{receptive field} - 1)$.

## F Additional discussion on the choice of output distribution

The DMoL uses a discretization of the continuous logistic distribution to define a mixture model over a discrete random variable. This allows it to parameterize multimodal distributions which can express ambiguity about the value of $\boldsymbol{x}_t$. The model can learn to maximize likelihood by assigning a bit of probability mass to multiple potential values of $\boldsymbol{x}_t$.

While this is well-suited for autoregressive modeling, for which the distribution was developed, the potential multimodality poses a challenge for non-autoregressive latent variable models which independently sample multiple neighboring observations at the output. In fact, if multiple neighboring outputs defined by the subsequence $\boldsymbol{x}_{t_1:t_2}$ have multimodal $p(\boldsymbol{x}_t|\cdot)$, we risk sampling a subsequence where each neighboring value expresses different potential realities, independently.

Interestingly, most work on latent variable models with non-autoregressive output distributions seem to ignore this fact and simply employ the mixture distribution with 10 mixture components (Maaløe et al., 2019; Vahdat & Kautz, 2020; Child, 2021). However, given the empirically good results of latent variable models for image generation, this seems to have posed only a minor problem in practice. We speculate that this is due to the high degree of similarity between neighbouring pixels in images. I.e. if the neighboring pixels are nuances of red, then, in all likelihood, so is the central pixel.

In the audio domain, however, neighbouring waveform frames can take wildly different values, especially at low sample rates. Furthermore, waveforms exhibit a natural symmetry between positive and negative amplitudes. Hence, it seems plausible that multimodality may pose a larger problem in non-autoregressive speech generation by causing locally incoherent samples than it seems to do in image modelling.

Finally, one can note that for continuous variables, the change of variables formula can in many cases convert the likelihood of a random variable following one distribution into the likelihood of another variable following a different distribution. However, the change of variables formula does not apply to conversions where one or both random variables are discrete.

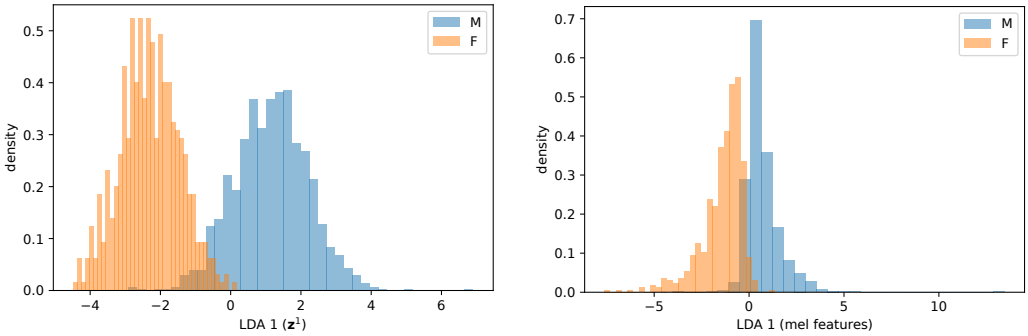
## G Graphical models for the Clockwork VAE

In figure 7 we show the unrolled graphical model of a three-layered CW-VAE with $k_1 = 1$ and $c = 2$ yielding $k_2 = 2$ and $k_3 = 4$. We show both the generative and inference models and highlight in blue the parameter sharing between the two models due to top-down inference.

In figure 8 we show the graphical model of the recurrent cell of the CW-VAE for a single time step. As noted in (Saxena et al., 2021), this cell is very similar to the one of the Recurrent State Space Model (RSSM) (Hafner et al., 2019).

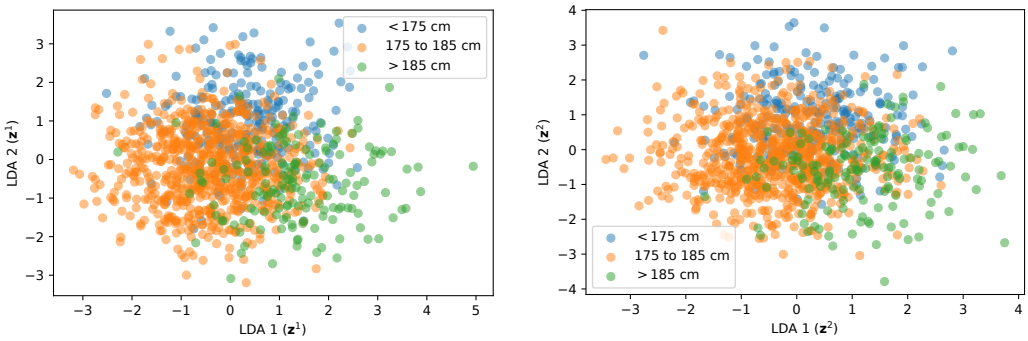
## H Distribution of phoneme duration in TIMIT

In figure 9 we plot a boxplots of the duration of each phoneme in the TIMIT dataset. We do this globally as well as for a single speaker to show that phoneme duration can vary between individual speakers.




(a) (b)

Figure 5: Clustering of speaker gender in an one-dimensional linear subspace defined by a linear discriminant analysis of the CW-VAE latent space and of a time-averaged mel spectrogram. The total overlap is slightly smaller in the subspace of the CW-VAE latent space and the separation between the distribution peaks is larger.




(a) (b)

Figure 6: Clustering of speaker height in an two-dimensional linear subspace defined by a linear discriminant analysis of the CW-VAE latent space.

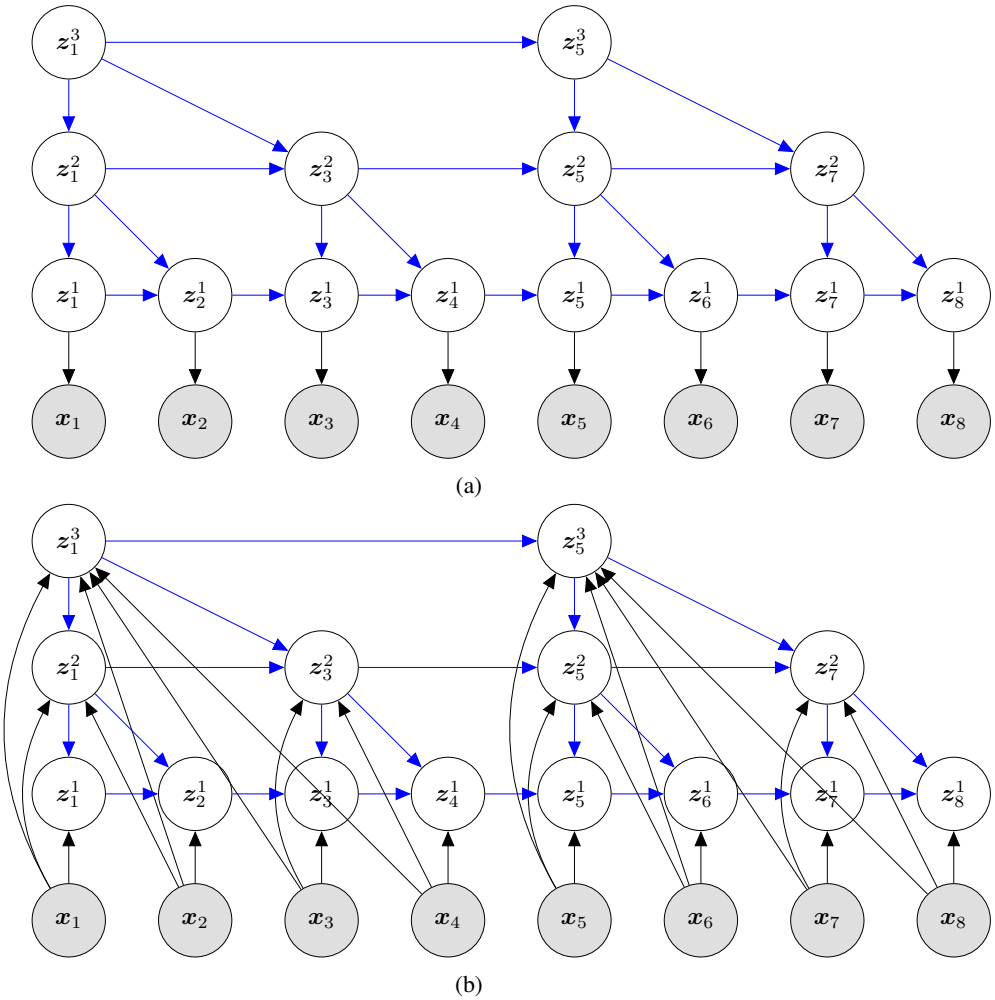



Figure 7: CW-VAE inference model $q(\boldsymbol{z}|\boldsymbol{x})$ in (a) and generative model $p(\boldsymbol{x}, \boldsymbol{z})$ in (b) for a three-layered model with $k_1 = 1$ and $c = 2$ giving $k_2 = 2$ and $k_3 = 4$. Blue arrows are (mostly) shared between the inference and generative models. See figure 8 for a detailed graphical model expanding on the latent nodes $\boldsymbol{z}_t^l$ and parameter sharing.

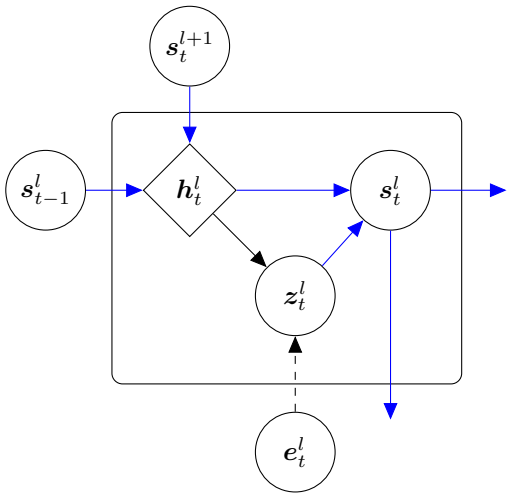
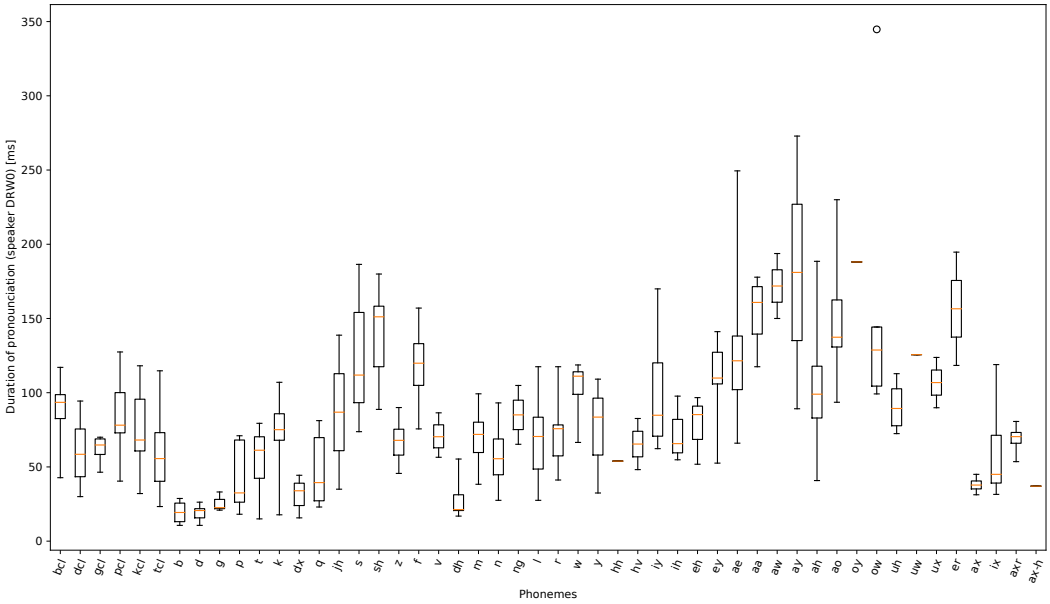
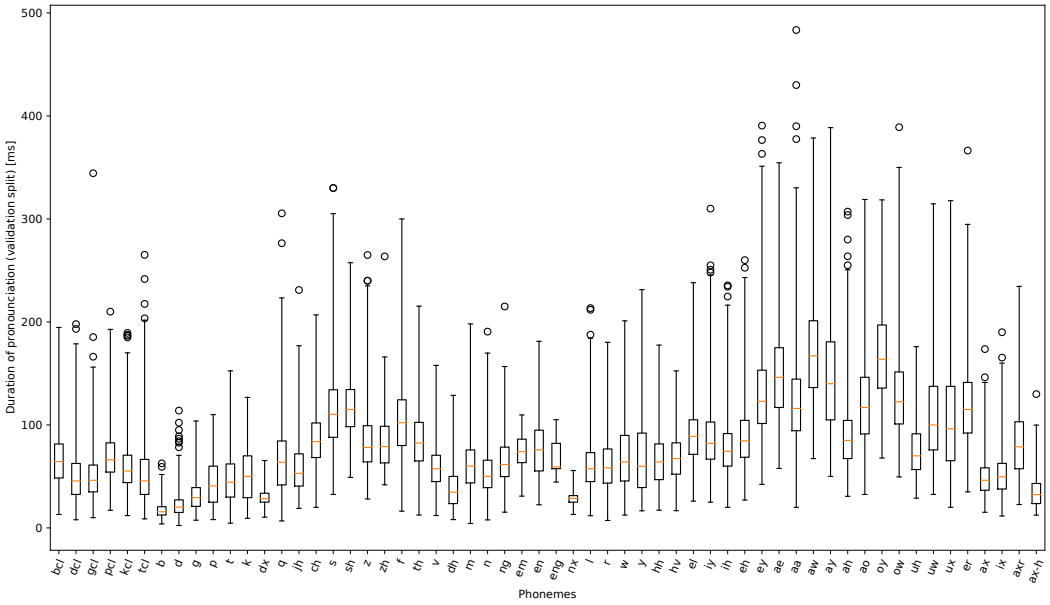



Figure 8: CW-VAE cell state $s_t^l$ update. All blue arrows are shared between generation and inference. The dashed arrow is used only during inference. The solid arrow has unique transformations during inference and generation.




(a)




(b)

Figure 9: Boxplots of the duration of the pronunciation of phonemes in TIMIT for a specific speaker DRW0 in (a) and globally in (b). Not all phonemes are pronounced by speaker DRW0 over the course of their 10 test set sentences and hence they are missing from the x-axis compared to the global durations.