Sampling-free Uncertainty Estimation in Gated Recurrent Units with Applications to Normative Modeling in Neuroimaging

Seong Jae Hwang[*1]
sjh@cs.wisc.edu

Ronak R. Mehta[*1]
ronakrm@cs.wisc.edu

Hyunwoo J. Kim[*5]
hyunwoojkim@korea.ac.kr

Sterling C. Johnson[3,4,6]
scj@medicine.wisc.edu

Vikas Singh[2,1]
vsingh@biostat.wisc.edu

Abstract

There has recently been a concerted effort to derive mechanisms in vision and machine learning systems to offer uncertainty estimates of the predictions they make. Clearly, there are benefits to a system that is not only accurate but also has a sense for when it is not. Existing proposals center around Bayesian interpretations of modern deep architectures – these are effective but can often be computationally demanding. We show how classical ideas in the literature on exponential families on probabilistic networks provide an excellent starting point to derive uncertainty estimates in Gated Recurrent Units (GRU). Our proposal directly quantifies uncertainty *deterministically*, without the need for costly sampling-based estimation. We show that while uncertainty is quite useful by itself in computer vision and machine learning, we also demonstrate that it can play a key role in enabling statistical analysis with deep networks in neuroimaging studies with normative modeling methods. To our knowledge, this is the first result describing *sampling-free* uncertainty estimation for powerful sequential models such as GRUs.

1 INTRODUCTION

Recurrent Neural Networks (RNNs) have achieved state-of-the-art performance in various sequence prediction tasks such as machine translation (Wu et al., 2016; Jozefowicz et al., 2016), speech recognition (Hinton et al., 2015; Amodei et al., 2016), language models (Cho et al., 2014) as well as medical applications (Jagannatha and Yu, 2016; Esteban et al., 2016). For sequences with long term dependencies, popular variants of RNN such as Long-Short Term Memory (Gers et al., 1999) and Gated Recurrent Unit (Chung et al., 2014) have shown remarkable effectiveness in dealing with the vanishing gradients problem and have been successfully deployed in a number of applications.

Point estimates, confidence and consequences. Despite the impressive predictive power of RNN models, the predictions rely on the "point estimate" of the parameters. The confidence score can often be overestimated due to overfitting (Fortunato et al., 2017) especially on datasets with insufficient sample sizes. More importantly, in practice, without acknowledging the level of uncertainty about the prediction, the model cannot be entirely trusted in mission critical applications. Unexpected performance variations with no sensible way of anticipating this possibility may also be a limitation in terms of regulatory compliance. When a decision made by a model could result in dangerous outcomes in real-life tasks such as an autonomous vehicle not detecting a pedestrian, missing a disease prediction due to some artifacts in a medical image, or radiation therapy misplanning (Lambert et al., 2011), knowing how 'certain' the model is about its decision can offer a chance to look for alternative solutions such as alerting the driver to take over or recommending a different disease test to prevent undesirable outcomes made by erroneous decisions.

Uncertainty. When operating with predictions involving data and some model, there are mainly two sources of unpredictability. First, there may be uncertainty that arises from an imperfect dataset or observations — *aleatoric* uncertainty. Second, the lack of certainty resulting from the model itself (i.e., model parameters) is called *epistemic* uncertainty (Der Kiureghian and Ditlevsen, 2009). Aleatoric uncertainty comes from the observations *exter-*

[*]Corresponding Authors; [1]Dept. of Computer Sciences, Univ. of Wisconsin-Madison; [2]Dept. of Biostatistics and Medical Informatics, Univ. of Wisconsin-Madison; [3]Dept. of Medicine, Univ. of Wisconsin-Madison; [4]William S. Middleton VA Hospital, Madison; [5]Dept. of Computer Science and Engineering, Korea University; [6]Wisconsin Alzheimer's Disease Research Center, Madison; Appendix is available at `http://pages.cs.wisc.edu/~sjh/`

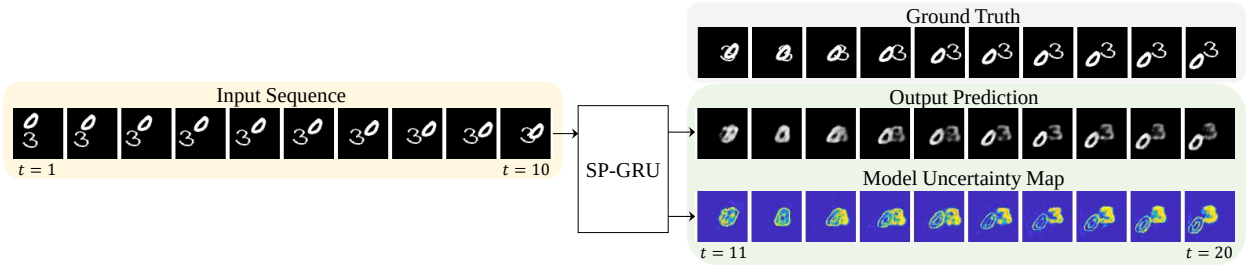

Figure 1: Image sequence prediction with uncertainty. Given the first 10 frames of an input sequence (left), our model SP-GRU makes the **Output Prediction** and the pixel-level **Model Uncertainty Map** where bright regions indicate high uncertainty. SP-GRU estimates the uncertainty *deterministically* without sampling model parameters.

nally such as noise and other factors that cannot typically be inferred systematically. Algorithms instead attempt to calculate *the epistemic uncertainty resulting from the model itself*. This is often also referred to as *model uncertainty* (Kendall and Gal, 2017).

Related work on uncertainty in Neural networks. The importance of estimating the uncertainty aspect of neural networks (NN) has been acknowledged in the literature. Several early ideas investigated a suite of schemes related to Bayesian neural networks (BNN): Monte Carlo (MC) sampling (MacKay, 1992a), variational inference (Hinton and Van Camp, 1993) and Laplace approximation (MacKay, 1992b). More recent works have focused on efficiently approximating posterior distributions to infer predictive uncertainty. For instance, scalable forms of variational inference approaches (Graves, 2011) suggest estimating the evidence lower bound (ELBO) via Monte Carlo estimation to efficiently approximate the marginal likelihood of the weights. Similarly, several proposals have extended the variational Bayes approach to perform probabilistic back propagation with assumed density filtering (Hernández-Lobato and Adams, 2015), explicitly update the weights of NN in terms of the distribution parameters (i.e., expectation) (Blundell et al., 2015), or apply stochastic gradient Langevin dynamics (Welling and Teh, 2011) at large scales. These methods, however, theoretically rely on the correctness of the prior distribution, which has shown to be crucial for reasonable predictive uncertainties (Rasmussen and Quinonero-Candela, 2005) and the strength or validity of the assumption (i.e., mean field independence) for computational benefits. An interesting and different perspective on BNN uncertainty based on Monte Carlo dropout was proposed by Gal et al. (Gal and Ghahramani, 2016), wherein the authors approximate the predictive uncertainty by using dropout (Srivastava et al., 2014) at prediction time. This approach can be interpreted as an ensemble method where the predictions based on "multiple networks" with different dropout structures (Lakshminarayanan et al., 2017) yield estimates for uncertainty. However, while the estimated *predictive uncertainty* is less dependent on the data by using a fixed dropout rate independent from the data, uncertainty estimation on the network parameters (i.e. weights) is naturally compromised since the fixed dropout rates are already imposed on the weights by the algorithm itself. In summary, while the literature is still in a nascent stage, a number of researchers are studying ways in which uncertainty estimates can be derived for deep architectures similar to those from traditional statistical analysis for various applications (Ribeiro et al., 2018; Sedlmeier et al., 2019).

Other gaps in our knowledge. While the above methods focus on predictive uncertainty, most strategies do not explicitly attempt to estimate the uncertainty of all *intermediate* representations of the network such as neurons, weights, biases and so on. Such information is understandably less attractive in traditional applications, where our interest mainly lies in the prediction made by the final output layer. However, RNN-type sequential NNs often utilize not only the last layer of neurons but also directly operate on the intermediate neurons in making a sequence of predictions (Mikolov et al., 2010). Several Bayesian RNNs have been proposed (Lakshminarayanan et al., 2017; Fortunato et al., 2017) but are based on the BNN models described above. Their deployment is not always feasible under practical time constraints for real-life tasks, especially with high dimensional inputs. Also, stochastic RNN models with stochastic layers with deterministic layers (Fraccaro et al., 2016) and stochastic state models for reinforcement learning (Gregor and Besse, 2018) have been proposed, but they do not explicitly estimate the uncertainty of intermediate representations. Further, empirically more powerful variants of RNNs such as LSTMs or GRUs have not been explicitly studied in the literature in the context of uncertainty.

Contributions. Here, our goal is is to enable uncertainty estimation on more powerful sequential neural networks, namely gated recurrent units (GRU), while addressing the issues discussed above in BNNs. To our knowledge, few (if any) other works offer this capability. We propose

a probabilistic GRU, where *all* network parameters follow exponential family distributions.We call this framework the SP-GRU, which operates *directly* on these parameters, inspired in part by an interesting result for non-sequential data (Wang et al., 2016). Our SP-GRU directly offers the following properties: **(i)** The operations within each cell in the GRU proceed only with respect to the natural parameters *deterministically*. Thus, the overall procedure is completely sampling-free. Such a property is especially appealing for sequential datasets with small sample sizes; **(ii)** Because weights and biases and *all intermediate neurons* of SP-GRU can be expressed in terms of a distribution, their uncertainty estimates can be directly inferred from the network itself. **(iii)** We focus on some well-known exponential family distributions (i.e., Gaussian, Gamma) which have nice characteristics that can be appropriately chosen with minimal modifications to the operations depending on the application of interest. **(iv)** We show how SP-GRU can be used on neuroimaging data for detecting early disease progression in an asymptomatic Alzheimer's disease cohort.

2 RECURRENT NEURAL NETWORKS AND EXPONENTIAL FAMILIES

Recurrent Neural Networks. The Gated Recurrent Unit (GRU) and the Long-Short Term Memory (LSTM) are popular variants of RNN where the network parameters are shared across layers. While they both deal with exploding/vanishing gradient issues with *cell* structures of similar forms, the GRU does not represent the cell state and hidden state separately. Specifically, its updates take the following form (order of operation is (1) Reset Gate and Update Gate, (2) State Candidate and (3) Cell State):

$$\textbf{Reset Gate: } r^t = \sigma(W_r x^t + b_r)$$
$$\textbf{Update Gate: } z^t = \sigma(W_z x^t + b_z)$$
$$\textbf{State Candidate: } \hat{h}^t = \tanh(U_{\hat{h}} x^t + W_{\hat{h}}(r^t \odot h^{t-1}) + b_{\hat{h}})$$
$$\textbf{Cell State: } h^t = (1 - z^t) \odot \hat{h}^t + z^t \odot h^{t-1}$$

where $W_{\{r,z,\hat{h}\}}$ and $b_{\{r,z,\hat{h}\}}$ are the weights and biases respectively for their corresponding updates. Typical implementations of both GRUs and LSTMs include an output layer outside of the cell to produce the desired outputs. However, they do not naturally admit more than point estimates of hidden states and outputs.

Exponential Families in Networks. In statistics, the properties of distributions within *exponential families* have been very well studied.

Definition 1 *Let $x \in X$ be a random variable with probability density/mass function (pdf/pmf) f_X. Then f_X is an exponential family distribution if*

$$f_X(x|\eta) = h(x) \exp(\eta^T T(x) - A(\eta)) \qquad (1)$$

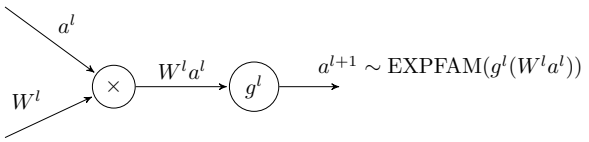

Figure 2: A single exponential family neuron. W^l are learned, and the output is a sample generated from the exponential family defined by $g^l(W^l a^l)$.

with natural parameters η, base measure $h(x)$, and sufficient statistics $T(x)$. Constant $A(\eta)$ (log-partition function) ensures that the distribution normalizes to 1.

Common distributions (e.g., Guassian, Bernoulli, Gamma) can be written in this unified 'natural form' with specific definitions of $h(x)$, $T(x)$ and $A(\eta)$ (e.g., Gaussian distribution with $\eta = (\alpha, \beta)$, $T(x) = (x, x^2)$ and $h(x) = 1/\sqrt{2\pi}$).

Two key properties of this family of distributions have led to their widespread use: (1) their ability to summarize arbitrary amounts of data $x \sim f_X$ through only their sufficient statistics $T(x)$, and (2) their ability to be efficiently estimated either directly through a closed form maximum likelihood estimator or a convex function with convex constraints.

Deep Exponential Families (DEFs) (Ranganath et al., 2015) explicitly models the output of any given layer as a random variable, sampled from an exponential family defined by natural parameters given by the linear product of the previous layer's output and a learnable weight matrix (see Fig. 2). While this formulation leads directly to distributions over hidden states and model outputs, we have not learned distributions over the *model parameters*. Computational feasibility is also neglected: the variational inference procedure used for learning these DEFs requires *Monte Carlo sampling at each hidden state many times for every input sample* (the cost of running just *text* experiments was $40K as stated by the authors). This also becomes a concern in many biomedical applications (e.g., medical imaging) where the model size grows proportionally to the dimensionality of data which often ranges from thousands to millions.

3 SAMPLING-FREE PROBABILISTIC NETWORKS

We now describe a probabilistic network fully operating on a set of natural parameters of exponential family distributions in a *sampling-free* manner. Inspired by a result from a few years back (Wang et al., 2016), the learning process, similar to traditional NNs, is deterministic yet still captures the probabilistic aspect of the output *and the network itself*, purely as a byproduct of typical NN procedures (i.e., backpropagation).

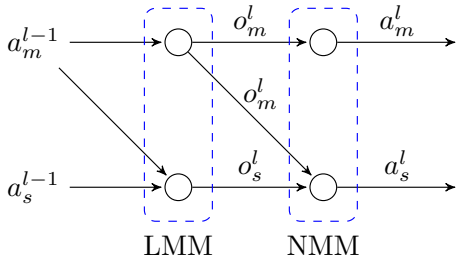

Figure 3: Linear Moment Matching (LMM) and Nonlinear Moment Matching (NMM) are performed at the weights/bias sums and activations respectively.

Unlike the probabilistic networks mentioned before, our GRU performs forward propagation in a series of *deterministic* linear and nonlinear transformations on the distribution of weights and biases. Throughout the entire process, all operations only involve distribution parameters while maintaining their desired distributions after every transformation. We focus on three exponential family distributions with two natural parameters: Gaussian, Gamma and Poisson.

3.1 LINEAR TRANSFORMATIONS

We describe the linear transformation on the input vector x with a matrix W of weights and a vector b of biases in terms of their natural parameters. We first apply the *mean-field* assumption on each of the weights and biases based on their individual distribution parameters α and β as $p(W|W_\alpha, W_\beta) = \prod_{i,j} p(W(i,j) \mid W_\alpha(i,j), W_\beta(i,j))$ where $\{W_\alpha, W_\beta\}$ and $\{b_\alpha, b_\beta\}$ are the model parameters. Thus, analogous to the linear transformation $o = Wa + b$ in ordinary neural networks on the previous layer output (or an input) a with W and b, our network operates purely on (α, β) to compute $\{o_\alpha, o_\beta\}$.

After each linear transformation, it is necessary to preserve the 'distribution property' of the outputs (i.e., o_α and o_β still define the same distribution) throughout the forward propagation so that the intermediate nodes and the network itself can be naturally interpreted in terms of their distributions. Thus, we *cannot* simply mimic the typical linear transformation on a_β and compute $o_\beta = W_\beta a_\beta + b_\beta$ if we want o_β to still be able to preserve the distribution (Wang et al., 2016).

We perform a second order moment matching on the mean and variance of the distributions. The mean m and variance s can easily be computed with an appropriate function $g(\cdot,\cdot)$ which maps $g : (\alpha, \beta) \to (m, s)$ for each exponential family distribution of interest (i.e., $g(\alpha, \beta) = (-\frac{\alpha+1}{\beta}, \frac{\alpha+1}{\beta^2})$ for a Gamma distribution). Thus, we compute the (m, s) counterparts of all the (α, β)-based components (i.e., $(o_m, o_s) = g(o_\alpha, o_\beta)$).

Using the linear output before the activation function, we can now apply Linear Moment Matching (LMM) on (1) the mean a_m following the standard linearity of random variable expectations and (2) the variance a_s as follows:

$$o_m = W_m a_m + b_m$$
$$o_s = W_s a_s + b_s + (W_m \odot W_m) a_s + W_s (a_m \odot a_m)$$

where $\odot$ is the Hadamard product. Then, we invert back to $(o_\alpha, o_\beta) = g^{-1}(o_m, o_s)$. For the exponential family distributions involving at most two natural parameters, matching the first two moments is sufficient.

3.2 NONLINEAR TRANSFORMATIONS

The next key step in NNs is the element-wise nonlinear transformation where we want to apply a nonlinear function $f(\cdot)$ to the linear transformation output o parametrized by $\eta = (o_\alpha, o_\beta)$. This is equivalent to a general random variable transformation given the probability density function (pdf) p_O for O to derive the pdf p_A of A transformed by $a = f(o)$: $p_A(a) = p_O(f(o))|f'(o)|$.

However, well-known nonlinear functions $f(\cdot)$ such as sigmoids and hyperbolic tangents cannot directly be utilized on (o_α, o_β) because the resulting $a = f(o)$ may not be from the same exponential family distribution. Thus, we perform another second order moment matching in terms of mean o_m and variance o_s via Nonlinear Moment Matching (NMM). Ideally, we need to marginalize over a distribution of o given (o_α, o_β) to compute $a_m = \int f(o) p_O(o \mid o_\alpha, o_\beta) do$ and the corresponding variance $a_s = \int f(o)^2 p_O(o \mid o_\alpha, o_\beta) do - a_m^2$ which we map back to (a_α, a_β) with an appropriate bijective mapping function $g(\cdot,\cdot)$. However, when the dimension of o grows, the computational burden of integral calculation becomes incredibly more demanding. The closed form approximations described below can efficiently compute the mean and variance of the activation outputs a_m and a_s (Wang et al., 2016). We show these approximations for sigmoids $\sigma(x)$ and hyperbolic tangents $\tanh(x)$ for a Gaussian distribution, as these will become the critical components used in our probabilistic GRU. Here, we use the fact that $\sigma(x) \approx \Phi(\zeta x)$ where $\Phi(\cdot)$ is a probit function and $\zeta = \sqrt{\pi/8}$ is a constant. Then, we can approximate the sigmoid functions for a_m and a_s as

$$a_m \approx \sigma_m(o_m, o_s) = \sigma\left(\frac{o_m}{(1 + \zeta^2 o_s)^{\frac{1}{2}}}\right)$$
$$a_s \approx \sigma_s(o_m, o_s) = \sigma\left(\frac{\nu(o_m + \omega)}{(1 + \zeta^2 \nu^2 o_s)^{\frac{1}{2}}}\right) - a_m^2$$

where $\nu = 4 - 2\sqrt{2}$ and $\omega = -\log(\sqrt{2}+1)$. The hyperbolic tangent can be derived from $\tanh(x) = 2\sigma(2x) - 1$.

Table 1: SP-GRU operations in mean and variance. $\odot$ and $[A]^2$ denotes the Hadamard product and $A \odot A$ of a matrix/vector A respectively. Note the Cell State does not involve nonlinear operations. See Fig. 4 for the illustration of cell structure.

Operation	**Linear Transformation**	**Nonlinear Transformation**
Reset Gate	$o_{r,m}^t = U_{r,m}x_m^t + W_{r,m}h_m^{t-1} + b_{r,m}$	$r_m^t = \sigma_m(o_{r,m}^t, o_{r,s}^t)$
	$o_{r,s}^t = U_{r,s}x_s^t + W_{r,s}h_s^{t-1} + b_{r,s} + [U_{r,m}]^2 x_s^t$	$r_s^t = \sigma_s(o_{r,m}^t, o_{r,s}^t)$
	$\quad + U_{r,s}[x_m^t]^2 + [W_{r,m}]^2 h_s^{t-1} + W_{r,s}[h_m^{t-1}]^2$	
Update Gate	$o_{z,m}^t = U_{z,m}x_m^t + W_{z,m}h_m^{t-1} + b_{z,m}$	$z_m^t = \sigma_m(o_{z,m}^t, o_{z,s}^t)$
	$o_{z,s}^t = U_{z,s}x_s^t + W_{z,s}h_s^{t-1} + b_{z,s} + [U_{z,m}]^2 x_s^t$	$z_s^t = \sigma_s(o_{z,m}^t, o_{z,s}^t)$
	$\quad + U_{z,s}[x_m^t]^2 + [W_{z,m}]^2 h_s^{t-1} + W_{z,s}[h_m^{t-1}]^2$	
State Candidate	$o_{\hat{h},m}^t = U_{\hat{h},m}x_m^t + W_{\hat{h},m}h_m^{t-1} + b_{\hat{h},m}$	$\hat{h}_m^t = \tanh_m(o_{\hat{h},m}^t, o_{\hat{h},s}^t)$
	$o_{\hat{h},s}^t = U_{\hat{h},s}x_s^t + W_{\hat{h},s}h_s^{t-1} + b_{\hat{h},s} + [U_{\hat{h},m}]^2 x_s^t$	$\hat{h}_s^t = \tanh_s(o_{\hat{h},m}^t, o_{\hat{h},s}^t)$
	$\quad + U_{\hat{h},s}[x_m^t]^2 + [W_{\hat{h},m}]^2 h_s^{t-1} + W_{\hat{h},s}[h_m^{t-1}]^2$	
Cell State	$h_m^t = (1 - z_m^t) \odot \hat{h}_m^t + z_m^t \odot h_m^{t-1}$	Not Needed
	$h_s^t = [(1 - z_s^t)]^2 \odot \hat{h}_s^t + [z_s^t]^2 \odot h_s^{t-1}$	

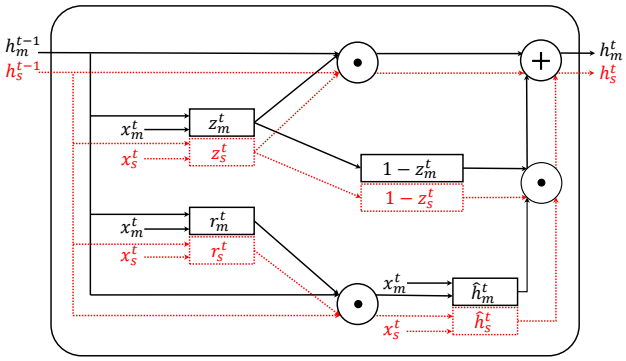

Figure 4: SP-GRU cell structure. Solid lines/boxes and red dotted lines/boxes correspond to operations and variables for mean m and variance s respectively. Circles are element-wise operators.

Note that other common exponential family distributions do not have obvious ways to make such straightforward approximations. Thus, we use an 'activation-like' mapping $f(x) = a - b\exp(-\gamma d(x))$ where $d(x)$ is an arbitrary activation of choice with appropriate constants a, b and γ of > 0. Nonlinear transformations of Gamma and Poisson distributions can then be formulated in closed form as well (e.g., $a = b = \gamma = 1$ is a good choice).

3.3 SAMPLING-FREE PROBABILISTIC GRU

Based on the probabilistic formulations described above, we present *Sampling-free Probabilistic GRU* (SP-GRU). The internal architecture is shown in Fig. 4. Here, we focus on adapting GRU with the sampling-free probabilistic formulation. We express all the variables related to the GRU in Table 1 in terms of their parameters $\eta = (\alpha, \beta)$. For instance, W_r is now expressed *only* in terms of its parameters $W_{r,\alpha}$ and $W_{r,\beta}$ (i.e., two weight matrices). We assume that all of the variables are factorized. Because the GRU consists of a series of operations with linear and nonlinear transformations, we can update each gate by the transformations defined in Table 1.

Assuming that the desired exponential family distribution provides an invertible parameter mapping function $g(\cdot, \cdot)$, we first transform all of the natural parameter variables to means and variances. Then, given an input sequence $x = \{x_m^1, x_s^1\}, \ldots, \{x_m^T, x_s^T\}$, we perform linear/nonlinear transformations with respect to means and variances for each GRU operation (Fig. 4 and Table 1).

The cell state computation does not involve a nonlinear transformation. For an output layer on the hidden states to compute the desired estimate $\hat{y}$, a typical layer can be defined in a similar manner to obtain both $\hat{y}_m$ and $\hat{y}_s$. In the experiments that follow, we add another such layer to compute the mean and variance of the sequence of predictions $\hat{y} = \{y_m^1, y_s^1\}, \{y_m^2, y_s^2\}, \ldots, \{y_m^T, y_s^T\}$.

Extensibility remarks. We note that despite the simplicity of the cell structure of the SP-GRU as shown in Fig. 4, our exponential family adaptation is *not* limited to GRU. For instance, the above formulation can be extended to other variants of RNNs such as LSTMs popularly used in medical applications (Jagannatha and Yu, 2016; Santeramo et al., 2018), flow-based models (Dinh et al., 2016, 2014), and invertible neural networks (Ardizzone et al., 2019).

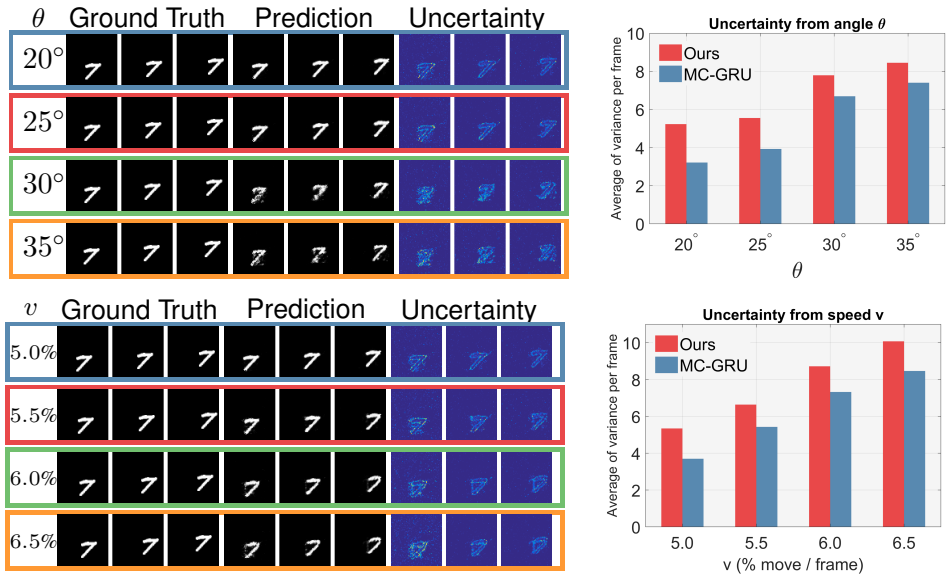

Figure 5: Predictions and uncertainties (sub-frames $\{11, 15, 20\}$ out of full predicted frames $\{11, \ldots, 20\}$) from testing varying deviations from trained trajectories (first of four rows, blue). Top: angle (colors match the paths in Fig. 6). Bottom: speed (colors match the paths in Appendix B.2). Right: [sum of pixel-level variances / frames] using SP-GRU and MC-GRU.

4 EXPERIMENTS

We first perform unsupervised learning of predicting image sequences from the moving MNIST dataset (Srivastava et al., 2015) for intuitive quantitative/qualitative evaluations. Second, we apply our model to a unique neuroimaging dataset, consisting of brain imaging acquisitions from individuals at risk for developing Alzheimer's disease. Models were trained on an NVIDIA GeForce GTX 1080 Ti GPU in TensorFlow with ADAM and an initial learning rate of 0.05, and decay parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$. We use the Gaussian distribution for all setups with the KL divergence between the final output distribution $N(o_m, diag(o_s))$ and the target mini-batch distribution $N(y_m, diag(y_s))$ as the error where y_m and y_s are the ground truth values of the mini-batch samples and their variances (w.r.t. the current mini-batch) respectively. This allows the model to learn both the means and variances.

4.1 UNSUPERVISED SEQUENCE LEARNING OF MOVING MNIST

Goal. For pixel-level tasks, prediction quality can be understood by the uncertainty estimate, i.e., estimated model variance of that pixel. In these experiments, we ask the following questions qualitatively and quantitatively: (1) Given a visually 'good looking' sequence prediction, how can we tell that its trajectory is correct? (2) If it is, can we derive a degree of uncertainty on its prediction?

Setup. The moving MNIST dataset consists of digits moving (randomly or controlled) in a 64×64 image over 20 frames. We split sequences into two halves (first 10 and second 10 frames). Then, we encode the first 10 frames to learn a hidden representation (size 1024) and predict the second 10 frames.

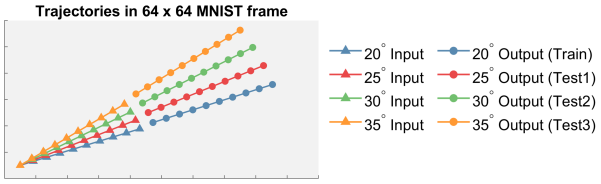

Figure 6: Controlled angle trajectories over 20 frames.

Controlled Paths. We first train our SP-GRU and Monte Carlo dropout GRU (MC-GRU) (Gal and Ghahramani, 2016) with the same number of parameters until they have similar test errors (independent of uncertainty) on simple one-digit MNIST sequences moving in a straight line (blue line in Fig. 6). We then construct three sets of 100 'unfamiliar' samples where each set consists of sequences deviating from the training sequence path (blue path in Fig. 6 with angle $\theta = 20°$ and speed $v = 5.0\%$ of width per frame) with varying angles ($25°$, $30°$, and $35°$ paths in Fig. 6) and speeds (5.5%, 6.0%, and 6.5% of width per frame). See Appendix B for details.

Results. For 'unfamiliar' angles and speeds, the predictions in Fig. 5 look visually sensible, but they do *not* actually follow the ground truth *paths* (e.g., the prediction of $35°$ still follows $20°$ path). We can quantify this directly by the [sum of pixel-level variances / frames] as shown in the right of Fig. 5. While we cannot evaluate

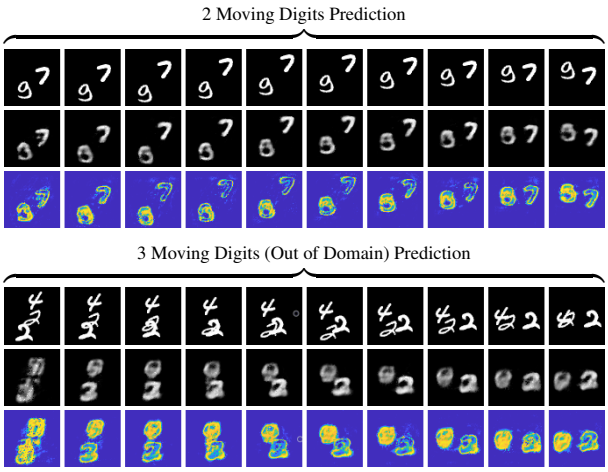

Figure 7: SP-GRU predictor results. Top 3 rows: 2 moving digits (top: ground truth, middle: mean prediction, bottom: uncertainty estimate). Bottom 3 rows: 3 moving digits which are out of domain (i.e., not seen in training).

the relative difference here because 'ground truth uncertainty' is not available for a true absolute comparison, we observe that the *uncertainty increases as the angle/speed deviation increases* for both SP-GRU and MC-GRU.

Computation Speed. From a practical perspective, the uncertainty inference should not sacrifice computational speed, e.g., real-time safety of an autonomous vehicle. With respect to this crucial aspect, SP-GRU greatly benefits from its *sampling-free* procedure: each epoch (30 sequences) takes $\sim$3 seconds while MC-GRU with a Monte Carlo sampling rate of 50 requires $\sim$40 seconds ($>$ 10 times SP-GRU) despite their comparable qualitative and quantitative performance. The MC sampling rate for these methods *cannot* simply be decreased: uncertainty will be underestimated. With SP-GRU, we compute this model uncertainty *in closed form*, without the need for any heavy lifting from large sample analysis.

Random Paths. To demonstrate that SP-GRU does not sacrifice base predictive power, we evaluate SP-GRU on the same setup by (Srivastava et al., 2015) (2 randomly moving digits).

Results. An example of two digit prediction result is illustrated in Fig. 7 (Top 3 rows) which shows quantifiable variance outputs as demonstrated in the controlled paths examples. We note that the mean prediction (middle row of Top 3 rows in Fig. 7) performance is also accurate by comparing our method to previous work in Table 2. SP-GRU with a basic predictor network setup performs comparably or better than other methods that do *not* provide model uncertainty. In these works, model performance often benefits from respective specific network structures: encoder-predictor composite models (Srivastava et al., 2015), generative adversarial networks (Kulharia et al., 2016), and external weight filters (De Brabandere et al., 2016). Further, more advanced models (Cricri et al., 2016) have achieved better results with large, more sophisticated pipelines. Extending SP-GRU to such setups becomes a reasonable modification, providing model uncertainty *without* sacrificing performance.

Table 2: Average cross entropy test loss per image per frame on Moving MNIST.

Model	Test Loss
Srivastava et al. 2015	341.2
Xingjian et al. 2015	367.1
Brabandere et al. 2016	285.2
Ghosh et al. 2016	241.8
SP-GRU (Ours)	277.1

We briefly evaluate how well SP-GRU is able to perform on *out-of-domain* samples (Fig. 7, Bottom 3 rows). Models deployed in real-world settings may not realistically be able to determine if a sample is far from their training distributions. However, with our specific modeling of uncertainty, we would expect that images or sequences distant from the training data will exhibit high variance. We construct sequences of 3 moving digits. Here, future reconstruction is generally quite poor. As has been observed in previous work (Srivastava et al., 2015), the model attempts to hallucinate two digits. Our model is *aware of this issue*: the variance for a large number of pixels is extremely high, *even if the digits overlap*.

Other Methods. Deep Markov Model (DMM) (Krishnan et al., 2017) is a variant of Structured Variational Autoencoders introduced recently that naturally give rise to a probabilistic interpretation of predictions from deep temporal models. However, upon application of this model to Moving MNIST we were unable to obtain any reasonable prediction, across a range of hidden dimension sizes and trajectory complexities, even with significant training time (days vs. hours for SP-GRU). Shown in Fig. 8 are results using a hidden dimension size of 1024 (equal to our setup). We note that the experimental setups described in (Krishnan et al., 2017) are small in dimension and complexity compared to Moving MNIST, and it may be the case that additional technical development with DMMs may lead to promising and comparable uncertainty results.

4.2 NORMATIVE MODELING IN PRECLINICAL NEUROIMAGING DATA

Goal. In a *preclinical cohort* of individuals at risk for developing Alzheimer's disease (AD), effect sizes are small and statistical signal is often weak among those who will and will not go on to develop AD. Even with

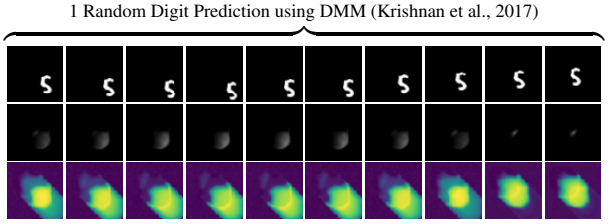

Figure 8: Deep Markov Model results. Compared to our results on a single digit (Fig. 5), the mean and variance estimations using DMM cannot be estimated well on Moving MNIST.

high-dimensional brain imaging data, it is often the case that specific imaging modalities do not lead to significant group differences. Early detection of risk factors associated with the eventual development of AD are of critical importance in facilitating the prevention of onset, and identifying individuals who subtly deviate from expected decline is a required step in that direction. We aim to identify an out-of-domain sample via *normative modeling* (Marquand et al., 2016): Given that we have a SP-GRU model trained on a preclinical cohort, can we *predict with confidence* those individuals who are *at risk*?

Data. Imaging data from 139 individuals was derived from two distinct modalities: Positron emission tomography (PET) and diffusion-weighted MRI (DWI) (Chua et al., 2008). PET imaging is used to determine mean amyloid-plaque burden (^{11}C Pittsburgh Compound B (PiB) radiotracer), known to be strongly associated with AD pathology and often *preceding* observable cognitive decline (Johnson et al., 2014). An individual is deemed *at risk* if the average amyloid burden within specific regions (eight bilateral) is greater than 1.12 (Johnson et al., 2014). DWI captures the diffusion of water through a specific voxel in a brain image; the mean diffusion of water through a *tract* within the brain is a measure of connectivity strength. For each individual, 1761 unique brain connectivities derived from the IIT atlas (Varentsova et al., 2014) are computed from each DWI. Additionally, we have a neuropsychological test score for each individual, the Rey Auditory Visual Learning Test (RAVLT) (Rosenberg et al., 1984) known to be correlated with both amyloid load and structural connectivity. Since our data is cross-sectional, we use RAVLT as our "temporal" analog of cognitive decline.

Model Setup. To generate our sequential training data, we first place *all* individuals into 8 bins based on their RAVLT scores (i.e., 8 evenly ranged intervals between [$\text{RAVLT}_{\text{max}}$, $\text{RAVLT}_{\text{min}}$]). This gives us the sample means and variances of each connectivity in each bin. Then, we generate samples of 1761 connectivities across 8 bins (timepoints) by independently sampling each connectivity in each bin from a normal distribution with the corresponding sample mean and variance. Each sample sequence (1761 variables for each of the 8 timepoints) thus simulates how connectivity may evolve over decreasing RAVLT (modeling potential AD progression) within our preclinical population. See Appendix C for details. We train SP-GRU on the generated samples to predict $t = 5, 6, 7, 8$ given $t = 1, 2, 3, 4$.

Evaluation. We follow existing work in identifying at-risk individuals. Refer to Fig. 9 for the full pipeline. First, after we train our SP-GRU predictor, we generate $N = 100$ new test sequences and predict $t = 5, 6, 7, 8$ given $t = 1, 2, 3, 4$ (Fig. 9 (1)-(2)). Thus, for subject i, time t and connectivity k, we obtain a mean response $\bar{y}_{itk}$ and an expected level of variation σ_{itk}. Note that we also have the true response y_{itk} with a bin-level variance of σ_{ntk}. Then, we compute a *normative probability map* (NPM) per timepoint for each subject and connectivity (Ziegler et al., 2014). We compute Z-scores across timepoints, connectivities, and subjects as $z_{itk} = (y_{itk} - \bar{y}_{itk})/\sqrt{\sigma^2_{itk} + \sigma^2_{ntk}}$ (Fig. 9 (3)). Applying the procedure described in (Marquand et al., 2016) we compute subject-level empirical distributions of all connectivities per timepoint. Then the robust mean of the top 5% of absolute statistics defines the extreme value statistic (EVS) describing that subject (Fig. 9 (4)). Collecting across subjects we fit a generalized extreme value distribution (GED) per time point (Fig. 9 (5)).

Results. We aim to identify those sequences which correspond to individuals deviating from the norm defined by our estimated GED. Using PET mean amyloid burden, we can separate our cohort into two distinct groups, one of which is considered to be 'cognitively healthy', the other to be 'at risk'. Sampling 100 sequences each using the binning above applied to both groups, we can then apply the EVS procedure (i.e., compute EVS following (1)-(4) in Fig. 9 with the same SP-GRU). Then, we use these EVS to identify sequences within those groups which significantly deviate from the overall population (Fig. 9 (6)-(7)). With an $\alpha = 0.01$ cutoff (with the Bonferroni correction) we identify 9 outlier sequences in the cognitively healthy group and 19 in the at risk group. While further scientific analysis is necessary, these results suggest that larger absolute fluctuations in DWI connectivity may be a good indicator for disease risk as measured by amyloid burden. This sets a promising direction in preclinical AD research since brain connectivity is one of the early indicators of AD progression (Greicius et al., 2009; Kim et al., 2015, 2019; Hwang et al., 2019) characterizing the overall integrity of brain (see Appendix C for additional details and discussions).

Remarks. Although this process is feasible with any model providing expected variance, an ideal model needs to possess the following three traits: (i) Strong modeling

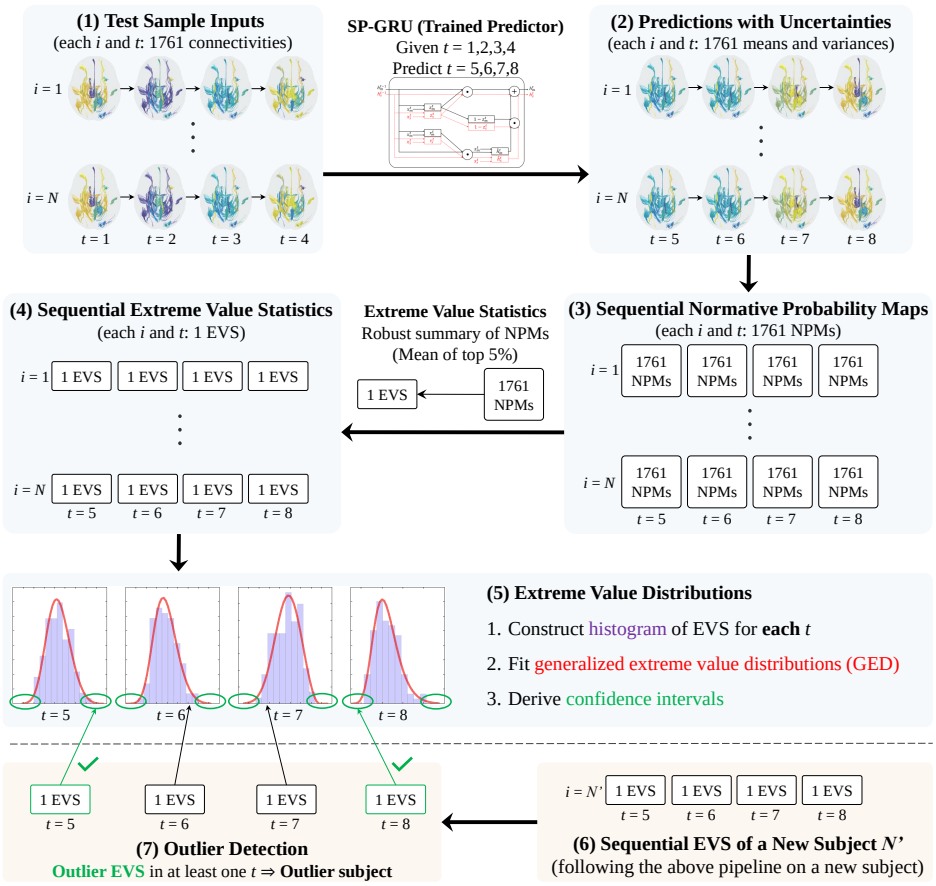

Figure 9: Normative modeling pipeline for preclinical AD. **(1)** Given a set of test inputs ($t = 1, 2, 3, 4$), **(2)** use the pretrained SP-GRU to make mean and variance predictions for each connectivity and $t = 5, 6, 7, 8$. **(3)** Compute NPM for each prediction, and **(4)** derive EVS for each sample i and t. **(5)** Fit GED and construct confidence intervals based on N EVS for each t. **(6)** Given a new sample, derive EVS following (1)-(4), and **(7)** check the confidence intervals from (5) to determine heterogeneity.

ability to capture the subtle underlying abnormality of biomarkers in the early AD stage. (ii) Sequential modeling of longitudinal progression of biomarkers which is often more advantageous due to the variability among cross-sectional samples. (iii) Accurate and practical uncertainty estimation of every variable of interest. Despite the availability of successful recurrent neural network models, they only satisfy (i) and (ii) by construction. Similarly, non-RNN models that satisfy (ii) and (iii) may lack predictive power compared to popular RNNs. Here, we take direct advantage of SP-GRU possessing all three traits in determining the statistic used for detection to overcome the subtle signal in preclinical longitudinal settings which would otherwise be unidentifiable. Also, SP-GRU requires much less prediction time (∼1 seconds) compared to MC-GRU (∼11 seconds, 50 sampling rate) for 100 sequences.

5 CONCLUSION

In this work, we show how uncertainty estimates for a powerful class of sequential models, GRUs, can be derived without compromising either predictive power or computation speed using our SP-GRU. Complementary to the developing body of work on Bayesian perspectives on deep learning, we show how a mix of old and new ideas can enable deriving uncertainty estimates for a powerful class of models, GRUs, while also being easily extensible to other sequential models. Competitive results are first shown on a standard dataset used for sequential models, while offering uncertainty as a natural byproduct. We then demonstrated a direct application of SP-GRU for normative modeling of preclinical Alzheimer's disease cohort for outlier detection yielding results consistent with the findings in the field. The code is available at `https://github.com/vsingh-group`.

6 ACKNOWLEDGMENTS

Research supported in part by NIH (R01AG040396, R01AG021155, R01AG027161, P50AG033514, R01AG059312, R01EB022883, R01AG062336), the Center for Predictive and Computational Phenotyping (U54AI117924), NSF CAREER Award (1252725), and a predoctoral fellowship to RRM via T32LM012413.

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML*, 2016.

Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, et al. Analyzing Inverse Problems with Invertible Neural Networks. In *ICLR*, 2019.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Network. In *International Conference on Machine Learning*, pages 1613–1622, 2015.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

Terence C Chua, Wei Wen, Melissa J Slavin, and Perminder S Sachdev. Diffusion tensor imaging in mild cognitive impairment and Alzheimer's disease: a review. *Current opinion in neurology*, 21(1):83–92, 2008.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Francesco Cricri, Mikko Honkala, Xingyang Ni, Emre Aksu, and Moncef Gabbouj. Video ladder networks. *arXiv preprint arXiv:1612.01756*, 2016.

Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NIPS*, 2016.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2016.

Cristóbal Esteban, Oliver Staeck, Stephan Baier, Yinchong Yang, and Volker Tresp. Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *ICHI*, 2016.

Meire Fortunato, Charles Blundell, and Oriol Vinyals. Bayesian Recurrent Neural Networks. *arXiv preprint arXiv:1704.02798*, 2017.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pages 2199–2207, 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. 1999.

Alex Graves. Practical variational inference for neural networks. In *NIPS*, 2011.

Karol Gregor and Frederic Besse. Temporal difference variational auto-encoder. *arXiv preprint arXiv:1806.03107*, 2018.

Michael D Greicius, Kaustubh Supekar, Vinod Menon, and Robert F Dougherty. Resting-state functional connectivity reflects structural connectivity in the default mode network. *Cerebral cortex*, 19(1):72–78, 2009.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *ICML*, 2015.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Computational learning theory*, 1993.

Seong Jae Hwang, Nagesh Adluru, Won Hwa Kim, Sterling C Johnson, Barbara B Bendlin, and Vikas Singh. Associations Between Positron Emission Tomography Amyloid Pathology and Diffusion Tensor Imaging Brain Connectivity in Pre-Clinical Alzheimer's Disease. *Brain connectivity*, 9: 162–173, 2019.

Abhyuday N Jagannatha and Hong Yu. Bidirectional rnn for medical event detection in electronic health records. In *Association for Computational Linguistics*, 2016.

Sterling C Johnson, Bradley T Christian, Ozioma C Okonkwo, Jennifer M Oh, Sandra Harding, Guofan Xu, Ansel T Hillmer, Dustin W Wooten, Dhanabalan Murali, Todd E Barnhart, Lance T Hall, Annie M Racine, William E Klunk, Chester A Mathis, Howard A Rowley, Bruce P Hermann, N. Maritza Dowling, Sanjay Asthana, and Mark A Sager. Amyloid burden and neural function in people at risk for Alzheimer's disease. *Neurobiology of aging*, 35(3):576–584, 2014.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep learning for Computer Vision? In *NIPS*, 2017.

Won Hwa Kim, Nagesh Adluru, Moo K Chung, Ozioma C Okonkwo, Sterling C Johnson, Barbara B Bendlin, and Vikas Singh. Multi-resolution statistical analysis of brain connectivity graphs in preclinical alzheimer's disease. *NeuroImage*, 118:103–117, 2015.

Won Hwa Kim, Annie M Racine, Nagesh Adluru, Seong Jae Hwang, Kaj Blennow, Henrik Zetterberg, Cynthia M Carlsson, Sanjay Asthana, Rebecca L Koscik, Sterling C Johnson, et al. Cerebrospinal fluid biomarkers of neurofibrillary tangles and synaptic dysfunction are associated with longitudinal decline in white matter connectivity: A multi-resolution graph analysis. *NeuroImage: Clinical*, 21, 2019.

Rahul G Krishnan, Uri Shalit, and David Sontag. Structured Inference Networks for Nonlinear State Space Models. In *AAAI*, pages 2101–2109, 2017.

Viveka Kulharia, Arnab Ghosh, Amitabha Mukerjee, Vinay Namboodiri, and Mohit Bansal. Contextual RNN-GANs for abstract reasoning diagram generation. In *AAAI*, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *NIPS*, 2017.

Jonathan Lambert, Peter B Greer, Fred Menk, Jackie Patterson, Joel Parker, Kara Dahl, Sanjiv Gupta, Anne Capp, Chris Wratten, Colin Tang, et al. MRI-guided prostate radiation therapy planning: Investigation of dosimetric accuracy of MRI-based dose planning. *Radiotherapy and Oncology*, 98 (3):330–334, 2011.

David JC MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992a.

David JC MacKay. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992b.

Andre F Marquand, Iead Rezek, Jan Buitelaar, and Christian F Beckmann. Understanding heterogeneity in clinical cohorts using normative models: beyond case-control studies. *Biological psychiatry*, 80(7):552–561, 2016.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. Deep exponential families. In *Artificial Intelligence and Statistics*, pages 762–771, 2015.

Carl Edward Rasmussen and Joaquin Quinonero-Candela. Healing the relevance vector machine through augmentation. In *ICML*, 2005.

Fabio De Sousa Ribeiro, Francesco Caliva, Mark Swainson, Kjartan Gudmundsson, Georgios Leontidis, and Stefanos Kollias. Deep Bayesian Uncertainty Estimation for Adaptation and Self-Annotation of Food Packaging Images. *arXiv preprint arXiv:1812.01681*, 2018.

Samuel J Rosenberg, Joseph J Ryan, and Aurelio Prifitera. Rey auditory-verbal learning test performance of patients with and without memory impairment. *Journal of clinical psychology*, 40(3):785–787, 1984.

Ruggiero Santeramo, Samuel Withey, and Giovanni Montana. Longitudinal detection of radiological abnormalities with time-modulated lstm. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 326–333. 2018.

Andreas Sedlmeier, Thomas Gabor, Thomy Phan, Lenz Belzner, and Claudia Linnhoff-Popien. Uncertainty-based out-of-distribution detection in deep reinforcement learning. *arXiv preprint arXiv:1901.02219*, 2019.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15 (1):1929–1958, 2014.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

Anna Varentsova, Shengwei Zhang, and Konstantinos Arfanakis. Development of a high angular resolution diffusion imaging human brain template. *NeuroImage*, 91:177–186, 2014.

Hao Wang, SHI Xingjian, and Dit-Yan Yeung. Natural-parameter networks: A class of probabilistic neural networks. In *NIPS*, 2016.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*, 2011.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Gabriel Ziegler, Gerard R Ridgway, Robert Dahnke, Christian Gaser, and Alzheimer's Disease Neuroimaging Initiative. Individualized Gaussian process-based prediction and detection of local and global gray matter abnormalities in elderly subjects. *NeuroImage*, 97:333–348, 2014.