A Deep Variational Convolutional Neural Network for Robust Speech Recognition in the Waveform Domain

Dino Oglic
Department of Engineering
King's College London

Zoran Cvetkovic
Department of Engineering
King's College London

Peter Sollich
Department of Mathematics
King's College London

Abstract

We investigate the potential of probabilistic neural networks for learning of robust waveform-based acoustic models. To that end, we consider a deep convolutional network that first decomposes speech into frequency sub-bands via an adaptive parametric convolutional block where filters are specified by cosine modulations of compactly supported windows. The network then employs standard non-parametric wide-pass filters, i.e., 1D convolutions, to extract the most relevant spectro-temporal patterns while gradually compressing the structured high dimensional representation generated by the parametric block. We rely on a probabilistic parametrization of the proposed architecture and learn the model using stochastic variational inference. This requires evaluation of an analytically intractable integral defining the Kullback–Leibler divergence term responsible for regularization, for which we propose an effective approximation based on the Gauss–Hermite quadrature. Our empirical results demonstrate a superior performance of the proposed approach over relevant waveform-based baselines and indicate that it could lead to robustness. Moreover, the approach outperforms a recently proposed deep convolutional network for learning of robust acoustic models with standard filterbank features.

1 Introduction

Speech recognition systems typically operate in low-dimensional feature spaces designed to implement invariances inherent to speech production and human speech recognition [38, 46]. Log Mel-filter bank values (FBANK) and their de-correlated variant known as Mel-frequency cepstral coefficients (MFCC) are two most frequently used feature extraction techniques of this kind [13, 17]. Several comparative studies of automatic and human speech recognition [5, 43, 51] suggest that the information loss inherent to such feature extraction techniques can adversely affect robustness [2, 73]. Motivated by this, we propose a principled approach for learning of robust acoustic models in the waveform domain. Robustness in automatic speech recognition is usually addressed through multi-condition training, in which the training set comprises of speech examples across the many required acoustic conditions, often constructed by mixing speech with noise at different signal-to-noise ratios (SNR). For a limited set of acoustic conditions these techniques can work well, but they are inefficient, typically requiring several thousand of hours of training data, and still the resulting models experience performance degradation in unseen environments and on different tasks. Moreover, the sheer size of the training data required for learning of robust acoustic models imposes substantial computational challenges. For instance, the requirement for more than 2 000 hours of speech in [62, 77] translates into weeks of training on a typical device with a GPU support. Our aim is to tackle these problems by incorporating relevant inductive bias into the learning process and allow for learning of robust acoustic models using moderately sized datasets. There are two main components in our approach, one dealing with the design of neural architectures and the other with inference of its parameters.

Section 2 is concerned with the design of neural architecture, which should incorporate properties known to be relevant for robustness such as invariance to local translations and stability to small diffeomorphisms that distort speech signals [40]. Moreover, the network should perform automatic feature extraction by avoiding fast compression schemes associated with information loss when operating with standard filterbank features [5, 43, 51]. To account for all of this, we design the network as a Lipschitz continuous operator mapping speech waveforms into a feature space such that small perturbations in the inputs caused by local translations and diffeomorphisms result in relatively small changes in the pre-softmax network outputs. As we operate in the waveform domain, the first layer of our convolutional network extracts information relevant for discrimination between phonetic units by decomposing a speech frame into frequency sub-bands using a set of parametric band-pass filters. The filters are defined by cosine modulations of compactly supported windows and allow for embedding of waveform signals into a structured high-dimensional space where we hypothesize that units will be easier to separate. The network then employs 1D convolutional layers with standard non-parametric wide-pass filters for extraction of relevant spectro-temporal patterns while gradually compressing the structured representation generated by the sub-band decomposition. The outputs of the last such convolutional block are passed to a multi-layer perceptron (MLP) with a softmax output.

We deal with the second component of our approach in Section 3. More specifically, we propose to learn a probabilistic parametrization of our architecture using variational inference. A typical acoustic model employs an artificial neural network with real-valued parameters. Such a *deterministic* parametrization of the network fails to capture the uncertainty of individual parameters and their importance for the learning task. Bayesian machine learning provides a principled framework for modelling uncertainty by finding plausible models that could explain the observed data [10, 24]. In particular, a (deterministic) neural network with fixed parameter values models the conditional probability of a sub-phonetic unit given a speech frame. In *probabilistic* neural networks one additionally assumes that the parameters follow some prior distribution. The latter coupled with the aforementioned likelihood gives rise to a posterior distribution of parameter values conditioned on the observed data. Such posteriors are typically defined via analytically intractable integrals that can be approximated using scalable inference techniques such as stochastic variational inference [11, 14, 28]. In particular, the main idea is to approximate intractable posteriors by optimizing over parameters of an a priori selected family of variational distributions. The optimization objective in variational inference consists of two terms: *i*) expected negative log-likelihood of the model, where the expectation is taken with respect to the variational distribution, and *ii*) Kullback–Leibler divergence that is responsible for regularization. The expectation in the first term is approximated by sampling the variational distribution which is typically given by a Gaussian mean field. In this way, variational formulation injects randomness into the forward pass that computes the loss associated with a particular mini-batch. As a result, probabilistic neural networks can capture parameter uncertainty and are less sensitive to perturbations in parameter values, as well as less susceptible to over-fitting [11, 28]. A further regularization effect, incorporated via Kullback–Leibler divergence is specified with an analytically intractable integral, for which we propose an effective approximation based on the Gauss–Hermite quadrature. This type of inference has been used previously in speech recognition (albeit in a different context) to maintain the balance between a dataset size and model complexity [67, 68]. In addition to this, a high correlation between parameter uncertainty and their importance for speech recognition has been observed in probabilistic recurrent nets [12, 28]. Previous work, however, does not operate in the waveform domain, focuses on recurrent nets, and considers variational inference separately from the properties encoded into an architecture (i.e., Lipschitz continuity of the operator).

In Section 4 we cover the relationship with prior work on speech recognition in the waveform domain. Following that, we evaluate the proposed approach empirically on two benchmark tasks for automatic speech recognition: TIMIT and AURORA4. A summary of our empirical results is provided in Section 5. Results on the first task demonstrate that our approach does not over-fit despite using a rather large network on what is considered to be a small dataset in speech recognition. The second task deals with learning in a noisy setting and our results show that the approach is capable of learning a robust model, competitive with state-of-the-art baselines for waveform-based speech recognition.

2 Parznets — Neural Architecture for Waveform-based Speech Recognition

We would like to design an architecture capable of embedding redundancies into the representation, thereby avoiding significant overlaps between positioning of different phonetic units while allowing
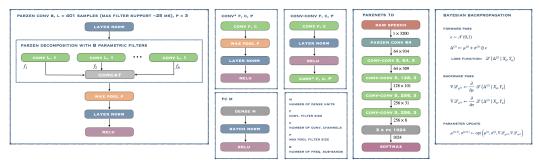
Figure 1: The figure provides a schematic for PARZNETS with 1D convolutional operators. This is supplemented with an illustration of Parzen convolutional block (the leftmost panel) that decomposes a raw speech frame into frequency sub-bands and a pseudo-code description of Bayesian backpropagation used in variational inference.

for a fair amount of additive noise and distortion at inputs. Motivated by this, we extract information relevant for discrimination between phonetic units via a parametric Parzen convolutional block (covered in Section 2.1) that decomposes a waveform frame into frequency sub-bands, thereby embedding the signal into a high-dimensional space of high-resolution spectro-temporal patterns (illustrated in Figure 1, PARZNETS 1D). To extract relevant patterns from such a representation, we rely on standard non-parametric wide-pass filters and pass the Parzen sub-bands to double convolutional blocks with 5 sample long filters (see CONV-CONV in Figure 1). The gradual compression of the spectro-temporal representation is achieved by applying the max pooling operator with size 3 (after each pair of non-parametric convolutional blocks). It has been established recently that neural networks with RELU activations realize piecewise linear functions, and, thus, we use that non-linearity throughout the network [16]. The main motivation behind this choice is to avoid further confounding effects between signal and noise by introducing additional source of non-linearity into the automatic feature extraction process. The features extracted by the last convolutional block are passed to an MLP block with three hidden layers (see FC in Figure 1), followed by a softmax output block. Section 2.2 provides a brief discussion of theoretical underpinning behind the design of our neural architecture.

2.1 Parzen Convolutional Block for Sub-band Decomposition of Speech Signals

In speech recognition, band-pass filtering of signals is traditionally performed by (weighted) averaging of power spectra [e.g., see 17, 21, or Appendix A] computed over speech frames of fixed duration. As speech signals are real-valued the moduli of their Fourier coefficients are symmetric around the origin in the frequency domain and power spectra are typically computed over non-negative frequencies only. Alternatively, the signal can be convolved by a filter directly in the time domain. To that end, we consider a family of differentiable band-pass filters based on cosine modulations of compactly supported Parzen windows [49]. In particular, we employ the squared Epanechnikov window function given by $k_\gamma(t) = \max\left\{0, 1 - \gamma t^2\right\}^2$, where γ is a parameter controlling the window width, and implicitly its frequency bandwidth. To allow for flexible placement of the centre frequency we rely on cosine modulation. Thus, Parzen filters are defined with only two differentiable parameters, η controlling the modulation frequency and γ controlling the filter bandwidth, i.e.,

$$\phi_{\eta,\gamma}(t) = \cos(2\pi\eta t) \cdot k_\gamma(t) \ . \tag{1}$$

As illustrated in Figure 1 (the leftmost panel), for each filter configuration $\{(\eta_i, \gamma_i)\}_{i=1}^B$, we use Eq. (1) to generate a one dimensional convolutional filter with maximum length given by the number of samples in 25 ms of speech; filters with shorter support are symmetrically padded with zeros. The outputs of parametric convolutions are concatenated into a high dimensional spectro-temporal decomposition of a signal and then passed to a max pooling operator, followed by layer normalization [8]. As all of the operations in this parametric block are differentiable, it is possible to construct an auto-differentiation graph that seamlessly provides gradients with respect to parameters of Parzen filters. Parzen filters are real-valued and, thus, the corresponding convolutions are simpler to implement compared to their complex-valued counterparts with exponentially modulated windows [74]. In comparison to wavelet filters [34], the Parzen convolutional block offers additional flexibility by allowing independent control over bandwidth and modulation frequency. Moreover, there is no need for the Hamming transform (e.g., employed by SINCNET filters in [57]) as an extra step to suppress the riple effects because the filters themselves are compactly supported.

3

2.2 Theoretical Background

It has been demonstrated recently that feature extraction operators (reviewed in Appendix A) that combine band-pass filtering with the modulus (square) non-linearity and (weighted) local averaging are approximately locally translation invariant and Lipschitz continuous [7]. A potential shortcoming of these operators is the fact that filter parameters are selected a priori without relying on data. As a result, the hypothesis space is selected beforehand and does not necessarily provide an ideal inductive bias for all learning tasks. Moreover, power spectra averaging characteristic to these operators is typically performed over speech segments of 25 or 32 ms [7, 40], which could be compressing the relevant information too fast into the resulting features. As a result of such compression, the feature extraction operator might be discarding the information relevant for robustness. Motivated by this, we have designed the Parzen convolutional block to address these shortcomings. In particular, the block does not rely on a priori selected filters but learns these via parametric convolutions that have a strongly encoded inductive bias. Moreover, the adaptive Parzen convolutional block along with other relevant network components performs gradual compression of the representation generated by the sub-band decomposition using the max pooling operator. The compression is done with a factor of 3 compared to sharp dimensionality reduction by a factor of over 150, characteristic to MFCC and FBANK coefficients. As previous work [26] has demonstrated that a composition of convolution with max pooling also tends to provide approximate local time-translation invariance, in our preliminary experiments, we have investigated the effectiveness of max and (weighted) ℓ_p average pooling operators, and observed that the former works the best in combination with RELU activations. Another notable difference compared to non-adaptive feature extraction operators is the use of RELU activation function instead of the modulus operator. In [41], it has been demonstrated that this change in activation function does not affect the theoretical properties of such operators. Thus, to ensure that our architecture has the properties relevant for robustness we need to establish its Lipschitz continuity. In [27], it has been demonstrated that RELU activation function is Lipschitz continuous with constant one. This activation function is also monotonic and, thus, defines a contraction. The same holds for the max operator used for signal pooling. From [27] it now follows that provided the weights of convolutional and fully connected blocks are bounded, the proposed operator is Lipschitz continuous (a composition of Lipschitz continuous operators is also Lipschitz continuous). To ensure that this happens, we propose to use a probabilistic parametrization for our network and learn the corresponding parameters using stochastic variational inference (see the next section for more details).

3 Learning Probabilistic Parznets using Stochastic Variational Inference

In deterministic neural networks, parameters/weights are real-valued and one performs inference by optimizing a loss function over them. Performing inference in probabilistic neural networks, on the other hand, requires posterior distribution over parameters given data [28]. For a fixed setting of weights, a (deterministic) neural network with the *softmax* output block models the conditional probability of a categorical label $y \in \mathcal{Y}$ given an instance $x \in \mathcal{X}$ using an exponential family model [e.g., see 33, or Appendix B]. In probabilistic networks, it is further assumed that weights have a prior distribution $p_r(\Delta \mid \eta)$, where Δ denotes all the parameters in the network and η are prior hyper-parameters. The posterior distribution of network parameters conditioned on a set of IID examples $\{(x_i, y_i)\}_{i=1}^n$ with $X_n = \{x_i\}_{i=1}^n$ and $Y_n = \{y_i\}_{i=1}^n$ is typically given by an analytically intractable integral, with parameter-specific posterior probabilities $p(\Delta \mid X_n, Y_n)$ satisfying

$$\log p(\Delta \mid X_n, Y_n) \propto \log p_r(\Delta \mid \eta) + \sum_{i=1}^n \log p(y_i \mid x_i, \Delta) .$$

Variational inference [11, 14, 28, 36] is a technique for the approximation of posterior distributions involving analytically intractable integrals. The main idea is to introduce a family of variational probability density functions $q(\Delta \mid \mu, \sigma)$, with μ and σ denoting variational parameters such that a set of these specifies a family of probability distributions. Typically, the variational family is parametrically much simpler than the posterior distribution over network parameters $p(\Delta \mid X_n, Y_n)$. The main idea is to approximate the posterior $p(\Delta \mid X_n, Y_n)$ by optimizing a lower bound on the log-marginal likelihood of the model over the parameters of the variational distribution, i.e.,

$$\min_{q \in \mathcal{Q}} \mathrm{KL}(q \mid\mid p_r) - \sum_{i=1}^n \mathbb{E}_{\Delta \sim q(\Delta \mid \mu, \sigma)} [\log p(y_i \mid x_i, \Delta)] , \tag{2}$$

where $\mathcal{Q}$ is a family of variational distributions specified by domains of parameters μ and σ. The Gaussian mean field approximation assumes that the variational distribution is the product of univariate Gaussian distributions, i.e., $q(\Delta \mid \mu, \sigma) = \prod_{i=1}^{p} \mathcal{N}(\Delta_i \mid \mu_i, \sigma_i^2)$, where p is the total number of parameters in the model, Δ_i is the i-th component of the parameter vector Δ, and $\mathcal{N}(\Delta_i \mid \mu_i, \sigma_i^2)$ denotes the fact that Δ_i follows the univariate Gaussian distribution with mean μ_i and variance σ_i^2.

The expected log-likelihood of the model $L_n(q) = \sum_{i=1}^{n} \mathbb{E}_{\Delta \sim q(\Delta \mid \mu, \sigma)}[\log p(y_i \mid x_i, \Delta)]$ is not analytically tractable and an evaluation of this expectation is required for the forward-pass when computing the loss function for a setting of variational parameters μ and σ. Stochastic variational inference approximates this term in the forward-pass by sampling the variational distribution [36]:

$$L_n(q) \approx \tilde{L}_m(q) = \frac{n}{m} \sum_{i=1}^{m} \log p(y_i \mid x_i, \Delta),$$

with $\Delta_j = \mu_j + \epsilon_j \sigma_j$ being a sample from $\mathcal{N}(\Delta_j \mid \mu_j, \sigma_j^2)$ given by $\epsilon_j \sim \mathcal{N}(\epsilon_j \mid 0, 1)$ $(1 \le j \le p)$, and where $\{(x_i, y_i)\}_{i=1}^{m}$ is a mini-batch with m random examples. As illustrated in Figure 1 (the rightmost panel), the parameters of neural network are populated with a random sample Δ drawn from the variational distribution and with that setting one computes the loss function for a particular mini-batch. The forward-pass sequence of actions is differentiable with respect to variational parameters $\upsilon = \{(\mu_i, \sigma_i)\}_{i=1}^{p}$ and unbiased. Consequently, the gradient of this estimator is also unbiased and can be computed in backward-pass by $\nabla_\upsilon L_n(q) \approx {}^n\!/_m \sum_{i=1}^{m} \nabla_\upsilon \log p(y_i \mid x_i, \Delta)$, where the network parameters Δ originate from the forward-pass components and are given by $\Delta_j = \mu_j + \epsilon_j \sigma_j$. Thus, probabilistic neural networks update the variational mean and variance parameters during gradient descent and use back-propagation for the computation of the gradients with respect to these parameters. At test time, the parameters of neural architecture are populated with variational means. In this way, a probabilistic neural network injects randomness into network parameters for each mini-batch. As a result, the inferred model can capture parameter uncertainty and is likely to be more stable to parameter perturbations than an equivalent deterministic model. A further regularization effect can be achieved via the Kullback–Leibler divergence term (Eq. 2), discussed in the next section.

3.1 Approximation of Kullback–Leibler Divergence

The Kullback–Leibler divergence term is responsible for regularization (Eq. 2) and it is defined with an analytically intractable integral that is typically approximated by Monte Carlo estimates using samples from the variational distribution [11] or prior specific second order approximations [36, 45]. We make a theoretical contribution and propose an approximation scheme based on the Gauss–Hermite quadrature, which independently of the used prior function allows for an approximation with a polynomial of arbitrarily high degree. More specifically, variational inference typically relies on Gaussian mean field approximations and this implies that the divergence term can be expressed as the sum of one dimensional integrals with respect to univariate Gaussian measures. Such integrals can be effectively approximated using the Gauss-Hermite quadrature [e.g., see 1, or Appendix C], which is a quadrature with the weighting function $\exp(-u^2)$ over the interval $u \in (-\infty, \infty)$.

The *log-scale uniform* prior was first proposed in [36], where it was argued that posteriors arising from that prior can be used to provide a theoretical justification of the dropout regularization technique [64] frequently used in the training of neural networks. The Bayesian aspect of that justification has recently been disputed in [31] but the technique can still be viewed as performing penalized log-likelihood estimation with the Kullback–Leibler divergence term responsible for regularization. The prior is given by $p_{r,\mathrm{lsu}}(\log|\Delta_i|) \propto \mathrm{const.} \Leftrightarrow p_{r,\mathrm{lsu}}(|\Delta_i|) \propto {}^1\!/_{|\Delta_i|}$, where Δ_i is some network parameter. Two different second order approximations of Kullback–Leibler divergence between Gaussian mean field posteriors and this prior distribution have been provided in [36] and [45]. We propose an alternative Gauss–Hermite approximation, formalized in the following proposition (a proof is given in Appendix D). Just as in [64] and [36], we employ a parametrization of variational Gaussian mean field known as the *dropout posterior*, with mean parameter μ_j and variance $\sigma_j^2 = \alpha_j \mu_j^2$ specified via an additional scaling parameter $\alpha_j > 0$ (for all $1 \le j \le p$).

Proposition 1. KL *divergence between a Gaussian distribution with the dropout parametrization of variance and a log-scale uniform prior can be approximated by*

$$\mathrm{KL}(q \mid\mid p_{r,\mathrm{lsu}}) \approx -{}^1\!/_2 \log \alpha + {}^1\!/_{\sqrt{\pi}} \sum_{i=1}^{s} w_i \log|v_i| + \mathrm{const.} \quad \mathit{with} \quad v_i = \sqrt{2\alpha}\, u_i + 1$$

and where $\{u_i\}_{i=1}^s$ are the roots of the Hermite polynomial with quadrature weights $\{w_i\}_{i=1}^s$.

The scale-mixture is another prior distribution frequently used in variational inference, first proposed in [11]. It resembles the so called spike and slab prior [15, 23, 44] and is given by

$$p_{r,\mathrm{sm}}\left(\Delta_i \mid \xi, \eta_1, \eta_2, \lambda\right) = \lambda \cdot \mathcal{N}\left(\Delta_i \mid \xi, \eta_1^2\right) + (1 - \lambda) \cdot \mathcal{N}\left(\Delta_i \mid \xi, \eta_2^2\right) \ ,$$

where Δ_i is a parameter of the model (see Eq. 2), η_1^2 and η_2^2 are prior (variance) hyper-parameters with $\eta_1 \ll \eta_2$, ξ is the prior mean, and $0 \leq \lambda \leq 1$ is the mixture scale. The hyper-parameters of the prior distributions (i.e., η_1, η_2, λ, and ξ) are kept fixed during optimization and can be chosen via cross-validation. The first mixture component is chosen such that $\eta_1 \ll 1$, which forces many of the variational parameters to concentrate tightly around the prior mean ξ (e.g., around zero for $\xi = 0$). The second mixture component has higher variance and heavier tails allowing parameters to move further away from the mean. The prior variance hyper-parameters are shared between all the network parameters and this is an important difference compared to approaches based on the spike and slab prior [44, 23, 15], where each model parameter has a different prior variance. The following proposition provides a mean to approximate the divergence term between Gaussian mean field variational distribution and this prior function (a proof is given in Appendix D).

Proposition 2. KL *divergence between a Gaussian distribution with the dropout parametrization of variance and a scale-mixture prior can be approximated by*

$$\mathrm{KL}\left(q \mid\mid p_{r,\mathrm{sm}}\right) \approx -\log\sqrt{2\pi\alpha\mu^2} - 1/\sqrt{\pi}\sum_{i=1}^s w_i \log p_{r,\mathrm{sm}}\left(v_i\right) - 1/2 \ \ with \ \ v_i = \left(\sqrt{2\alpha}u_i + 1\right)\mu$$

where $\{u_i\}_{i=1}^s$ are the roots of the Hermite polynomial with corresponding quadrature weights $\{w_i\}_{i=1}^s$, α and μ are variational parameters, and $p_{r,\mathrm{sm}}$ is some scale-mixture prior distribution.

4   Related Work

Whilst speech production embeds redundancies relevant for robustness, there are several challenges when dealing with these highly correlated raw speech inputs. In particular, the high dimensionality of waveform signals typically requires a larger number of parameters compared to standard features and a prolonged training time. Another difficulty is the fact that raw speech is known to be characterized by large number of variations such as temporal distortions and speaker variability [25, 62]. Acoustic models based on neural networks operating directly in the waveform domain are, thus, likely to over-fit on small and moderately sized datasets without appropriate inductive bias. In this sense, our approach that combines variational inference with Lipschitz continuity of the operator mapping provides a theoretical underpinning for the design and learning of effective waveform-based acoustic models. Previous work has also resorted to similar techniques for maintaining the balance between dataset size and model complexity. Watanabe et al. [68, 69] have used variational inference for clustering of states in triphone hidden Markov models (HMM) and learning the appropriate number of components in Gaussian mixture models (GMM). In contrast to this, we use variational inference to learn a probabilistic convolutional network that models the conditional probability of a triphone state-id given an input waveform frame. Graves [28] and Braun and Liu [12] have used variational inference to learn a recurrent neural network as part of an end-to-end acoustic model. In both of these works it has been observed that parameter uncertainty is correlated with their importance for considered speech recognition tasks. The main difference to this line of work is that neither of those models operate in the waveform domain, but rely on low-dimensional feature spaces generated by FBANK or MFCC features. This allows for scalable inference of recurrent models which is known to be computationally expensive for high dimensional inputs such as waveform signals. Moreover, prior work in speech recognition (to the best of our knowledge) considers variational inference independently of principles incorporated into the architecture such as Lipschitz continuity encoded into our operator mapping. Recently, an approach for modulation filter-learning based on encoder-decoder architecture and variational inference has been considered in [3] and [4]. The encoder takes as input a Mel-spectrogram constructed using speech segments of fixed length and learns its latent representation. The optimization of encoder-decoder parameters is performed using variational inference and the learned filters are then used to generate features that are used as input to an MLP. In contrast to this, we use variational inference to learn filters jointly with other network parameters.

A common characteristic of previous approaches for waveform-based speech recognition is the use of relatively large datasets [62, 77]. In such a regime, waveform-based acoustic models are competitive

with architectures relying on standard features (i.e., MFCC, FBANK, and FMLLR). Another difference compared to our approach is that previous architectures typically employ a convolutional layer with weighted ℓ_1 or ℓ_2 pooling (25 ms long frames) to emulate filterbank features and reduce the dimension of the representation quickly [30, 47]. In contrast to this, we perform gradual compression of the waveform sub-band decomposition via max pooling and thus overcome the information loss inherent to standard features. Moreover, we use the RELU non-linearity throughout the network and do not apply the LOG operator to the outputs of the initial block. Sainath et al. [62] propose an architecture which takes raw speech inputs and applies time-domain followed by frequency-domain one dimensional convolutions, designed to extract band-pass features from the waveform. The architecture itself is a recurrent net that requires more than 2 000 hours of training data to match the performance of models with standard features. Similarly, Zhu et al. [77] combine two convolutional layers with recurrent blocks in end-to-end training, requiring more than 2 400 hours of training data for state-of-the-art results. Ghahremani et al. [25] proposed a feedforward architecture based on convolutional feature extraction layer, with the outputs of that block passed to a TDNN. The empirical results indicates that the approach is competitive with MFCC-based architectures on large datasets. It has not been evaluated on noisy speech and it is unclear how well it generalizes on small datasets.

Our architecture performs parametric sub-band decomposition of speech waveforms and it is most closely related to SINCNET [57] that employs three 1D convolutional layers on top of the parametric block. SINCNET is considered to be the state-of-the art model for waveform-based speech recognition. A related architecture is SINC^{2}NET that links a parametric convolution block to an MLP [39]. Recently, complex-valued parametric filters have been used to initialize a complex non-parametric convolution block in a deep network for end-to-end speech recognition [74–76]. In comparison to [74], we show that our approach generalizes better on the small TIMIT dataset. In our experiments, we use the SINCNET architecture (with code available online) as a representative baselines from this class.

5   Experiments

We evaluate the proposed approach on two different datasets: TIMIT [20] and AURORA4 [48]. The first task demonstrates that the proposed approach does not over-fit on what is considered to be a small dataset in speech recognition. Moreover, the results also indicate that a combination of variational inference and Lipschitz continuous architectures for waveform-based speech recognition such as PARZNETS does not require large training datasets to outperform models based on standard filterbank features. The second task deals with noisy speech and shows that the proposed approach can learn an effective noise robust representation of waveform signals. In all the experiments, we train a context dependent hybrid HMM model based on frame labels (i.e., HMM state ids) generated using a triphone model from Kaldi [53] with 25 ms frames and 10 ms stride between the successive frames. The data splits (train/validation/test) originate from the Kaldi framework. In the pre-processing step, we assign the Kaldi frame label to the 200 ms long segment of raw speech centered at an original Kaldi frame (keeping 10 ms stride between the successive frames of raw speech). An extensive analysis containing the results of our experiments with different approximation schemes for the Kullback–Leibler divergence term defined by log-scale uniform and scale mixture priors, along with an assessment of the improvement in effectiveness of acoustic models due to adaptive modulation filters can be found in Appendix E. In the remainder of the section, we provide a brief overview of the results with log-scale uniform prior and the proposed Gauss–Hermite approximation scheme.

METHOD	AVG	MIN
A. RAW SPEECH BASELINES (OPTIMIZED FILTERS)		
VARIATIONAL PARZNETS	**16.5**	**16.2**
DETERMINISTIC PARZNETS	17.7	17.5
SINCNET [57, 60]	17.5	17.2
SINC^{2}NET [39]	–	16.9
END-TO-END CNN [74]	–	18.0
RAW SPEECH CNN [57]	18.3	18.1
B. STANDARD FEATURES (NON-ADAPTIVE FILTERS)		
FMLLR + MLP	16.9	16.7
MFCC + MLP [58]	18.1	17.8
MULTI-RES DSS + CNN & MLP [50]	-	17.4

Table 1: The table compares the error rates obtained in our experiments on TIMIT to the ones reported for relevant feedforward nets.

Table 1 summarizes our empirical results in comparison to state-of-the-art feedforward architectures on TIMIT. In addition to the lowest obtained error rate (denoted with MIN), we also report the average result over 5 simulations. A comparison to previously reported results for waveform-based speech recognition indicates that our approach performs the best on average on this task. Moreover, this is the first such approach that outperforms all the feedforward architectures built on top of standard statically extracted features. The results also show that variational inference contributes to 7.4% relative improvement on this dataset over a relevant deterministic network (see Appendix E) regularized with standard dropout technique [29]. We note

here that recent work has reported lower error rates on TIMIT using recurrent nets and statically extracted features. In particular, [59] reports the following error rates for gated recurrent units (GRU): LI-GRU 15.8% and LI-GRU FMLLR 14.8%. More recently, Baevski et al. [9] have used 960 hours of data from LIBRISPEECH and unsupervised pre-training to learn an effective model for TIMIT with error rate 11.4% (VQ-WAV2VEC + BERT). Our future work will explore recurrent architectures in the waveform-domain, combined with regularization mechanisms provided by variational inference.

AURORA4 is a medium vocabulary task based on clean speech from the Wall Street Journal (WSJ0) corpus [22]. The clean speech was corrupted by six different noise types at different SNRs. The tests sets consist of noise corrupted utterances recorded by a primary and a secondary microphone. In Table 2 we provide a summary of our results on this dataset relative to state-of-the-art feedforward architectures (see also Appendix E). The first experiment compares our approach (8 X CNN1D) to the state-of-the-art archi-

METHOD	A	B	C	D	AVG
A. RAW SPEECH & VARIATIONAL BASELINES (OPTIMIZED FILTERS)					
VAR. PARZNETS (10 X CNN1D)	2.78	5.06	5.27	15.18	**9.25**
VAR. PARZNETS (8 X CNN1D)	2.88	5.05	5.59	15.53	9.42
SINCNET [60]	3.42	6.33	6.13	16.99	10.68
CVAE FEATS + MLP [3, 4]	3.50	7.40	6.90	17.10	11.20
B. STANDARD FEATURES (NON-ADAPTIVE FILTERS)					
FBANK + VD10 X CNN2D [55]	4.13	6.62	5.92	14.53	9.78
FBANK + VD8 X CNN2D [55]	3.72	6.57	5.83	14.79	9.84
FMLLR + MLP	3.34	6.27	5.74	16.04	10.21
MFCC + MLP	4.28	7.44	8.73	18.71	12.14

Table 2: The error rates obtained on different test samples from AURORA4 using context frames of 200 ms (A: clean speech with same microphone, B: average error rate for noisy speech with same microphone, C: clean speech with different microphones, D: average error rate for noisy speech with different microphones).

tecture for waveform-based speech recognition [57, SINCNET] and shows a statistically significant [18, 71, Wilcoxon test, 95% confidence] improvement over that baseline. We also compare to a recent approach for modulation filter-learning using encoder-decoder architecture and variational inference [3, 4]. The results again show (with 95% confidence) that the proposed approach is statistically significantly better than the baseline from [3, 4]. Following this, we compare our results to the error rates reported in [55] for 8 and 10-layer deep 2D convolutional networks (VDCNN2D) based on statically extracted features using 200 ms long raw-speech segments (i.e., 17 FBANK frames). This might be an unfair comparison to our approach, because we use the less expressive 1D convolutions in our architecture. Still, the results indicate that variational PARZNETS architecture with 8 convolutional layers outperforms the network with 10 CNN2D layers from [55]. Furthermore, we extend our architecture (Figure 1) to 10 convolutional layers by employing time-padding in 1D convolutions to allow for another double convolutional block. The results indicate a further improvement in accuracy as a result of this modification. Another particularly interesting observation is that the gains of our approach over noisy samples do not come as a result of performance degradation over clean speech. We note here that [55] reports a slightly better error rate with 2D convolutions and FBANK features when the context size is increased beyond 200 ms (i.e., 21 frames), in combination with time and frequency padding (WER 8.81%). Moreover, a recent approach based on multi-octave convolutions and 15 such convolutional layers has achieved the error rate of 8.31% on this dataset [61]. Our future work will explore variational inference in combination with deep 2D convolutional networks, which are challenging to fit into our GPU devices due to a gradual compression of waveform representation. Finally, we have also evaluated our approach relative to MLPs based on frequently used feature extraction techniques: MFCC and FMLLR; again observing a significant improvement in accuracy.

Conclusion

We have outlined a principled framework for learning of robust waveform-based acoustic models. The framework combines stochastic variational inference with a Lipschitz continuous architecture/operator that learns to gradually extract features relevant for robustness. The approach operates directly in the waveform domain to avoid potential information loss inherent to standard feature extraction techniques such as MFCC and FBANK coefficients. In our experiments, the approach outperforms recently proposed architectures for waveform-based speech recognition (e.g., SINCNET) as well as a relevant deep convolutional network for learning of robust acoustic models using FBANK features. Moreover, our empirical results show that the proposed approach allows for learning of effective acoustic models using relatively small datasets. Our future work will explore the potential of probabilistic recurrent architectures operating in the waveform domain as well as different priors that could further improve the inductive bias via the regularization mechanism provided by the Kullback–Leibler divergence term. To the best of our knowledge, this is the first time that a variational approach has achieved results competitive with state-of-the-art on continuous speech recognition.

Acknowledgements

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1972.

[2] M. Ager, Z. Cvetkovic, and P. Sollich. Combined waveform-cepstral representation for robust speech recognition. *IEEE ISIT*, 2011.

[3] P. Agrawal and S. Ganapathy. Deep variational filter learning models for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019a.

[4] P. Agrawal and S. Ganapathy. Modulation filter learning using deep variational networks for robust speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 2019b.

[5] L. Alsteris and K. Paliwal. Further intelligibility results from human listening tests using the short-time phase spectrum. *Speech Communication*, 48, 2006.

[6] Y. Altun, A. J. Smola, and T. Hofmann. Exponential families for conditional random fields. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 2–9. AUAI Press, 2004.

[7] J. Andén and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 2014.

[8] L. J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

[9] A. Baevski, S. Schneider, and M. Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *International Conference on Learning Representations*, 2020.

[10] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[11] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[12] S. Braun and S. Liu. Parameter uncertainty for end-to-end speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[13] J. S. Bridle and M. Brown. An experimental automatic word-recognition system. Technical Report 1003, JSRU, Ruislip, UK, 1974.

[14] W. L. Buntine and A. S. Weigend. Bayesian back-propagation. *Complex Systems*, 1991.

[15] H. Chipman. Bayesian variable selection with related predictors. *Canadian Journal of Statistics*, 1996.

[16] F. Croce and M. Hein. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. In *International Conference on Learning Representations*, 2020.

[17] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1980.

[18] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 2006.

[19] T. Dozat. Incorporating Nesterov momentum into Adam. 2015.

[20] W. Fisher, G. Doddington, and K. Goudie-Marshall. The DARPA speech recognition research database: specifications and status. In *Proc. of DARPA Workshop on Speech Recognition*, 1986.

[21] M. Gales and S. Young. The application of hidden Markov models in speech recognition. *Foundations and Trends in Signal Processing*, 2007.

[22] J. Garofolo, D. Graff, D. Paul, and D. Pallett. CSR-I (WSJ0) Complete LDC93S6A. *Linguistic Data Consortium*, 1993.

[23] E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[24] Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015. On Probabilistic models.

[25] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur. Acoustic modelling from the signal domain using CNNs. In *INTERSPEECH*, 2016.

[26] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[27] H. Gouk, E. Frank, B. Pfahringer, and M. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *arXiv:1804.04368*, 2018.

[28] A. Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc., 2011.

[29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

[30] Y. Hoshen, R. Weiss, and K. W. Wilson. Speech acoustic modeling from raw multichannel waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[31] J. Hron, A. Matthews, and Z. Ghahramani. Variational Bayesian dropout: pitfalls and fixes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2019–2028. PMLR, 10–15 Jul 2018.

[32] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

[33] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[34] H. Khan and B. Yener. Learning filter widths of spectral decompositions with wavelets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[35] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

[36] D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*. 2015.

[37] S. Kullback. *Information Theory and Statistics*. Wiley, 1959.

[38] F. Li, A. Trevino, A. Menon, and J. Allen. A psychoacoustic method for studying the necessary and sufficient perceptual cues of american english fricative consonants in noise. *The Journal of the Acoustical Society of America*, 132:2663–75, 2012.

[39] E. Loweimi, P. Bell, and S. Renals. On learning interpretable cnns with parametric modulated kernel-based filters. In *INTERSPEECH*, 2019.

[40] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 2012.

[41] S. Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2016.

[42] A. May, A. B. Garakani, Z. Lu, D. Guo, K. Liu, A. Bellet, L. Fan, M. Collins, D. Hsu, B. Kingsbury, M. Picheny, and F. Sha. Kernel approximation methods for speech recognition. *Journal of Machine Learning Research*, 2019.

[43] B. Meyer, M. Wächter, T. Brand, and B. Kollmeier. Phoneme confusions in human and automatic speech recognition. *INTERSPEECH*, 2007.

[44] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 1988.

[45] D. Molchanov, A. Ashukha, and D. Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[46] R. C. Moore, T. Lee, and F. E. Theunissen. Noise-invariant neurons in the avian auditory cortex: Hearing the song in noise. *PLOS Computational Biology*, 9(3):1–14, 2013.

[47] D. Palaz, R. Collobert, and M. Magimai-Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. In *INTERSPEECH*, 2013.

[48] N. Parihar and J. Picone. Aurora working group: DSR front end LVCSR evaluation AU/384/02, 2002.

[49] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 1962.

[50] V. Peddinti, T. Sainath, S. Maymon, B. Ramabhadran, D. Nahamoo, and V. Goel. Deep scattering spectrum with deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

[51] S. Peters, P. Stubley, and J.-M. Valin. On the limits of speech recognition in noise. 1999.

[52] M. Plancherel. Contribution á l'étude de la représentation d'une fonction arbitraire par les intégrales définies. In *Rendiconti del Circolo Matematico di Palermos*, 1910.

[53] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The Kaldi speech recognition toolkit. In *IEEE ASRU*, 2011.

[54] P. Prandoni and M. Vetterli. *Signal Processing for Communications*. New York: Taylor & Francis Group, 1st edition, 2008.

[55] Y. Qian, M. Bi, T. Tan, and K. Yu. Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2016.

[56] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2005.

[57] M. Ravanelli and Y. Bengio. Speech and speaker recognition from raw waveform with SincNet. *arXiv:1812.05920*, 2018.

[58] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Batch-normalized joint training for dnn-based distant speech recognition. *IEEE Spoken Language Technology Workshop*, 2016.

[59] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computing*, 2, 2018.

[60] M. Ravanelli, T. Parcollet, and Y. Bengio. The PyTorch-Kaldi speech recognition toolkit. *arXiv:1811.07453*, 2018.

[61] J. Rownicka, P. Bell, and S. Renals. Multi-scale octave convolutions for robust speech recognition. In *IEEE ASRU*, 2019.

[62] T. Sainath, R. J. Weiss, K. Wilson, A. W. Senior, and O. Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *INTERSPEECH*, 2015.

[63] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, 2016.

[64] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

[65] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, 2002.

[66] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[67] S. Watanabe and A. Nakamura. Bayesian approaches to acoustic modeling: a review. *APSIPA Transactions on Signal and Information Processing*, 1, 2012.

[68] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Bayesian acoustic modeling for spontaneous speech recognition. 2003.

[69] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Variational bayesian estimation and clustering for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 12(4):365–381, 2004.

[70] B. L. Welch. The generalization of student's problem when several different population variances are involved. *Biometrika*, 1947.

[71] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1945.

[72] S. Xiang and F. Bornemann. On the convergence rates of Gauss and Clenshaw–Curtis quadrature for functions of limited regularity. *arXiv:1203.2445v3*, 2012.

[73] J. Yousafzai, P. Sollich, Z. Cvetkovic, and B. Yu. Combined features and kernel design for noise robust phoneme classification using support vector machines. *IEEE Transactions on Audio, Speech and Language Processing*, 19:1396–1407, 2011.

[74] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schatz, G. Synnaeve, and E. Dupoux. Learning filterbanks from raw speech for phone recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

[75] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux. End-to-end speech recognition from the raw waveform. In *INTERSPEECH*, pages 781–785, 2018.

[76] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert. Fully convolutional speech recognition. *arXiv:1812.06864*, 2018.

[77] Z. Zhu, J. Engel, and A. Hannun. Learning multiscale features directly from waveforms. In *INTERSPEECH*, 2016.

A Scattering Operators

A.1 First Order Scattering: FBANK and MFCC

We start with a brief review of Mel-frequency cepstral coefficients [17, 21], which is a frequently used feature extraction operator based on band-pass filtering of speech signals. The main idea behind the approach is to perform averaging of power spectra with band-pass filters and in this way obtain the (approximate) Lipschitz continuity of the operator mapping. More formally, Mel-frequency spectrogram of a signal is given by

$$(\mathcal{M}f)(t \mid \eta, \alpha, \beta) = \frac{1}{2\pi} \int \left| \hat{f}(t, \omega) \right|^2 \hat{\psi}(\omega \mid \eta, \alpha, \beta)^2 \, \mathrm{d}\omega \,,$$

where $\left| \hat{f}(t, \omega) \right|^2 = \left| \int f(u) \zeta_t(u) \exp(-i\omega u) \, \mathrm{d}u \right|^2$ is the power spectra of a signal f (i.e., modulus squared of the Fourier transform coefficients), $|\cdot|$ is the modulus of a complex number, ζ_t is a window of duration T centered at some time-index t with $\int \zeta_t(u) \, \mathrm{d}u = 1$, $\hat{\psi}(\omega \mid \eta, \alpha, \beta)$ is the square root of a triangular probability distribution with mode η and support on the interval $[\alpha, \beta]$, and ω is a frequency component. Mel-frequency spectrograms are typically defined with a family of triangular distributions (e.g., 40 band-pass filters). The modes of these distributions are selected so that they are equidistant in the log-space of the spectrum (the Mel-scale characteristic to this family of filters corresponds to the natural logarithm) and the support of each distribution is defined by the modes of the neighboring filters. As a result of this, Mel-frequency spectrograms perform power spectra averaging over high frequency components with larger bandwidths compared to low frequency components and are typically stable to actions of a small diffeomorphism. Moreover, in [7] it is argued that Mel-spectrograms typically define a translation invariant Lipschitz continuous operator. The features obtained by passing Mel-spectrograms through log-activation function are known as FBANK features in speech recognition. The Mel-frequency cepstral coefficients are obtained by applying the cosine transform to FBANK features and taking only the resulting cepstral coefficients of lower orders (this is also a way to perform de-correlation of FBANK coefficients).

An alternative band-pass filtering approach for spectral decomposition of signals has been outlined in [40]. The approach is motivated by the fact that as a result of power spectra averaging with respect to 25 long ms windows, Mel-cepstral coefficients can contribute to information loss, which can have a negative impact on the performance of a supervised learning algorithm. The main idea is to re-arrange the terms in the integral defining the Mel-spectrogram of a signal and in this way introduce an operator capable of performing the filtering of the whole signal instead of filtering only windows of fixed length determined by ζ_t. More specifically, Mel-frequency spectrogram can be written as

$$(\mathcal{M}f)(t \mid \theta) = \frac{1}{2\pi} \int \left| \hat{f}(t, w) \right|^2 \left| \hat{\psi}(w \mid \theta) \right|^2 \mathrm{d}w = \int \left| \int f(u) \zeta_t(u) \psi(v - u \mid \theta) \, \mathrm{d}u \right|^2 \mathrm{d}v \,,$$

where $\hat{\psi}(\cdot \mid \theta)$ is the Fourier transform of some filter $\psi(\cdot \mid \theta)$ (abbreviated ψ_θ) defined with a hyperparameter vector θ. The second equality is a consequence of *Plancherel's theorem* (and convolution theorem), which states that *the integral of the square of the Fourier transform of a function is equal to the integral of the square of the function itself* [e.g., see 52, 54]. Now, the main idea in [40] is to re-arrange the terms appearing in the integral defining the Mel-spectrogram and filter the whole signal instead of filtering it by parts determined by a window of fixed length T. The resulting operator is called the (squared) first order scattering operator and it is defined by [7, 40]

$$\mathcal{S}_1^2 f(t \mid \theta) = \int \left| \int f(u) \psi(v - u \mid \theta) \, \mathrm{d}u \right|^2 |\zeta(t - v)|^2 \, \mathrm{d}v = \left(|f * \psi_\theta|^2 * |\zeta|^2 \right)(t) \,,$$

where $*$ denotes the convolution of one dimensional signals. As the modulus squared operator can amplify large coefficients, Andén and Mallat [7] and Mallat [40] have proposed to replace it with the modulus operator and, thus, define a more stable signal representation $\mathcal{S}_1 f(t \mid \theta) = (|f * \psi_\theta| * \zeta)(t)$. In the latter operator, the windowing function ζ acts as a low-pass filter and performs weighted l_1-average pooling of the previously filtered signal. The scattering operator can be extended to a higher order signal decomposition by applying the scattering operation to already filtered signals [7]. In [40] it is argued that the scattering operator is a contraction which can provide stability to actions of a small diffeomorphism (see also the next section on Lipschitz continuity).

A.2 A Review of Lipschitz Continuity for Operators

Let $\mathcal{L}(\mathbb{R})$ denote the space of square integrable functions defined on $\mathbb{R}$ and assume that a continuous signal $f \in \mathcal{L}(\mathbb{R})$. An operator $\Phi\colon \mathcal{L}(\mathbb{R}) \to \mathcal{H}$ is a mapping of a signal into a Hilbert space $\mathcal{H}$. Let $T_c f(t) = f(t-c)$ denote the translation of a signal f by some constant $c \in \mathbb{R}$. An operator Φ is called *translation invariant* if $\Phi(T_c f) = \Phi(f)$ for all $f \in \mathcal{L}(\mathbb{R})$ and $c \in \mathbb{R}$. The power spectra of a signal (also known as spectrogram) is an operator that can provide an approximately locally time-translation invariant representation over durations limited by a window [7]. While the spectrogram of a signal can provide local time-translation invariance, Mallat [40] has demonstrated that it does not necessarily provide stability to the action of a small diffeomorphism.

Let $D_\tau f(t) = f(t - \tau(t))$ denote a diffeomorphism of a signal given by a displacement field $\tau(t) \in \mathcal{C}^2(\mathbb{R})$. For example, one can take $\tau(t) = \epsilon t$ with $\epsilon \in \mathbb{R}$ and $\epsilon \to 0$. To preserve stability relative to a small diffeomorphism of a signal, it is sufficient to ensure that the operator Φ is Lipschitz continuous [40, 7]. A translation invariant operator Φ is Lipschitz continuous with respect to actions of $\mathcal{C}^2$-diffeomorphisms if for any compact $\Omega \subset \mathbb{R}$ there exists a constant L such that for all signals $f \in \mathcal{L}(\mathbb{R})$ supported on Ω and all $\tau \in \mathcal{C}^2(\mathbb{R})$ it holds that [for more details see, e.g., 40]

$$\|\Phi(f) - \Phi(D_\tau f)\|_{\mathcal{H}} \le L \,\|\mathbb{I} - D_\tau\|_\infty \|f\| := L \left(\sup_{t \in \Omega} \|\nabla \tau(t)\| + \sup_{t \in \Omega} \|\nabla \nabla \tau(t)\| \right) \|f\| \ .$$

The Lipschitz continuity of operator Φ implies invariance to *local translations* and/or signal warping by a diffeomorphism $\tau(t)$, up to the first and second order deformation terms [40]. The scattering operators [17, 40] covered in the previous section are designed to provide the stability to actions of small diffeomorphisms and approximate time-translation invariance.

B   Neural Networks as Conditional Exponential Family Models

Let $\mathcal{X} \subset \mathcal{L}(\mathbb{R})$ be an instance space containing speech signals (e.g., 200 ms long frames of speech) in its interior and $\mathcal{Y}$ the space of categorical labels (e.g., state ids in hybrid HMM models). Suppose that a set of labeled examples $\{(x_i, y_i)\}_{i=1}^n$ has been drawn independently from a latent Borel probability measure defined on $\mathcal{X} \times \mathcal{Y}$. We assume that the conditional probability of a label $y \in \mathcal{Y}$ given an instance $x \in \mathcal{X}$ can be approximated with a conditional exponential family model [6, 33]

$$p(y \mid x, \theta, W) = \frac{\exp\left(\theta^\top \phi(x, y \mid W)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\theta^\top \phi(x, y' \mid W)\right)} \ ,$$

where $\theta \in \Theta$ is a parameter vector defining the so called *softmax probabilities* and $\phi(x, y \mid W)$ is a sufficient statistic of $y \mid x$, defined with some set of network parameters W. The set of parameters W includes the filter parameterization as well as convolutional and fully connected blocks of our model, up to the final softmax output block (see Figure 1 for more details). Typically, the sufficient statistic of the model is selected such that $\phi(x, y \mid W) = \mathrm{vec}(\mathbf{e}_y \, \hat{\phi}(x \mid W)^\top)$, where $\mathbf{e}_y^\top$ is the so called *one-hot* vector with one at position of the categorical label y and zero elsewhere, and $\hat{\phi}(x \mid W)$ is a sufficient statistic of $x \in \mathcal{X}$ modeled by the neural network. Denoting all the parameters of the neural architecture with $\Delta = (\theta, W)$ and taking some prior distribution on Δ, we obtain that the posterior distribution of the network parameters conditioned on some set of labeled examples satisfies

$$\log p(\Delta \mid X_n, Y_n) \propto \log p_r(\Delta \mid \eta) + \sum_{i=1}^n \log p(y_i \mid x_i, \Delta) \ .$$

where $X_n = \{x_i\}_{i=1}^n$, $Y_n = \{y_i\}_{i=1}^n$, and $p_r(\Delta \mid \eta)$ is the prior with hyper-parameters η.

C   Gauss–Hermite Quadrature

Theorem 3. *(Abramowitz and Stegun [1]) For a univariate function h and an integral*

$$\mathcal{J} = \int_{-\infty}^{\infty} h(u) \exp\left(-u^2\right) \mathrm{d}u \ ,$$

the Gauss-Hermite approximation of order s satisfies $\mathcal{J} \approx \sum_{i=1}^s w_i h(u_i)$, where $\{u_i\}_{i=1}^s$ are the roots of the physicist's version of the Hermite polynomial $H_s(u) = (-1)^s \exp\left(u^2\right) \frac{\mathrm{d}^s}{\mathrm{d}u^s} \exp\left(-u^2\right)$ and the corresponding weights $\{w_i\}_{i=1}^s$ are given by $w_i = 2^{s-1} s! \sqrt{\pi} / s^2 H_{s-1}(u_i)^2$.

The Gauss–Hermite approximation of order s is exact and, thus, optimal for all polynomials of degree $2s - 1$ or less [1]. For functions $h \in \mathcal{C}^{2s}$, the error of the Gauss–Hermite quadrature is given by [65]

$$\mathcal{E}_s(h) = \int_{-\infty}^{\infty} h(u) \exp\left(-u^2\right) \mathrm{d}u - \sum_{i=1}^{s} w_i h(u_i) = \frac{s! \cdot \sqrt{\pi}}{2^s \cdot (2s)!} h^{(2s)}(\hat{u}) \,, \tag{3}$$

where $\hat{u} \in (-\infty, \infty)$. Xiang and Bornemann [72] have studied convergence rates of the Gaussian quadrature for functions of limited regularity. The regularity of an integrand is expressed via the decay rate of its expansion coefficients in the basis formed by the Chebyshev polynomials of the first kind. In particular, if the expansion coefficients $a_i \in \mathcal{O}(i^{-p-1})$ for some $p > 0$ (where a_i corresponds to the Chebyshev polynomial of the i-th degree) then the error of the quadrature approximation of order s can be upper bonded by $\mathcal{O}(s^{-p-1})$ for $p > 2$. For $0 < p < 2$, on the other hand, the guaranteed convergence rate is sightly slower and can be upper bounded by $\mathcal{O}(s^{-3p/2})$.

D   Proofs

Proposition 1. KL *divergence between a Gaussian distribution with the dropout parametrization of variance and a log-scale uniform prior can be approximated by*

$$\mathrm{KL}\left(q \,||\, p_{r,\mathrm{lsu}}\right) \approx -1/2 \log \alpha + 1/\sqrt{\pi} \sum_{i=1}^{s} w_i \log |v_i| + \mathrm{const.} \quad \text{with} \quad v_i = \sqrt{2\alpha} u_i + 1$$

and where $\{u_i\}_{i=1}^{s}$ are the roots of the Hermite polynomial with quadrature weights $\{w_i\}_{i=1}^{s}$.

Proof. From Kingma et al. [36, Appendix C], we know that the Kullback–Leibler divergence term is given by

$$\mathrm{KL}\left(q \,||\, p_{r,\mathrm{lsu}}\right) = \mathbb{E}_{\mathcal{N}(\epsilon|1,\alpha)}\left[\log |\epsilon|\right] - \frac{1}{2} \log \alpha + \mathrm{const.}$$

The expectation with respect to the Gaussian random variable ϵ can be re-written as

$$\mathbb{E}_{\mathcal{N}(\epsilon|1,\alpha)}\left[\log |\epsilon|\right] = \frac{1}{\sqrt{2\pi\alpha}} \int \exp\left(-\frac{(\epsilon - 1)^2}{2\alpha}\right) \log |\epsilon| \, \mathrm{d}\epsilon = $$

$$\frac{1}{\sqrt{\pi}} \int \log\left|\sqrt{2\alpha} t + 1\right| \exp\left(-t^2\right) \mathrm{d}t \,.$$

The result now follows from Theorem 3 by taking $h(t) = \log\left|\sqrt{2\alpha} t + 1\right|$. $\qquad \square$

Proposition 2. KL *divergence between a Gaussian distribution with the dropout parametrization of variance and a scale-mixture prior can be approximated by*

$$\mathrm{KL}\left(q \,||\, p_{r,\mathrm{sm}}\right) \approx -\log\sqrt{2\pi\alpha\mu^2} - 1/\sqrt{\pi} \sum_{i=1}^{s} w_i \log p_{r,\mathrm{sm}}(v_i) - 1/2 \; \text{with} \; v_i = \left(\sqrt{2\alpha} u_i + 1\right) \mu$$

where $\{u_i\}_{i=1}^{s}$ are the roots of the Hermite polynomial with corresponding quadrature weights $\{w_i\}_{i=1}^{s}$, α and μ are variational parameters, and $p_{r,\mathrm{sm}}$ is some scale-mixture prior distribution.

Proof. We can re-write the Kullback–Leibler divergence term as

$$\mathrm{KL}\left(q \,||\, p_{r,\mathrm{sm}}\right) = \int q(u) \log q(u) \, \mathrm{d}u - \int q(u) \log p_{r,\mathrm{sm}}(u) \, \mathrm{d}u = -H(q) - \mathbb{E}_q\left[\log p_{r,\mathrm{sm}}(u)\right],$$

where $H(q)$ denotes the entropy of the univariate Gaussian distribution given by

$$q(u) = \frac{1}{\sqrt{2\pi\alpha\mu^2}} \exp\left(-\frac{(u - \mu)^2}{2\alpha\mu^2}\right) \,.$$

As the entropy of a Gaussian distribution defines an analytically tractable integral [e.g., see 37, 56], we have that the entropy of q is given by

$$H\left(q\right) = \log\sqrt{2\pi\alpha\mu^2} + \nicefrac{1}{2}\,.$$

On the other hand, the expected log-likelihood of the scale-mixture prior can be approximated using the Gauss-Hermite quadrature by observing that

$$\mathbb{E}_q\left[\log p_{r,\mathrm{sm}}\left(u\right)\right] = \quad \frac{1}{\sqrt{2\pi\alpha\mu^2}}\int \exp\left(-\frac{\left(u-\mu\right)^2}{2\alpha\mu^2}\right)\log p_{r,\mathrm{sm}}\left(u\right)\,\mathrm{d}u =$$

$$\frac{1}{\sqrt{\pi}}\int \log p_{r,\mathrm{sm}}\left(\sqrt{2\alpha\mu^2}t + \mu\right)\exp\left(-t^2\right)\,\mathrm{d}t\,.$$

The result now follows from Theorem 3 by taking $h\left(t\right) = \log p_{r,\mathrm{sm}}\left(\sqrt{2\alpha\mu^2}t + \mu\right)$. $\qquad\square$

E Detailed Results and Experimental Setup

The data splits (training/development/test) originate from the Kaldi framework [53]. In all the experiments, we train a context dependent hybrid HMM model based on frame labels (i.e., HMM state ids) generated using a triphone model from Kaldi with 25 ms frames and 10 ms stride between the successive frames. In the pre-processing step, we assign the Kaldi frame label to a 200 ms long segment of raw speech centered at the original Kaldi frame (keeping 10 ms stride between the successive frames of raw speech). A similar choice of input raw-speech frame length has been reported in [7] and [57]. After completion of training, we take the resulting log-posterior scores (i.e., log of the ratios between likelihoods coming from the neural network and corresponding class priors) and pass them to Kaldi decoding to obtain the reported word error rates. We note here that on the TIMIT dataset the Kaldi recipe computes the error as the number of miss-classified phonemes, while the model trains against HMM pdf-state ids. To be consistent with our baselines for waveform-based speech recognition on TIMIT, we generate frame labels using the DNN triphone model and decoding configuration from [57]. For AURORA4, on the other hand, we generate frame labels using the standard GMM triphone model and default decoder configuration from KALDI. In total, there are 1936 and 3408 categorical labels (i.e., HMM state ids) for TIMIT and AURORA4, respectively.

We train our models using the approach described in Section 3. In all the experiments, the minibatch size was set to 256 samples. For our deterministically trained baselines, we have tried two batch sizes 256 and 128; reporting the better of the two error rates in our tables. For DETERMINISTIC PARZNETS (illustrated in Figure 2), we have employed the same architecture as in Figure 1 with the addition of standard Bernoulli dropout layers [29] after each RELU activation prior to the softmax output block. This is a standard procedure for regularization of deterministic neural networks. The feature extraction parameters involving Parzen filters and convolution layers that synthesize features across filtered signals are optimized using the RMSPROP algorithm [66] with initial learning rate 0.0008. The fully connected blocks have been optimized using the standard stochastic gradient descent with initial learning rate 0.08. This combination of optimization algorithms (with all the blocks trained jointly) has been found to be the most effective, confirming the findings in [57]. Alternative algorithms that were tried and found to be too aggressive (low training error but not as good generalization) were ADAM [35], NADAM [19], and SGD with momentum. The learning rates were decreased by a factor of $\nicefrac{1}{2}$ if at the end of an epoch the relative improvement in validation error was below a specified threshold (e.g., 0.1% for the frame classification error). Moreover, if the validation error degraded the training would continue using the model from the previous epoch (learning rates would again be decreased by $\nicefrac{1}{2}$). We terminate the training process after at most 25 epochs or upon observing no improvement in the validation error for 3 successive epochs. The training procedure and the Bayesian backpropagation components have been implemented using the MXNET package for PYTHON.

In previous work [63, 45] it has been established that for some priors stochastic variational inference tends to trim too many parameters in the early stages of the training. To address this issue, [63] has proposed to rescale the Kullback–Leibler regularization term with a hyperparameter ρ_t such that $\rho_{t+1} = \min\{1, \rho_t + c\}$ with $\rho_0 = 0$ and some constant $0 < c < 1$ (e.g., $c = 0.2$), and where t denotes the epoch number (starting from $t = 0$). We have followed this heuristic in all of our experiments and observed an improvement in accuracy. Following the findings in [42], we have also
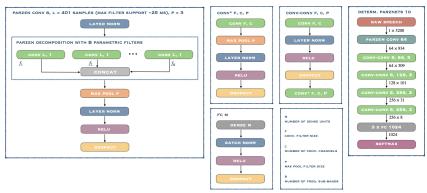
Figure 2: The figure provides a schematic for DETERMINISTIC PARZNETS with 1D convolutional operators. This architecture is a deterministic counterpart for the PROBABILISTIC PARZNETS from Figure 1

considered two notions of validation error in our preliminary experiments: classification error of raw-speech frames and entropy regularized log-loss [42]. The empirical results from [42] indicate that the latter error correlates better with the token error rate of continuous speech recognition. Indeed, our best results have been obtained using the entropy regularized log-loss as the validation objective. Just as in [11], we have observed an improvement in accuracy for models trained using batch-specific importance weighting of the divergence term. However, the cooling schedule proposed in [11, Eq. 9] was too strong for the datasets considered here because of the much larger number of batches. To address this, we have replaced base 2 proposed in [11] with another constant, computed such that the minimal importance weight is equal to machine precision for 32-bit floating point arithmetics. In addition to these findings, we have also observed that in some cases the optimization (overly) focuses on the maximization of the log-likelihood for the already correctly classified speech frames. To mitigate this and ensure that the optimization objective is always bounded, we have transformed the log-softmax probabilities (denoted with p) via

$$\log p \quad \rightarrow \quad \log\left((1 - 2\kappa)\, p + \kappa\right) \;, \tag{4}$$

with κ denoting a small jitter constant (e.g, $\kappa = 10^{-8}$).

| | VI – LOG-SCALE UNIFORM | | | | VI – SCALE MIXTURE | |
| SAMPLE | SQUARED EPANECHNIKOV | | | GAUSS | SQUARED EPANECHNIKOV | |
	MEL-FILT (STATIC)	KL (HG QUAD)	KL (MOLCH. ET AL)	KL (HG QUAD)	KL (HG QUAD)	KL (MCMC)
DEV	15.02 ($\pm$0.26)	14.95 ($\pm$0.14)	**14.77** ($\pm$0.15)	14.83 ($\pm$0.13)	15.64 ($\pm$0.11)	15.58 ($\pm$0.20)
TEST	16.95 ($\pm$0.25)	**16.52** ($\pm$0.22)	16.63 ($\pm$0.23)	16.60 ($\pm$0.22)	17.41 ($\pm$0.17)	17.56 ($\pm$0.16)

Table 3: The table reports the average phoneme error rates (standard deviations are provided in the brackets), obtained using probabilistic PARZNETS 1D and Gaussian mean field (variational) inference on the TIMIT dataset.

E.1 The Effects of Modulation Filtering Learning on Accuracy

In the first set of experiments, our goal is to demonstrate that filter optimization can be statistically significantly more effective than static filtering. To this end, we train two operators with identical architecture (see Figure 1) using variational inference with the Kullback–Leibler divergence term approximated using the Hermite–Gauss (HG) quadrature: *i)* operator with fixed/static Parzen filters initialized just as in Mel-frequency coefficients (denoted with MEL-FILT in Tables 3 and 4), and *ii)* joint filter and operator learning proposed in this work (see HG QUAD under log-scale uniform prior VI in Tables 3 and 4). The Parzen filters of the second operator are initialized using Mel-frequencies, just as in the static filtering operator. To assess whether a method performs statistically significantly better than the other on TIMIT, we perform the paired Welch t-test [70] based on 5 repetitions of the experiment. The t-test indicates that filter optimization is with 90% confidence statistically significantly better than static filtering. AURORA4 is a much larger dataset than TIMIT and repeated training is time consuming and expensive. However, the dataset contains 14 different test samples and this allows us to employ the Wilcoxon signed rank test [71, 18] to establish whether an approach is statistically significantly better than the other. The test indicates that filter learning is with 95% confidence statistically significantly better than static filtering on AURORA4 (e.g., see Table 4).

We conclude by demonstrating that the optimization of filters changes the initial distribution of modulation frequencies and bandwidths. Figure 3 provides a comparison of kernel density estimators

CONDITIONS	VI – LOG-SCALE UNIFORM				VI – SCALE MIXTURE	
	8 X CNN			10 X CNN	8 X CNN	
	MEL-FILT STATIC	KL HG QUAD	KL MOLCH. ET AL	KL HG QUAD	KL HQ QUAD	KL MCMC
A. SAME MICROPHONE						
CLEAN (A)	3.05	2.88	2.84	2.78	3.12	**2.71**
B. SAME MICROPHONE						
CAR	3.29	3.34	3.14	**3.10**	3.29	3.25
BABBLE	4.63	4.33	4.84	**4.26**	4.54	4.84
RESTAURANT	6.46	**6.00**	6.18	6.54	6.65	6.37
STREET	5.87	5.87	5.88	**5.70**	6.22	6.16
AIRPORT	4.76	4.45	4.58	**4.43**	4.78	4.61
TRAIN	6.41	6.33	**6.30**	6.35	**6.30**	6.35
AVERAGE (B)	5.24	**5.05**	5.15	5.06	5.30	5.26
C. DIFFERENT MICROPHONES						
CLEAN (C)	5.90	5.59	6.02	**5.27**	6.09	5.96
D. DIFFERENT MICROPHONES						
CAR	9.79	9.30	9.36	**9.10**	9.84	10.14
BABBLE	15.84	15.41	16.01	**14.78**	16.07	16.16
RESTAURANT	20.08	20.77	21.39	**19.56**	21.15	21.24
STREET	17.31	**16.80**	17.71	17.28	17.65	18.61
AIRPORT	14.70	13.88	14.65	**13.30**	14.70	14.94
TRAIN	17.43	**16.99**	17.49	17.07	17.64	17.90
AVERAGE (D)	15.86	15.53	16.10	**15.18**	16.18	16.50
AVERAGE (ALL)	9.68	9.42	9.74	**9.25**	9.86	9.95

Table 4: The table reports the word error rates obtained using different test samples from AURORA4.
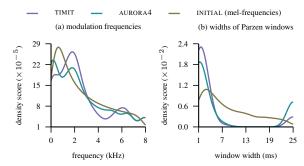

Figure 3: This figures compares the initial distributions of modulation frequencies and bandwidths to those at the end of the training process.

for modulation frequencies and filter bandwidths. From the figure, it is evident that the initial and optimized distributions are quite different for filter bandwidths on both datasets. Moreover, there is an interesting difference between the distributions of modulation frequencies between TIMIT and AURORA4 datasets, which might be due to multi-condition training and various noise conditions characteristic to AURORA4.

E.2 The Effectiveness of the Gauss–Hermite Approximation

Having established that filter optimization can be significantly better than static filtering, we proceed to show that Hermite–Gauss quadrature is an effective mean for the approximation of the Kullback–Leibler divergence term acting as a regularizer in variational inference. In particular, we compare the operators learned via variational inference and currently employed strategies for approximation of the Kullback–Leibler divergence term defined using the log-scale uniform [45] and scale mixture priors [11]. Table 3 (see Squared Epanechnikov modulation filters, TEST sample) provides the results on TIMIT and shows that the approximation based on the Hermite–Gauss quadrature is on average better than currently employed approximation schemes. However, the Welch t-test does not show a statistically significant improvement of the Hermite–Gauss quadrature over the alternatives on this dataset. Table 4 summarizes our results on AURORA4 and demonstrates a significant improvement over the baselines when using the Hermite–Gauss quadrature to approximate the Kullback–Leibler divergence term. More specifically, the Wilcoxon signed rank test in the case of log-scale uniform prior shows that the approximation based on the Hermite–Gauss quadrature is with 95% confidence statistically significantly better than the state-of-the-art approximation from [45].

SAMPLE	VI – LOG-SCALE UNIFORM				VI – SCALE MIXTURE	
	SQUARED EPANECHNIKOV			GAUSS	SQUARED EPANECHNIKOV	
	MEL-FILT (STATIC)	KL (HG QUAD)	KL (MOLCH. ET AL)	KL (HG QUAD)	KL (HG QUAD)	KL (MCMC)
DEV	16.12 ($\pm$0.28)	16.02 ($\pm$0.15)	**15.86** ($\pm$0.15)	15.98 ($\pm$0.13)	16.84 ($\pm$0.09)	16.76 ($\pm$0.21)
TEST	18.06 ($\pm$0.29)	**17.76** ($\pm$0.21)	17.88 ($\pm$0.25)	17.80 ($\pm$0.23)	18.54 ($\pm$0.15)	18.76 ($\pm$0.17)

Table 5: The table reports the average phoneme error rates (standard deviations are provided in the brackets) for continuous speech recognition tasks on TIMIT, obtained using the proposed model and Gaussian mean field (variational) inference. The only difference compared to results in Table 3 is that Kaldi decoding has been performed with the script *score.sh* originating from Kaldi (see Appendix G).

F   Model and Initialization Scheme

We initialize the centers of Parzen filters by keeping them equidistant in the Mel-scale. We limit the filter lengths in time domain such that the width of a Parzen window is at least 1 ms and at most 25 ms long (note that for Epanechnikov filters the time-domain filter has finite support, whereas the Gaussian filter has infinite support). The center frequency of a Parzen filter was bounded/clipped so that the minimal possible frequency is 50 Hz and the maximal one is 7950 Hz. The Kullback–Leibler divergence term was re-scaled as in Sønderby et al. [63] with $c = 0.2$ (see also the definition of the scaling hyperparameter ρ_t in Section 5). The dropout parameter α is stored in the log-form and the initial value across blocks (apart from normalization and softmax blocks) was set to -3.0, which corresponds to variational standard deviation of $\approx 0.22\,|\mu|$, where μ denotes the variational mean of a network parameter. The parameter α was also bounded/clipped as in Molchanov et al. [45] so that the minimal value is 0.0001 and the maximal one is set to 16. Moreover, for normalization layers the prior parameter α was set to value close to zero because dropout is rarely applied to these network blocks (to the best of our knowledge). The fully connected layers were initialized by sampling uniformly at random from the interval $(-0.01/\sqrt{p+q}, 0.01/\sqrt{p+q})$, where p and q denote the number of inputs and outputs corresponding to such a block. The bias parameters corresponding to fully connected blocks are initialized with zero-vectors. The mean and scale parameters in normalization layers [8, 32] were initialized to zero and one, respectively. The convolution parameters are initialized by sampling uniformly at random from the interval $(-1/\sqrt{r}, 1/\sqrt{r})$, where r denotes the total number of parameters in a convolution filter (the same interval and sampling strategy was used to initialize the convolution bias parameters).

We have placed zero-mean priors on the weights of fully connected blocks, convolution filters, as well as on the parameters for means and scales (parameterized as $1 - \text{scale}$) of normalization blocks. For the weights of Parzen filters, on the other hand, we have opted for the prior mean to be equal to the initial values of the parameter filters (the variances are scaled-means, just as in dropout posteriors).

The best results with scale-mixture priors were obtained using the following combination of parameters: $\lambda = 0.25$, $\eta_1 = 0.0005$, and $\eta_2 = 1.0$ or $\eta_2 = 0.75$.

G   Scoring Script

Table 5 provides a summary for a series of experiments on TIMIT, decoded using the original *score.sh* script from the Kaldi library. Ravanelli and Bengio [57] have for the purpose of the *pytorch-kaldi* framework modified the default *score.sh* script for TIMIT decoding as follows:

pytorch-kaldi score.sh [57]
local/timit_norm_trans.pl -i $data/stm -m $phonemap -from 48 -to 39 | sed 's: sil: (sil):g' > $dir/scoring/stm_39phn

Kaldi score.sh
local/timit_norm_trans.pl -i $data/stm -m $phonemap -from 48 -to 39 >$dir/scoring/stm_39phn.

In our experiments, we have found that this modification consistently improves the accuracy by 1-1.2%. The best phoneme error rate after 5 simulations of our model with the original Kaldi scoring script was 17.4%.