Latent Time Neural Ordinary Differential Equations

Srinivas Anumasa, P.K. Srijith

Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
cs16resch11004@iith.ac.in,srijith@cse.iith.ac.in

Abstract

Neural ordinary differential equations (NODE) have been proposed as a continuous depth generalization to popular deep learning models such as Residual networks (ResNets). They provide parameter efficiency and automate the model selection process in deep learning models to some extent. However, they lack the much-required uncertainty modelling and robustness capabilities which are crucial for their use in several real-world applications such as autonomous driving and healthcare. We propose a novel and unique approach to model uncertainty in NODE by considering a distribution over the end-time T of the ODE solver. The proposed approach, latent time NODE (LT-NODE), treats T as a latent variable and apply Bayesian learning to obtain a posterior distribution over T from the data. In particular, we use variational inference to learn an approximate posterior and the model parameters. Prediction is done by considering the NODE representations from different samples of the posterior and can be done efficiently using a single forward pass. As T implicitly defines the depth of a NODE, posterior distribution over T would also help in model selection in NODE. We also propose, adaptive latent time NODE (ALT-NODE), which allow each data point to have a distinct posterior distribution over end-times. ALT-NODE uses amortized variational inference to learn an approximate posterior using inference networks. We demonstrate the effectiveness of the proposed approaches in modelling uncertainty and robustness through experiments on synthetic and several real-world image classification data.

Introduction

Deep learning models such as Residual networks (ResNet) (He et al. 2016) have brought advances in several computer vision tasks (Ren et al. 2017; He et al. 2020; Wang, Chen, and Hu 2019). They used skip connections to allow the models to grow deeper and improve performance without suffering from the vanishing gradient problem. Recently, neural ordinary differential equations (NODEs) (Chen et al. 2018) were proposed as a continuous depth generalization to ResNets. The feature computations in ResNet can be seen as solving an ordinary differential equation (ODE) with Euler method (Lu et al. 2018; Haber and Ruthotto 2017; Ruthotto and Haber 2019). Here, the ODE is parameterized by a neural network and the NODE can grow to an arbitrary depth as defined by the end-time T. It was shown that NODE is more robust (Hanshu et al. 2019) than traditional deep learning models, and is invertible, parameter efficient and maintains a constant memory cost with respect to growth in depth.

Modeling uncertainty is paramount for many high-risk applications such healthcare (Ker et al. 2017) and autonomous driving vehicles (Fridman et al. 2019). However, standard NODE models compute a point estimate of predictions which fail to capture uncertainty in predictions. Like ResNets, they tend to make high confidence wrong predictions on out-of-sample observations (Anumasa and Srijith 2021), restricting their use in high-risk applications. There exist very few works trying to address the uncertainty in NODE (Anumasa and Srijith 2021; Kong, Sun, and Zhang 2020; Dandekar et al. 2021) and their uncertainty modelling capabilities are restricted by architectural and training assumptions. Though, NODE models have addressed the model selection in deep learning to a great extent, it still require the user to define the parameters such as end time T to the ODE solver. This implicitly determines the depth of the NODE. In this work, we propose a unique approach to model uncertainty in NODE, latent time neural ODE (LT-NODE), which addresses these drawbacks by learning a distribution over end-time T.

LT-NODE is based on the idea of capturing uncertainty by treating the end time T as a latent variable. This allows us to define a distribution over T and the representations of the data point at different values of T sampled from the distribution provides an estimate of uncertainty. To capture uncertainty and to obtain a good generalization capability, it is important to learn the distribution over T from the data and we employ Bayesian inference techniques such as variational inference to learn an approximate posterior. Consequently, the posterior over T will also help in addressing the model selection in NODE in determining an appropriate end time. The proposed approach can get uncertainty estimates using single forward pass and is very efficient unlike other uncertainty modelling techniques which require multiple model evaluations. Moreover, it provides uncertainty estimates with hardly any increase in the number of parameters, as it only needs to estimate two additional parameters associated with the variational posterior.

Recently, it was shown that a NODE with a different depth (end-time) for different data points can overcome the draw-

backs of standard NODEs, for e.g., in solving *concentric annuli* and *reflection* tasks (Dupont, Doucet, and Teh 2019; Massaroli et al. 2020). Inspired by this, we propose a variant, adaptive latent time neural ODE (ALT-NODE), which allows each sample to have a separate posterior distribution over end-time. To learn the posterior, we consider an amortized variational inference, where we specify an inference network which provides the variational approximation over T for each sample. We develop ALT-NODEs which also do predictions efficiently, by requiring only one forward pass through the model. Moreover, the proposed uncertainty estimation techniques for neural ODEs are generic and can be applied to several recent variants of the NODE model and architectures. We demonstrate the superior uncertainty modelling capability of LT-NODE and ALT-NODE under different experimental setups on synthetic and several real-world image classification data sets such as CIFAR10, SVHN, MNIST, and F-MNIST. Our main contributions can be summarized as follows.

1. We propose a novel and unique approach to model uncertainty in NODE by treating end-time T as latent and learns a posterior distribution over end-times which also aids in model selection.
2. We propose a variant which learns input dependent posterior distribution over latent end-times.
3. We develop variational inference and amortized variational inference techniques for the proposed model to learn an approximate posterior distribution over latent end-times.
4. We demonstrate the uncertainty and robustness modelling capability of the proposed models on different experimental setups and on several image classification data sets.

Related work

Neural ODEs (Chen et al. 2018) are continuous depth generalization of ResNets (He et al. 2016) and was shown to provide competitive results on several image classification tasks. Recently, several NODE variants were proposed which improved the generalization performance in NODE. For instance, Zhuang et al. (2020) addressed the flaws in the adjoint sensitive method used to learn parameters in NODE to improve gradient computation and performance. Augmented NODE (ANODE) (Gholami, Keutzer, and Biros 2019) augmented the latent layers with additional dimension and was found to be more effective than NODE in solving complex problems such as *concentric annuli* and *reflection*. (Massaroli et al. 2020) addressed it by assuming depth of the NODE to be adaptive and data dependent. They also provide NODE variants which generalizes ANODE to consider data dependent and higher order augmentation. There are NODE variants which improves performance by letting the parameters to change over time (Massaroli et al. 2020; Zhang et al. 2019) or through regularization (Finlay et al. 2020; Ghosh et al. 2020). However, very few works aim to address the lack of uncertainty modelling and robustness capabilities in the neural ODE models. Although NODE (Hanshu et al. 2019) was shown to be more robust than similar ResNet architecture, they lack the required robustness and uncertainty modelling capabilities (Anumasa and Srijith 2021).

NODE-GP replaced the fully connected neural network layer in NODE with Gaussian processes to improve uncertainty and robustness capabilities in NODE (Anumasa and Srijith 2021). SDE-Net (Kong, Sun, and Zhang 2020) tries to address this by using the framework of stochastic differential equations. SDE-Net uses an additional diffusion network which learns to provide a high diffusion for the computed state trajectories of the data outside the training distribution. However, SDE-Net suffers some drawbacks in that it requires an additional diffusion network which needs to be trained explicitly on an out-of-distribution (OOD) data and require multiple forward passes through the model to get uncertainty estimates. Explicit training on an OOD data is practically infeasible for several applications. Concurrent to our work, Bayesian neural ODE (Dandekar et al. 2021) proposes to model uncertainty using the standard technique of learning a distribution over weights through the black-box inference techniques based on Markov chain Monte Carlo (MCMC) methods. We propose an uncertainty modelling technique unique to NODE and yet generalizable to several NODE architectures, where the uncertainty is modeled by considering a distribution over end-times. The proposed approach requires only a single forward pass through the model to obtain uncertainty aware predictive probability and models the uncertainty with just 2 additional parameters (variational parameters). This fully Bayesian approach which computes posterior over end-time is different from Ghosh et al. (2020) which uses random end-times only as a regularization technique during training and does not model uncertainty over predictions. Our approach to model uncertainty corroborates well with some recent advances in modelling uncertainty in discrete depth networks (Dikov and Bayer 2019; Antoran, Allingham, and Hernández-Lobato 2020; Wenzel et al. 2020). They show that uncertainty can be modelled effectively by considering representations from different layers or through distributions over hyper-parameters or architectures of a deep neural network. The proposed approach differs from them as NODE requires different probabilistic modelling, learning objective and training due to its continuous depth character. In addition, the use of amortized variational inference to learn input specific posterior distribution over end-times, further makes the proposed approach unique and novel.

The proposed approaches are different from the latent NODEs (Rubanova, Chen, and Duvenaud 2019; Yildiz, Heinonen, and Lahdesmaki 2019) which are generative NODEs used for modeling the latent state dynamics associated with time series data. They assume the initial state to be latent and a posterior distribution learnt over the initial state is used to generate the time series data. In contrast, we address the regression and classification problems with a fixed initial state (input image or a transformation of it). Hence, we consider a distribution over latent end-times and associated representations in a feed forward NODE to model uncertainty. However, this may not be a suitable for time series data where measurement times are observed.

Background

We consider a supervised learning problem and let $\mathcal{D} = \{X, \mathbf{y}\} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be the set of training data points with input $\mathbf{x}_i \in \mathcal{R}^D$ and $y_i \in \{1, \ldots, C\}$ for a classification and $y_i \in \mathcal{R}$ for regression. For a discrete deep learning model, the hidden representation at layer l is denoted as $\mathbf{h}_l$. We consider the hidden layer representations obtained through neural ODE transformations for a point $\mathbf{x}$ at time t as $\mathbf{h}_\mathbf{x}(t)$, where $\mathbf{h}_\mathbf{x}(t) \in \mathcal{R}^H$. The neural network transformations defining the ODE (NODE block) is denoted as $f(\mathbf{h}_\mathbf{x}(t), t, \boldsymbol{\theta}_h)$, with $\boldsymbol{\theta}_h$ being the neural network parameters. Typically, a NODE block is a stack of convolution layers or fully connected layers with nonlinear activation functions. The fully connected neural network (FCNN) transforming the hidden representation to a probability over the output y is represented as $g_y(\mathbf{h}_\mathbf{x}(t), \boldsymbol{\theta}_g)$, with parameters $\boldsymbol{\theta}_g$. We denote $\boldsymbol{\theta}$ to represent all the parameters in the NODE including the initial down-sampling block.

Neural Ordinary Differential Equations

ResNets transform the input to an output using a sequence of neural network transformations $f(\cdot)$ with skip connections between layers. The operations on a hidden representation $\mathbf{h}_t$ to obtain $\mathbf{h}_{t+1}$ in ResNets can be expressed as $\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \boldsymbol{\theta}_h)$. Neural ordinary differential equations (NODEs) show that a sequence of such transformations can be obtained as a solution to an ordinary differential equation of the following form, $\frac{d\mathbf{h}_\mathbf{x}(t)}{dt} = f(\mathbf{h}_\mathbf{x}(t), t, \boldsymbol{\theta}_h)$ Here, we assume the latent representations $\mathbf{h}_\mathbf{x}(t)$ is a function of time and changes continuously over time as defined by this ordinary differential equation. Solving the ODE requires one to provide an initial value $\mathbf{h}_\mathbf{x}(0)$ (initial value problem) and is typically considered as the input data $\mathbf{x}$ or a transformation using a down-sampling block $d(\cdot)$. Given $\mathbf{h}_\mathbf{x}(0)$, hidden representation at some end-time T can be obtained as $\mathbf{h}_\mathbf{x}(T) = \mathbf{h}_\mathbf{x}(0) + \int_0^T f(\mathbf{h}_\mathbf{x}(t), t, \boldsymbol{\theta}_h)$. Since the direct computation of $\mathbf{h}_\mathbf{x}(T)$ is intractable, numerical techniques such as Euler method or adaptive numerical techniques such as Dopri5 are used to obtain the final representation (ODESolve$(f(\mathbf{h}_\mathbf{x}(t), t, \boldsymbol{\theta}_h), \mathbf{h}_\mathbf{x}(0), 0, T)$). For e.g., Euler method is a single step method where $\mathbf{h}_\mathbf{x}(t)$ is updated sequentially until end-time T with a step size dt. A particular step in the Euler method can be written as $\mathbf{h}_\mathbf{x}(t+1) = \mathbf{h}_\mathbf{x}(t) + dt f(\mathbf{h}_\mathbf{x}(t), t, \boldsymbol{\theta}_h)$ We see that this is equivalent to the transformations performed in ResNet. On the other hand, adaptive numerical methods compute hidden representations at arbitrary times as determined by the error tolerance until the user specified end-time T. The end-time T implicitly determines the number of transformations and consequently the depth of the network. The hidden representation $\mathbf{h}_\mathbf{x}(T)$ is taken as the final layer representation and is passed through a fully connected neural network to obtain the probability of predicting the output y, i.e. $p(y|\mathbf{x}, T, \boldsymbol{\theta}) = g_y(\mathbf{h}_\mathbf{x}(T), \boldsymbol{\theta}_g)$. This predictive probability is then used with an appropriate loss function, for e.g., cross-entropy loss for classification, to obtain the final objective function. This is optimized to learn the parameters in the model using techniques such as adjoint sensitive method. NODE based models provide a generalization performance close to ResNets with much smaller number of parameters and automates the model selection (depth selection) to some extent.

Latent Time Neural Ordinary Differential Equations

NODEs were found to be useful for many computer vision applications. However, their application to high-risk real-world problems such as healthcare and autonomous driving is limited by their lack of uncertainty modelling capability. We aim to develop efficient NODE models which can provide good uncertainty estimates and make them amenable to such problems. We propose a novel approach, latent time neural ODE (LT-NODE), which is based on the idea of modelling uncertainty through the uncertainty over end-time T. The proposed approach considers the hidden representations at different end-times to obtain the predictive probability capable of modelling the model uncertainty or epistemic uncertainty. All the representations from different end-times T do not equally contribute to the predictive performance. Some of them will have a higher contribution than others. To account for this, we treat the end-time T as a latent variable and learn a distribution over it from the data. To achieve this, we follow Bayesian learning principles (Bishop 2006) where we define a prior distribution over T and learn a posterior distribution T from the data. The prediction is done using the representations corresponding to the end-time sampled from the posterior over T. The disagreements in the representations help to compute the model uncertainty. A side benefit of the proposed LT-NODE approach is that it automates the model selection over end time T. The posterior distribution over T allows the model to learn the end-time from the data. Moreover, our approach is generic and can be applied to model uncertainty with any recent NODE architecture.

The end-time T associated with NODE takes a positive real value and we would like to evaluate the representations at arbitrary times in the positive real valued interval to compute uncertainty. This makes NODEs challenging and different from discrete depth neural networks and we need an appropriate distribution which allows this. This motivates us to use Gamma distribution whose support is $(0, \infty)$ as the prior over T with shape and rate parameters being α_p and β_p respectively, thus $p(T|\alpha_p, \beta_p) = \text{Gamma}(T|\alpha_p, \beta_p) = \frac{\beta_p^{\alpha_p}}{\Gamma(\alpha_p)} T^{\alpha_p - 1} e^{-\beta_p T}$, where $\Gamma(\alpha_p)$ is gamma function. Gamma distribution can be useful to model the end-times as it is more flexible than exponential distribution and can place its probability density over end-times in any arbitrary region. The likelihood of modelling outputs $\mathbf{y}$ given a value for the end-time T and inputs X is denoted as $p(\mathbf{y}|T, X, \boldsymbol{\theta}) = \prod_{i=1}^N p(y_i|T, \mathbf{x}_i, \boldsymbol{\theta})$ where $p(y_i|T, \mathbf{x}_i, \boldsymbol{\theta}) = g_{y_i}(\mathbf{h}_{\mathbf{x}_i}(T), \boldsymbol{\theta}_g)$. Given the likelihood and the prior, the posterior over the latent variable T can be computed using the Bayes theorem (Bishop 2006) as

$$p(T|\mathbf{y}, X; \boldsymbol{\theta}) = \frac{p(\mathbf{y}|T, X; \boldsymbol{\theta}) p(T|\alpha_p, \beta_p)}{\int_0^\infty p(\mathbf{y}|T, X; \boldsymbol{\theta}) p(T|\alpha_p, \beta_p) dT} \quad (1)$$

However, the posterior cannot be computed in a closed form as the end-time T appears as a complex non-linear function in the likelihood. Consequently, the marginal likelihood term in the denominator of (1) also cannot be computed. Hence, we resort to approximate inference techniques such as the variational inference (Blei, Kucukelbir, and McAuliffe 2017) to obtain an approximate posterior over the T.

Variational Inference

In our approach, we choose Gamma distribution as the variational posterior over T (due to positive real valued T) with variational parameters α_q and β_q, thus $q(T|\alpha_q, \beta_q) = \text{Gamma}(T|\alpha_q, \beta_q)$. We derive the variational lower bound or evidence lower bound (ELBO) for our setting as follows.

$$\log(p(\mathbf{y}|X;\boldsymbol{\theta}) \geq \mathbb{E}_{q(T|\alpha_q,\beta_q)}[\log(p(\mathbf{y}|T,X;\boldsymbol{\theta}))] - \mathbb{KL}((q(T|\alpha_q,\beta_q)||p(T|\alpha_p,\beta_p))). \quad (2)$$

We learn the variational posterior parameters by maximising the lower bound. The $\mathbb{KL}$ term in (2) can be computed in closed form (Bauckhage 2014) as

$$\alpha_q \log \beta_q - \alpha_p \log \beta_p + log(\Gamma(\alpha_p)) - log(\Gamma(\alpha_q)) + (\psi(\alpha_q) - log\beta_q)(\alpha_q - \alpha_p) + \frac{\Gamma(\alpha_q+1)}{\Gamma(\alpha_q)}\frac{\beta_p}{\beta_q} - \alpha_q$$

where ψ is a digamma function. We approximate the computation of the expectation term in the ELBO by discretizing the space of T into a uniform grid and use S samples of T from the uniform grid to approximate the expectation as

$$\mathbb{E}_{q(T|\alpha_q,\beta_q)}[log(p(\mathbf{y}|T,X,\boldsymbol{\theta}))] = \sum_{i=1}^{N}\sum_{s=1}^{S} \log(p(y_i|T_s,\mathbf{x}_i,\boldsymbol{\theta}))q(T_s|\alpha_q,\beta_q), \quad (3)$$

where, $T_s \sim Uniform(T|a,b)$. We decided to use the uniform grid approximation rather than Monte Carlo approximation because of two reasons. Firstly, the latent variable T is a scalar quantity and consequently this approach will not suffer from the sampling inefficiency typically associated with high dimensional variables which motivate the use of Monte Carlo sampling techniques. Secondly, uniform grid approximation allows us to consider the variational distribution $q(T|\alpha_q, \beta_q)$ explicitly in the objective function and makes sampling independent of the parameters to be estimated. This will ease the estimation of variational parameters using gradient descent. We maximize ELBO and back-propagate the gradients to estimate both the variational and model parameters ($\boldsymbol{\theta}$).

Let $\bar{S} = \{T_1, T_2, \ldots, T_S\}$ be the S end-times sampled from the uniform distribution. During forward propagation, intermediate feature vectors are computed using an adaptive numerical technique such as Dopri5 (Kimura 2009)) until T_S. The feature vectors at $T_i \in \bar{S}$ obtained using the adaptve numerical technique and interpolation are used to compute the approximate log probability of training samples (3) and consequently in (2) to obtain ELBO. We note that this can be computed efficiently by ordering the sampled times (in ascending order) and obtaining the features vectors at these times in a single forward pass.

Algorithm 1: Forward pass in LT-NODE, computing predictive probability for datapoint $\mathbf{x}$.

%Sample S end-times from the variational posterior $q(T|\alpha_q, \beta_q)$, Initialize $\bar{S} = \{\}$
while $|\bar{S}| \leq S$ **do**
  Sample $T_s \sim q(T|\alpha_q, \beta_q)$
  $\bar{S} = \bar{S} \cup T_s$
Sort $\bar{S}$ in increasing order
Transform input using the downsampling: $\mathbf{h_x}(0) = d(\mathbf{x})$
initialize : $t = 0$, $\mathbf{prob\_vec} = \mathbf{0}$
for T_s in $\bar{S}$
  $\mathbf{h_x}(T_s) = \text{ODESolve}(f, \mathbf{h_x}(t), t, T_s)$
  $t = T_s$
  for $y = 1, \ldots, C$
    $\mathbf{sample\_prob\_vec}(y) = g_y(\mathbf{h_x}(t), \boldsymbol{\theta}_g)$
  $\mathbf{prob\_vec} = \mathbf{prob\_vec} + \mathbf{sample\_prob\_vec}$
return $\mathbf{prob\_vec} = \frac{\mathbf{prob\_vec}}{S}$

The model parameters and variational parameters learnt by maximizing ELBO are used to predict the test data. First, we sample the end-times from the learnt variational posterior $q(T|\alpha_q, \beta_q)$. The sampled end-times are ordered, and predictions are done efficiently using a single forward pass through the model in a similar manner as discussed for training. We compute the predictive probability of a test data point $\mathbf{x}$ to be classified to a class y as $\frac{1}{S}\sum_{s=1}^{S} p(y|T_s, \mathbf{x}, \boldsymbol{\theta})$, where $T_s \sim q(T|\alpha_q, \beta_q)$. LT-NODE provides good uncertainty estimates with only 2 additional parameters (α_q and β_q). A schematic representation and a detailed algorithm of the proposed LT-NODE are shown in Figure 1 and Algorithm 1.

Adaptive Latent Time Neural Ordinary Differential Equations

LT-NODE computes a posterior distribution over end-time T which helps to model uncertainty as well as aid in model selection. However, end-time T is treated as a global latent variable and the distribution over T is assumed to be same for all the data points. Though this gives a good uncertainty estimate, NODE modelling capability can be improved by considering the end-time to be different across different data points. Massaroli et al. (2020) showed that a NODE with input specific depth will be able to model complex problems such as solving *concentric annuli* and *reflection* tasks. To improve the modelling capability, we propose a variant, adaptive latent time NODE (ALT-NODE), which allows each data point to have input specific distribution over end-time.

In ALT-NODE, we assume that every data point is associated with a latent variable T_i denoting the end-time associated with the data point. We assume the same Gamma prior over T_i with parameters α_p and β_p as before. The likelihood $p(y_i|T, \mathbf{x}_i, \boldsymbol{\theta})$ is also defined as in LT-NODE. Here, we will be learning a separate posterior distribution over T_i associated with each data point. As in LT-NODE, the posterior cannot be computed tractably, and we resort to variational inference to obtain an approximate posterior. For ALT-NODE, we associate a separate variational posterior $q(T_i)$ with each T_i. Due to the nature of T_i, we assume $q(T_i)$ to be Gamma

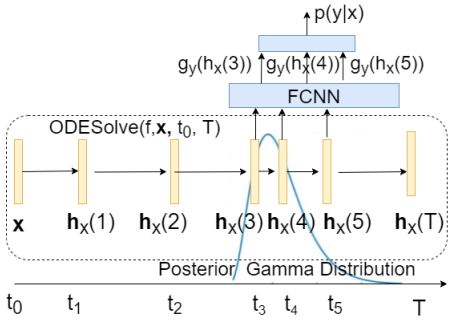


Figure 1: Representation of the LT-NODE. Posterior over end-times is Gamma distributed, and assume t_3, t_4, and t_5 are the end-times sampled from the Gamma. The representations at these times are passed through the FCNN and the output is averaged to get the final predictive probability.

distributed with parameters α_{qi} and β_{qi}. Treating the variational parameters as free form distribution can lead to a few drawbacks. Firstly, the number of variational parameters to learn increases linearly with number of data points and can become costly when number of data points is high. Secondly, it does not allow us to perform inference over new data points.

We use an amortized variational inference (Gershman and Goodman 2014; Kingma and Welling 2014) approach to address these drawbacks. The amortized VI assumes the variational parameters associated with the T_i can be obtained as a function of the input data points. It introduces an inference network, typically a parametric function such as neural networks, which can predict the variational parameters from the input data. Now, instead of learning the variational parameters, one can learn the parameters of the inference network from the variational lower bound. Learning of the inference network allows statistical strength to be shared across data points and helps in predicting the variational parameters for a new data point. Therefore, we introduce an inference network $r(\mathbf{x}_i;\phi)$ which predicts the variational parameters α_{qi} and β_{qi} associated with T_i. Consequently, we denote the variational distribution over T_i, to be parameterized by the inference network parameters ϕ and is conditioned on $\mathbf{x}_i$, i.e., $q(T_i|\mathbf{x}_i,\phi)$. We learn the inference network parameters ϕ and model parameters θ by maximizing the variational lower bound for ALT-NODE which is derived as follows

$$\sum_{i=1}^{N}[\mathbb{E}_{q(T_i|\mathbf{x}_i,\phi)}[\log(p(y_i|T_i,\mathbf{x}_i;\boldsymbol{\theta}))]- \mathbb{KL}((q(T_i|\mathbf{x}_i,\phi)||p(T_i|\alpha_p,\beta_p)))]. \quad (4)$$

We follow the approximation used in LT-NODE to evaluate the expectation term in the objective function (4). The KL divergence term can be obtained in closed form as before, but the variational parameters is a function of the inference network $r(\mathbf{x})$ parameterized by ϕ. We developed an efficient approach to perform single forward pass computation through the ALT-NODE similar to LT-NODE for training and prediction [1].

[1]Details of the approach in supplementary

Experiments

We conduct experiments to evaluate the uncertainty and robustness modelling capabilities of the proposed approaches, LT-NODE and ALT-NODE using synthetic and real-world data sets. The approaches are compared against standard NODE (Chen et al. 2018) and baselines which were recently proposed to model uncertainty in the NODE models, such as NODE-GP (Anumasa and Srijith 2021) and SDE-Net (Kong, Sun, and Zhang 2020). We also consider a baseline Uni-NODE which does not learn any posterior over T but only considers randomness over T by sampling it from a uniform distribution during training and testing. This baseline is a variant of (Ghosh et al. 2020) where the model considered a noisy end-time during training but not during testing.

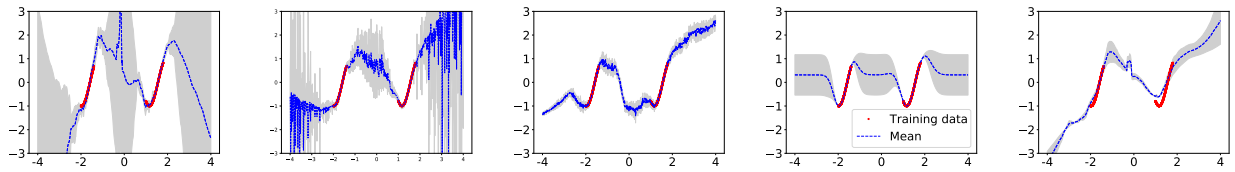
Synthetic Data Experiments

We consider a 1D synthetic regression dataset (Foong et al. 2019) to demonstrate the uncertainty modeling capability of the proposed models [2]. This dataset contains two disjoint clusters of training points. We expect the models to exhibit high variance in-between and away from these training data points. Figure 2 provide the predictive mean and standard deviation obtained with the proposed models and baselines. In this 1-D regression problem, LT-NODE model as shown in Figure 2(a) captures the uncertainty well with high variance on in-between and away data points on the left, and the variance grows smoothly. Infact, it is found to have highest in-between variance. ALT-NODE in Figure 2(b) also captures the uncertainty well, and due to the input conditioned posterior it is able to fit and learn the trends in data better than LT-NODE. We find that SDE-Net in Figure 2(c) exhibits some uncertainty but the variance does not increase as we move away from training data regime. NODE-GP in Figure 2(d) gives a good uncertainty modelling capability with a high variance on both in-between and away OOD data. But we show later that on high-dimensional image data, NODE-GP fails to model the uncertainty due to the inability of GPs to model high dimensional data. We also conducted experiments to demonstrate the importance of learning a posterior distribution over T. In Figure 2(e), we can observe that Uni-NODE which does a random sampling of T was not able to fit the data unlike other models but having a randomness over T provided some uncertainty modeling capability in the away data region.

Image classification

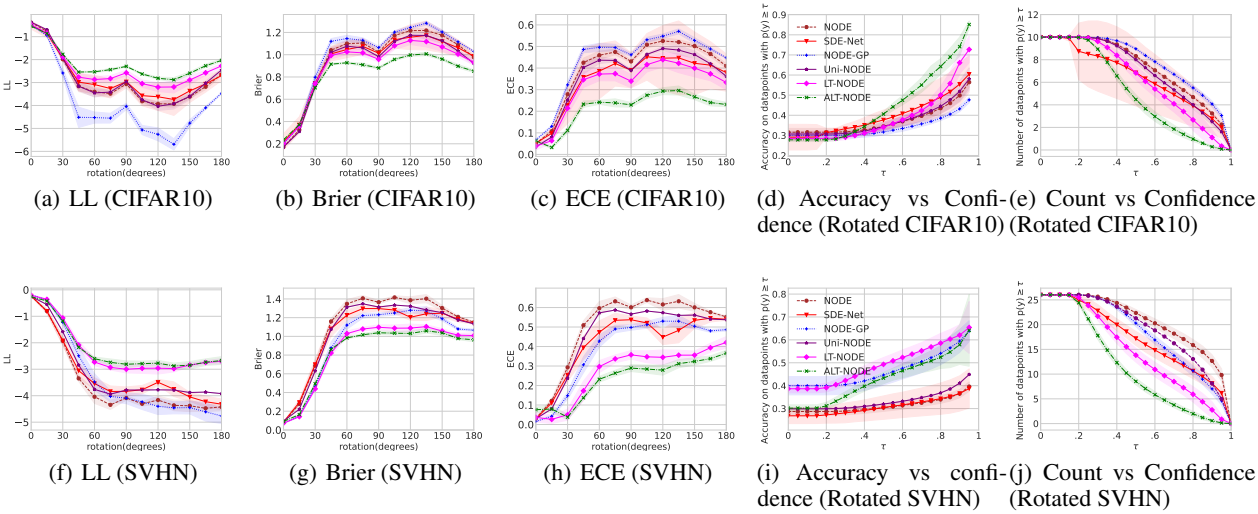
We conduct experiments to study uncertainty modelling and robustness capability of the proposed models on image classification problems. We consider popular data sets used in image classification such as CIFAR10 (Krizhevsky et al. 2009), SVHN (Netzer et al. 2011), MNIST (LeCun et al. 1998) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017). To measure their uncertainty modelling capability, we use several metrics such as Error, log-likelihood (LL), Bier score and

[2]We use Gamma$(2, 0.5)$ as prior and learn the approximate variational posterior by maximizing ELBO, for e.g., LT-NODE learnt Gamma$(1.27, 0.98)$ as the approximate posterior on the synthetic data, more details in the supplementary.



(a) LT-NODE (b) ALT-NODE (c) SDE-Net (d) NODE-GP (e) Uni-NODE

Figure 2: Results on 1-D synthetic regression data (Foong et al. 2019). Mean prediction is denoted by dotted blue line and shaded region represents mean $\pm$ std. deviation. We also provide average entropy (E) computed in the OOD interval $(-0.5, 0.5)$. (a) LT-NODE (E:2.42) exhibits a good uncertainty modelling capability followed by (d) NODE-GP (E:1.22) and (b) ALT-NODE (E:1.17). (c) SDE-Net (E:-0.31) exhibits some uncertainty, but it remains stagnant in the OOD regime. (e) Uni-NODE (E:-0.65) exhibits some uncertainty but does not fit the data well.



(a) LL (CIFAR10) (b) Brier (CIFAR10) (c) ECE (CIFAR10) (d) Accuracy vs Confidence (Rotated CIFAR10) (e) Count vs Confidence (Rotated CIFAR10)

(f) LL (SVHN) (g) Brier (SVHN) (h) ECE (SVHN) (i) Accuracy vs confidence (Rotated SVHN) (j) Count vs Confidence (Rotated SVHN)

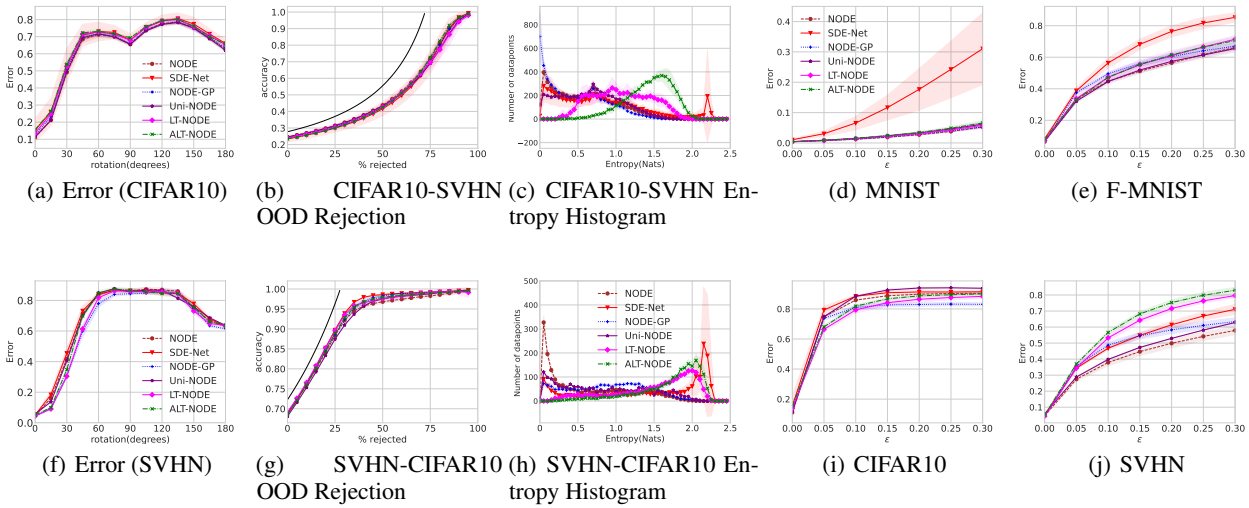
Figure 3: Performance under varying degrees of rotation in CIFAR10 (top) and SVHN (bottom). LT-NODE and ALT-NODE show better uncertainty modelling capability with higher LL values and lower Brier and ECE scores as we increase the rotation. Confidence distribution plots on rotated (45 degrees) images are shown in (d), (e), (i) and (j). (d) and (i) shows the accuracy, and (e) and (j) shows the count of predictions (y axis value multipled by 1000) done with confidence above a threshold τ on X-axis. The proposed models perform better than baselines giving higher accuracies and lower counts on high confident predictions.

expected calibration error (ECE). Error (1$-$ accuracy) and LL are the standard metrics used in image classification. LL consider the probability distribution over outputs and can measure the uncertainty modelling capability of the models (higher the better). Brier score (Blattenberger and Lad 1985) and ECE (Naeini, Cooper, and Hauskrecht 2015) are calibration metrics which tells us if the predictive probability of the model for a class label is close to the true proportion of those classes in the test data. Both the measures consider predictive probability and consequently can be used to measure uncertainty in predictions (lower values of Brier score and ECE are preferred). All the models follow the same architecture as standard NODE[3]. Additional networks are required for SDE-Net for diffusion and ALT-NODE for inference, both using 3 convolution layers followed by a fully connected layer [4]. The plots show mean and standard deviation obtained by training the models with 5-different initializations.

Performance under rotation We use CIFAR10 and SVHN data sets for studying the performance of the proposed models under rotation of images (Ovadia et al. 2019). The performance of the models degrades quickly with increase in amount of rotation applied on the image test data and can be seen in Figure 4(a) and (f). We want our models to be least overconfident when there is a significant shift in the data. To study this, in Figure 3 we plot the performance of the models in terms of LL, Brier score and ECE. LT-NODE and ALT-NODE have better Brier score, LL and ECE values compared to the baselines, demonstrating their improved uncertainty modelling capability. We consider confidence distribution for the methods when prediction is done on SVHN and CIFAR10 rotated by 45 degrees. We plot the accuracy (Figure 3 (d) and (i)) and count of test data points (Figure 3 (e) and (j)) when predictions are done with confidence above a threshold τ. In general, we want the counts of points predicted with high confidence to be lower and accuracy to be higher as we are dealing with a corrupted data. We can observe that this is the

[3]https://github.com/rtqichen/torchdiffeq

[4]Details of experiments and architecture is in the supplementary.



(a) Error (CIFAR10) (b) CIFAR10-SVHN OOD Rejection (c) CIFAR10-SVHN Entropy Histogram (d) MNIST (e) F-MNIST

(f) Error (SVHN) (g) SVHN-CIFAR10 OOD Rejection (h) SVHN-CIFAR10 Entropy Histogram (i) CIFAR10 (j) SVHN

Figure 4: Error of the models under varying degrees of rotation in CIFAR10 (a) and SVHN (f). OOD Rejection plot (b and g) and Entropy histogram (c and h) of the models for OOD experiments. For (b) and (g), training is done on CIFAR10 and testing on SVHN, while for (c) and (h) training is done on SVHN and testing on CIFAR10. Error obtained by the models under FGSM attack against varying ϵ (stepsize) on various data sets (d) MNIST (e) Fashion-MNIST (i) CIFAR10 and (j) SVHN.

case with the proposed approaches, beating baselines in both rotated CIFAR10 and SVHN. Accuracy of the proposed models are the highest with high confident predictions, making them more reliable. We have found that ALT-NODE performance is better than all the models. Learning input specific distribution helps in capturing uncertainty better. LT-NODE performed better than all except ALT-NODE, showing that learning a distribution over end-times in general improves uncertainty modeling capability.

Performance on Out of Distribution Data We conduct experiments to study the performance of the models on out-of-distribution (OOD) data by training them on either CIFAR10 or SVHN and testing them on the other. We expect a good model to exhibit a high uncertainty on the test data set which is measured using the entropy score. Entropy measures the spread of the predictive probability across the classes and expects a higher entropy (higher spread) on OOD data. We follow the OOD rejection setup (Filos et al. 2019), where we receive data from both in-distribution (ID) and OOD. An ideal model should be able to classify with high accuracy by ignoring the data points with high uncertainty. We study the classification accuracy of the models as we increase the proportion of rejected points in Figure 4(b) and (g). We can observe that the performances of all the models are close. So, we analyse the entropy histogram of the models for the OOD data in Figure 4(c) and (h). We can observe that for the case where models trained on CIFAR10 and tested on SVHN (Figure 4 (c)), the proposed models have a better entropy histogram. They have higher number of points with high entropy and vice-versa compared to the baselines, reflecting their superior uncertainty modelling capability on OOD data. In summary, the average entropy values on OOD data (SVHN) when trained on CIFAR10 are, NODE: 0.572 ± 0.062, SDE-Net: 0.792 ± 0.354, NODE-GP: 0.495 ± 0.0298, Uni-NODE: 0.668 ± 0.040, LT-NODE: 1.074 ± 0.078, ALT-NODE: 1.444 ± 0.050. Our proposed models having the higher entropy values, exhibiting higher uncertainty over OOD data. In Figure 4 (h) Although SDE-Net having larger number of points on the ends of the histogram spectrum, our proposed models have the better entropy values. NODE: 0.617 ± 0.021, SDE-Net: 1.421 ± 0.078, NODE-GP: 0.885 ± 0.034, Uni-NODE: 0.808 ± 0.111, LT-NODE: 1.537 ± 0.067, ALT-NODE: 1.723 ± 0.034.

Robustness Evaluation Deep learning models are prone to adversarial attacks (Goodfellow et al. 2016; Szegedy et al. 2013). To check robustness of models, we conduct experiments to evaluate their performance under FGSM (Goodfellow, Shlens, and Szegedy 2014) attack on MNIST, F-MNIST, CIFAR10 and SVHN. Figures 4(d),(e),(i), and (j) shows the robustness of the models in terms of error against increasing perturbation strength ϵ of FGSM attack. The proposed models LT-NODE and ALT-NODE performed well with low error on all the data sets except SVHN, demonstrating their robustness against adversarial attack.

Conclusion

We proposed a novel method to model uncertainty in NODE by learning a distribution over latent end-times. The proposed approaches can compute uncertainty efficiently in a single forward pass and helps in end-time selection in NODE. The proposed models, LT-NODE and ALT-NODE were shown to have good uncertainty modelling and robustness capabilities through experiments on synthetic and real-world data image classification data. We expect to further improve their performance by considering a multi-modal variational posterior distribution as a future work. The proposed NODE models could bring advances in computer vision applications like autonomous driving where uncertainty modeling is important.

Acknowledgements

We acknowledge the funding support from MoE, Govt of India and DST ICPS and computational support through JICA funds.

References

Antoran, J.; Allingham, J.; and Hernández-Lobato, J. M. 2020. Depth Uncertainty in Neural Networks. In *Advances in Neural Information Processing Systems*, volume 33, 10620–10634.

Anumasa, S.; and Srijith, P. 2021. Improving Robustness and Uncertainty Modelling in Neural Ordinary Differential Equations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4053–4061.

Bauckhage, C. 2014. Computing the kullback-leibler divergence between two generalized gamma distributions. *arXiv preprint arXiv:1401.6853*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag.

Blattenberger, G.; and Lad, F. 1985. Separating the Brier score into calibration and refinement components: A graphical exposition. *The American Statistician*, 39(1): 26–32.

Blei, D. M.; Kucukelbir, A.; and McAuliffe, J. D. 2017. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518): 859–877.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. In *Advances in neural information processing systems*, 6571–6583.

Dandekar, R.; Chung, K.; Dixit, V.; Tarek, M.; Garcia-Valadez, A.; Vemula, K. V.; and Rackauckas, C. 2021. Bayesian Neural Ordinary Differential Equations. *arXiv preprint arXiv:2012.07244*.

Dikov, G.; and Bayer, J. 2019. Bayesian learning of neural network architectures. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 730–738.

Dupont, E.; Doucet, A.; and Teh, Y. W. 2019. Augmented neural odes. In *Advances in Neural Information Processing Systems*, 3140–3150.

Filos, A.; Farquhar, S.; Gomez, A. N.; Rudner, T. G.; Kenton, Z.; Smith, L.; Alizadeh, M.; de Kroon, A.; and Gal, Y. 2019. A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks. *arXiv preprint arXiv:1912.10481*.

Finlay, C.; Jacobsen, J.-H.; Nurbekyan, L.; and Oberman, A. 2020. How to Train Your Neural ODE: the World of Jacobian and Kinetic Regularization. In *Proceedings of the 37th International Conference on Machine Learning*, 3154–3164.

Foong, A.; Li, Y.; Hernández-Lobato, J.; and Turner, R. 2019. In-Between Uncertainty in Bayesian Neural Networks. *arXiv preprint arXiv:1906.11537*.

Fridman, L.; Brown, D. E.; Glazer, M.; Angell, W.; Dodd, S.; Jenik, B.; Terwilliger, J.; Patsekin, A.; Kindelsberger, J.; Ding, L.; et al. 2019. MIT Advanced Vehicle Technology Study: Large-Scale Naturalistic Driving Study of Driver Behavior and Interaction with Automation. *IEEE Access*.

Gershman, S.; and Goodman, N. D. 2014. Amortized Inference in Probabilistic Reasoning. *Cognitive Science*, 36.

Gholami, A.; Keutzer, K.; and Biros, G. 2019. ANODE: unconditionally accurate memory-efficient gradients for neural ODEs. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 730–736. AAAI Press.

Ghosh, A.; Behl, H.; Dupont, E.; Torr, P.; and Namboodiri, V. 2020. STEER : Simple Temporal Regularization For Neural ODE. In *Advances in Neural Information Processing Systems*, volume 33, 14831–14843.

Goodfellow, I.; Bengio, Y.; Courville, A.; and Bengio, Y. 2016. *Deep learning*, volume 1. MIT Press.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Haber, E.; and Ruthotto, L. 2017. Stable architectures for deep neural networks. *Inverse Problems*, 34(1): 014004.

Hanshu, Y.; Jiawei, D.; Vincent, T.; and Jiashi, F. 2019. On Robustness of Neural Ordinary Differential Equations. In *International Conference on Learning Representations*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2020. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2): 386–397.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Ker, J.; Wang, L.; Rao, J.; and Lim, T. 2017. Deep learning applications in medical image analysis. *Ieee Access*, 6: 9375–9389.

Kimura, T. 2009. On dormand-prince method. *Japan Malaysia Technical Institute*, 40(10): 1–9.

Kingma, D. P.; and Welling, M. 2014. Auto-encoding Variational Bayes. In *2nd International Conference on Learning Representations (ICLR)*.

Kong, L.; Sun, J.; and Zhang, C. 2020. SDE-Net: Equipping Deep Neural Networks with Uncertainty Estimates. In *International Conference on Machine Learning*, 5405–5415. PMLR.

Krizhevsky, A.; et al. 2009. Learning multiple layers of features from tiny images.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lu, Y.; Zhong, A.; Li, Q.; and Dong, B. 2018. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, 3276–3285. PMLR.

Massaroli, S.; Poli, M.; Park, J.; Yamashita, A.; and Asama, H. 2020. Dissecting Neural ODEs. In *Advances in Neural Information Processing Systems*, 3952–3963.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.; Lakshminarayanan, B.; and Snoek, J. 2019. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6): 1137–1149.

Rubanova, Y.; Chen, R. T. Q.; and Duvenaud, D. 2019. Latent ODEs for Irregularly-Sampled Time Series.

Ruthotto, L.; and Haber, E. 2019. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 1–13.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Wang, W.; Chen, Z.; and Hu, H. 2019. Hierarchical attention network for image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8957–8964.

Wenzel, F.; Snoek, J.; Tran, D.; and Jenatton, R. 2020. Hyperparameter Ensembles for Robustness and Uncertainty Quantification. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 6514–6527.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.

Yildiz, C.; Heinonen, M.; and Lahdesmaki, H. 2019. ODE2VAE: Deep generative second order ODEs with Bayesian neural networks. In *Advances in Neural Information Processing Systems*, volume 32.

Zhang, T.; Yao, Z.; Gholami, A.; Gonzalez, J. E.; Keutzer, K.; Mahoney, M. W.; and Biros, G. 2019. ANODEV2: A Coupled Neural ODE Framework. In *Advances in Neural Information Processing Systems*, volume 32.

Zhuang, J.; Dvornek, N.; Li, X.; Tatikonda, S.; Papademetris, X.; and Duncan, J. 2020. Adaptive Checkpoint Adjoint Method for Gradient Estimation in Neural ODE. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 11639–11649. PMLR.