Robust Biophysical Parameter Estimation with a Neural Network Enhanced Hamiltonian Markov Chain Monte Carlo Sampler

Thomas Yu ✉[1], Marco Pizzolato ✉[1], Gabriel Girard[2], Jonathan Rafael-Patino[1], Erick Jorge Canales-Rodríguez[1,2,3,4], and Jean-Philippe Thiran[1,2,5]

[1] Signal Processing Lab (LTS5), École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
thomas.yu@epfl.ch, marco.pizzolato@epfl.ch
[2] Department of Radiology, Centre Hospitalier Universitaire Vaudois (CHUV), Lausanne, Switzerland
[3] FIDMAG Germanes Hospitalàries, Sant Boi de Llobregat, Barcelona, Spain
[4] Mental Health Research Networking Center (CIBERSAM), Madrid, Spain
[5] University of Lausanne, Lausanne, Switzerland

Abstract. Probabilistic parameter estimation in model fitting runs the gamut from maximum likelihood or maximum a posteriori point estimates from optimization to Markov Chain Monte Carlo (MCMC) sampling. The latter, while more computationally intensive, generally provides a better characterization of the underlying parameter distribution than that of point estimates. However, in order to efficiently explore distributions, MCMC methods ideally require generating uncorrelated samples while also preserving reasonable acceptance probabilities; this becomes particularly important in problematic regions of parameter space. In this paper, we extend a recently proposed Hamiltonian MCMC sampler parametrized by neural networks (L2HMC) by modifying the loss function to jointly optimize the distance between samples and the acceptance probability such that it is stable and efficient. We apply this enhanced sampler to parameter estimation in a recently proposed MRI model, the multi-echo spherical mean technique. We show that it generally outperforms the state of the art Hamiltonian No-U-Turn (NUTS) sampler, L2HMC, and a least squares fitting in terms of accuracy and precision, also enabling the generation of more informative parameter posterior distributions. This illustrates the potential of machine learning enhanced samplers for improving probabilistic parameter estimation for medical imaging applications.

Keywords: Markov Chain Monte Carlo · Hamiltonian MCMC Sampler · Magnetic Resonance Imaging · Optimization · Parameter Estimation.

1 Introduction

Given a data vector $\mathbf{s} \in \mathbb{R}^d$ generated from varying an independent experimental variable $v \in \mathbb{R}$, and a model $M(\mathbf{x}, v) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ with parameters $\mathbf{x} \in \mathbb{R}^n$ to

explain the data, probabilistic parameter estimation constructs a probabilistic model for the data and views the problem as inferring the parameters of a probability distribution [17]. For example, one can define the likelihood of the data given fixed parameters by treating each data point as an independent sample from a Gaussian distribution with mean at the model evaluated at the corresponding $v, \mathbf{x}$ and with a variance coming from the measurement noise. Using Bayes' theorem, the posterior probability distribution of the parameters given the data is proportional to the product of the likelihood function and the priors on the parameters:

$$p(\mathbf{x}|\mathbf{s}) \propto p(\mathbf{s}|\mathbf{x})p(\mathbf{x}), \tag{1}$$

where,

$$p(\mathbf{s}|\mathbf{x}) = \mathcal{N}(M(\mathbf{x}, \mathbf{v}), \sigma^2) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{(\mathbf{s}_i - M(\mathbf{x}, \mathbf{v}_i))^2}{2\sigma^2}), \tag{2}$$

and σ^2 denotes the noise variance. The prior distribution $p(\mathbf{x})$ encodes constraints such as sum constraints or upper/lower bounds through, for example, Dirichlet and uniform distributions [1]. An immediate candidate for a parameter estimate is then the maximum a posteriori (MAP) point estimate

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{s}). \tag{3}$$

This reduces the parameter estimation to an optimization problem. However, there are two potential problems with this point estimate. First, the general problems of uniqueness and feasibility of optimization, i.e. finding the MAP estimate. Second, the underlying assumption of this point estimate is that the mode is a good representation of the underlying probability distribution. Intuitively, we can see the truth of this assumption for many commonly used distributions in three or less dimensions e.g. normal or exponential. However, this assumption can fail as the dimensionality and complexity of the distribution increases due to the geometry of high dimensional spaces [3] and the concentration of measure phenomenon [3, 14]. Hence, inferring the parameters of a probabilistic model of high dimension and/or complexity through a mode point estimate can lead to spurious results. One approach to handle these issues is to first characterize the posterior distribution with Markov Chain Monte Carlo (MCMC) techniques [9] by sampling from the posterior distribution. One can then, as an example, use the mode, mean, median, etc. of the marginal posterior distributions of the parameters for the parameter estimate. In this paper, we use the expectation of each parameter over its marginal posterior, approximated by

$$\mathbf{x}^* \approx \frac{1}{N} \sum_{i}^{N} \mathbf{x_i}, \tag{4}$$

where the subscript i denotes one of the N samples.

The contributions of this paper are, first, to extend a Hamiltonian MCMC sampler parametrized with neural networks proposed in [15] . We modify the

loss function to balance acceptance probability and mixing such that both fast mixing and stable exploration of problematic regions of state space are possible. Second, we apply our extended sampler to estimate the parameters of a recently proposed MRI model and compare it to a least squares fitting and application of two state of the art Hamiltonian samplers.

2 Related Work

2.1 Hamiltonian Markov Chain Monte Carlo

In the following, we denote the posterior distribution from which we want to sample as $p(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^n$ being the state variables. MCMC methods sample from the posterior by generating a sequence of samples where each new sample $\mathbf{x}_t$ is generated from the previous sample $\mathbf{x}_{t-1}$ according to a transition distribution $T(\mathbf{x}_t|\mathbf{x}_{t-1})$ [4]. In order for the posterior to be the unique distribution to which this sequence converges, the transition distribution must satisfy ergodicity, which can usually be safely assumed, and an invariance property which is usually shown by proving a property called detailed balance $p(\mathbf{x}_t)T(\mathbf{x}_{t-1}|\mathbf{x}_t) = p(\mathbf{x}_{t-1})T(\mathbf{x}_t|\mathbf{x}_{t-1})$.

One well known way to construct a transition satisfying detailed balance called the Metropolis-Hastings algorithm [9] is as follows: given a proposal distribution $q(\mathbf{x}'|\mathbf{x}_{t-1})$, sample a candidate $\mathbf{x}'$; then, accept $\mathbf{x}'$ with probability $A(\mathbf{x}'|\mathbf{x}_{t-1}) = min(1, \frac{p(\mathbf{x}')q(\mathbf{x}_{t-1}|\mathbf{x}')}{p(\mathbf{x}_{t-1})q(\mathbf{x}'|\mathbf{x}_{t-1})})$. If accepted, $\mathbf{x}_t = \mathbf{x}'$. If rejected, $\mathbf{x}_t = \mathbf{x}_{t-1}$. However, even if a sampler satisfies these properties, the convergence is only proven asymptotically [4]. The typical procedure is to first have a burn-in stage where the sampler is run for some amount of steps in order for it to converge. Then, the actual sampling begins, with the burn-in samples being discarded [4].

For efficient exploration, the samples should ideally be uncorrelated, which can be accomplished by large distances between samples in the sample space, i.e. mixing. Autocorrelation analysis using multiple chains of samples can be used as a rough measure of how many samples are necessary. We emphasize that a balance must be found between the acceptance probability and the mixing; acceptance probabilities which are very high can mean the samples are very close/correlated and large distances between samples can lead to only a small number of samples being accepted. One powerful MCMC method which scales with the dimensionality and complexity of the posterior is Hamiltonian MCMC (HMCMC) [6]. In HMCMC, one generates proposal samples by integrating along trajectories of a Hamiltonian dynamical system constructed from combining the posterior distribution of interest with a momentum distribution. This is then followed by the Metropolis acceptance step to yield a new sample. Formally a

joint distribution is constructed with state variables $(\mathbf{x}, \mathbf{p})$:

$$p^H(\mathbf{x}, \mathbf{p}) \propto \exp(-U(\mathbf{x}) - K(\mathbf{p})), \tag{5}$$

$$p(\mathbf{x}) \propto \exp(-U(\mathbf{x})), \tag{6}$$

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{p}, \tag{7}$$

where we omit a normalizing constant and $\mathbf{p}$ are the momentum variables which are added. This form is motivated from statistical physics by the canonical distribution of energy states of a system, where U and K denote the potential and kinetic energy respectively, and the Hamiltonian (total energy) is $H = U + K$ [6]. HMCMC samples from $p^H(\mathbf{x}, \mathbf{p})$, and we can obtain the marginal distribution of $\mathbf{x}$ from the samples. H defines a dynamical system, which is a set of differential equations used to evolve $\mathbf{x}, \mathbf{p}$ forward in time from an initial sample.

In practice, these equations are integrated numerically, characterized by a step size ϵ and a number of steps L such that $L\epsilon$ is the time period over which a sample trajectory is evolved. The most common numerical scheme is the leapfrog scheme, which we write below for one time step with initial condition $(\mathbf{x}, \mathbf{p})$ and result $(\mathbf{x}', \mathbf{p}')$.

$$\mathbf{p}^{\frac{1}{2}} = \mathbf{p} - \frac{\epsilon}{2}\partial_x U(\mathbf{x}),\ \mathbf{x}' = \mathbf{x} + \epsilon\mathbf{p}^{\frac{1}{2}},\ \mathbf{p}' = \mathbf{p} - \frac{\epsilon}{2}\partial_x U(\mathbf{x}'). \tag{8}$$

Given an initial $\mathbf{x_0}$, an initial momentum $\mathbf{p_0}$ is sampled from a distribution, usually a standard Gaussian [4]. The proposed sample from running the dynamics, $(\mathbf{x}', \mathbf{p}')$, is then accepted in a Metropolis-Hastings step with probability $\alpha = min(1, \frac{\exp(-U(\mathbf{x}) - \frac{1}{2}\mathbf{p}^T\mathbf{p})}{\exp(-U(\mathbf{x_0}) - \frac{1}{2}\mathbf{p_0}^T\mathbf{p_0})})$. Ideally, this procedure is then repeated until the convergence of the samples to the distribution, with the output of each proposal becoming the new initial sample. The main advantage of HMCMC is that it generally proposes samples which are far away from the initial sample, thus efficiently exploring the posterior, while maintaining reasonable acceptance probabilities [4]. A state of the art HMCMC sampler called the No U Turn Sampler (NUTS) [10] improves on standard HMCMC by adaptively tuning L, ϵ to manage the distance between samples and acceptance probability. HMCMC can perform poorly in certain circumstances, in particular, in highly curved sample spaces such as those that might arise in the posteriors derived from parameter estimation of complex models [8].

2.2 L2HMC

Levy et. al. [15] recently proposed a framework called L2HMC which parametrizes the standard HMCMC sampler with a neural network and maximizes the expected distance between samples through minimization of a loss function which rewards large expected squared distances between samples. Furthermore, the parametrization is carefully tailored to preserve detailed balance and have a tractable Jacobian for the correction of the acceptance probability due to the

potential non-volume preserving dynamics. The algorithm of L2HMC is structurally similar to standard HMCMC, but modifications are made to the proposal stage. First, for each step t, $1 \leq t \leq L$, a random binary mask $m_t \in \{0,1\}^n$ is constructed such that approximately half of the entries of the mask are 1. The conjugate mask is denoted as m_t^c. Instead of updating $\mathbf{x}$ in one step according to the classical algorithm, the update is split into two steps each updating only the variables of $\mathbf{x}$ corresponding to m_t, m_t^c separately. These are denoted as $\mathbf{x}_{m_t} = \mathbf{x} \odot m_t$ and $\mathbf{x}_{m_t^c} = \mathbf{x} \odot m_t^c$ respectively, where $\odot$ is the component-wise multiplication operator. Each update equation is modified with scaling factors for each term depending on only variables which are not being updated. Concretely, let $\zeta_1 = (\mathbf{x}, \partial_x U(\mathbf{x}'), t)$. Then $\mathbf{p}$ is first updated according to

$$\mathbf{p}^{\frac{1}{2}} = \mathbf{p} \odot \exp(\frac{\epsilon}{2} S_p(\zeta_1)) - \frac{\epsilon}{2} \partial_x U(\mathbf{x}) \odot \exp(\epsilon Q_p(\zeta_1)) + T_p(\zeta_1), \tag{9}$$

where $S_p, Q_p, T_p : \mathbb{R}^{2n+1} \to \mathbb{R}^n$ are scaling functions parameterized by a neural network. Let $\zeta_2 = (\mathbf{x}_{\mathbf{m_t^c}}, \mathbf{p}, t)$ and $\zeta_3 = (\mathbf{x}^{\frac{1}{2}}_{\mathbf{m_t}}, \mathbf{p}, t)$. Then $\mathbf{x}$ is updated according to

$$\mathbf{x}^{\frac{1}{2}} = \mathbf{x}_{\mathbf{m_t^c}} + m_t \odot \left[\mathbf{x} \odot \exp(\epsilon S_x(\zeta_2)) + \epsilon(\mathbf{p}^{\frac{1}{2}} \odot \exp(\epsilon Q_x(\zeta_2)) + T_x(\zeta_2))\right], \tag{10}$$

$$\mathbf{x}' = \mathbf{x}_{\mathbf{m_t}} + m_t^c \odot \left[\mathbf{x} \odot \exp(\epsilon S_x(\zeta_3)) + \epsilon(\mathbf{p}^{\frac{1}{2}} \odot \exp(\epsilon Q_x(\zeta_3)) + T_x(\zeta_3))\right], \tag{11}$$

where $S_x, Q_x, T_x : \mathbb{R}^{2n+1} \to \mathbb{R}^n$ are also scaling functions parameterized by a neural network. Finally, let $\zeta_4 = (\mathbf{x}', \partial_x U(\mathbf{x}'), t)$:

$$\mathbf{p}' = \mathbf{p}^{\frac{1}{2}} \odot \exp(\frac{\epsilon}{2} S_p(\zeta_4)) - \frac{\epsilon}{2} \partial_x U(\mathbf{x}') \odot \exp(\epsilon Q_p(\zeta_4)) + T_p(\zeta_4). \tag{12}$$

These learned scaling functions, structured as a two layer neural network, can allow the sampler to learn, for example, how to carefully navigate regions of high curvature in the parameter space rather than having to manipulate ϵ and L to accomplish this. As in NUTS [10], the time reversed version of the above dynamics can also be used to propose samples, and L2HMC takes a random combination of the forward and backward dynamics proposal as the final proposal [15]. Let θ be the vector of parameters of the above functions. After each complete cycle of proposal and acceptance, the loss function is optimized using Adam [12]. Concretely, let $\xi = (x, p)$ be the initial sample, and $\xi' = (x', p')$ be the sample after the acceptance step. Let $\delta(\xi, \xi') = \|x - x'\|_2^2$ and $A(\xi', \xi)$ denote the acceptance probability. Then the loss function $\mathcal{L}(\theta)$ used is

$$\mathcal{L}(\theta) = \mathbb{E}_{p(\xi)}\left[-\frac{\delta(\xi, \xi')A(\xi', \xi)}{\lambda^2} + \frac{\lambda^2}{\delta(\xi, \xi')A(\xi', \xi)}\right], \tag{13}$$

where the expectation is taken over the batch of samples over which the training is taking place. λ is the typical length scale of the distribution, which Levy et. al. set in the case of a multivariate normal distribution, as the smallest standard deviation in the covariance matrix. For simplicity, in eq. 13, we omit an additional term with identical form as above [15] designed to enhance burn-in by using an arbitary proposal distribution. Fig 1 shows a flowchart of the algorithm of sampling with the neural network parametrization.

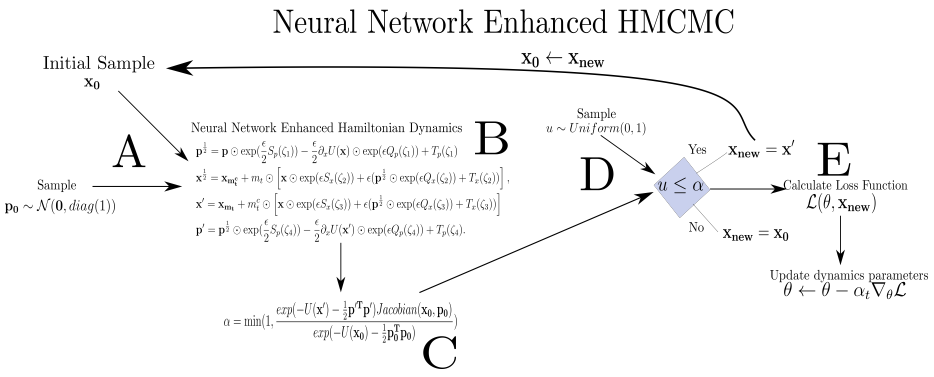

Fig. 1. Flowchart of the training algorithm for neural network parametrization of HMCMC. The components of the algorithm are very similar to standard HMCMC; however, the differences lie in the altered dynamics/proposal stage and the update of the neural network parameters after each step. When sampling, the network parameters are fixed at the last training values.

3 Methods

3.1 Neural Network Enhanced Hamiltonian MC (NNEHMC)

The first contribution of this paper is to extend L2HMC by augmenting the loss function to balance acceptance probability and the distance between samples. Let $A_{HMC}(\xi', \xi)$ denote the acceptance probability used in standard HMCMC. We introduce the loss function

$$\mathcal{L}^{NNEHMC}(\theta) = \mathbb{E}_{p(\xi)}\left[-\delta(\xi, \xi')A(\xi', \xi) - \beta A_{HMC}(\xi', \xi)\right]. \tag{14}$$

We removed the reciprocal distance term as it did not meaningfully change the dynamics of the sampling in the distributions we considered. Further, we do not integrate the time reversed dynamics in our sampling. We argue that this form of loss function more faithfully and naturally enhances the desirable properties of Hamiltonian dynamics. In theory, Hamiltonian dynamics preserve energy along trajectories; hence, since the probability of a sample is proportional to $\exp(-H)$, the acceptance probability is always 1 [4]. However, the introduction of numerical integration causes violation of this property; nonetheless HMCMC still, generally, provides high acceptance rates, with additional tuning possible through changing ϵ or L. One can view this tuning as reducing the error of the leapfrog scheme such that the numerical integration gets closer and closer to the theoretical Hamiltonian dynamics with its property of preserving energy. However, the dynamics of L2HMC is no longer a numerical approximation of Hamiltonian dynamics due to the scaling terms. Hence, while it is valid as an MCMC sampler, there is no theoretical basis for the sampler to produce samples with high acceptance probabilities which are largely independent of the squared distance as in standard HMCMC.

As a way of both inducing the sampler to remain close to Hamiltonian dynamics and balancing the acceptance probability and mixing, we add the negative standard HMCMC acceptance probability in the loss function, with the parameter β enforcing the tradeoff between it and the negative expected squared distance. We argue that this loss function can lead to two desirable properties. First, it could lead to faster mixing and faster convergence than in L2HMC since, from the beginning, it can balance learning the standard, approximately energy preserving Hamiltonian dynamics with opportunities to move great distances. One can interpret the additional term as enforcing approximate conformity, in some sense, to Hamiltonian dynamics, mediated by β. Second, crucial for parameter estimation, we argue that this term helps to keep the sampler stable when exploring high curvature regions. In these regions, the acceptance probability can drop to zero easily due to large distance steps and numerical issues can develop [8]. The acceptance probability of the neural network parametrized sampler differs from the classical acceptance by the Jacobian of the new, scaled dynamics, which is identically 1 in the standard case. Hence, if the standard acceptance probability is the dominant term, it can still enforce high acceptance probabilities for the sampler. We thus treat the neural network as an enhancement that allows the sampler to learn "approximate" Hamiltonian dynamics which can balance and enhance the desirable properties of HMCMC while learning to minimize its weaknesses. We henceforth refer to our sampler as Neural Network Enhanced Hamiltonian MC (NNEHMC).

In the results, we compare the performance of NNEHMC and L2HMC on a toy distribution also tested in [15]. The distribution is a strongly correlated 2-D Normal distribution with mean zero, and a covariance matrix obtained from $diag(100, 0.1)$ rotated by 45 degrees. For both samplers, we use the same $\epsilon = 0.1, L = 10$, initialize with the same 200 samples, train in batches of 200 samples for 5000 steps, then fix the neural network parameters and sample 200 chains for 2000 steps using the trained sampler [15]. We tune β in NNEHMC by looking at the autocorrelation analysis and the acceptance probabilities. We set $\lambda = 0.1$ as is done in [15]. We compare the two samplers by the autocorrelation of the samples as well as the effective sample size derived from the autocorrelation, which can be seen as a measure of how many of the samples are "useful" for inference [4].

3.2 Biophysical Parameter Estimation

The second contribution of this paper to apply NNEHMC to biophysical parameter estimation in a recently proposed MRI model, the Multi Echo Spherical Mean Technique (MESMT) [5].

Multi Echo Spherical Mean Technique (MESMT). MRI Diffusometry and T_2 relaxometry can be combined into a multi-modal analysis which jointly estimates diffusivities, T_2's, and water volume fractions of different tissue compartments. The extended spherical mean technique (SMT) framework introduced

by [5] is one example of this, generalizing the diffusion MRI model SMT [11] by including the effects of changing the echo time T_E in the acquisition on the MRI signal and using the additional information to simultaneously estimate the T_2's and diffusivities of the compartments in brain white matter. The model signal is a function of b and T_E. For given b, T_E

$$\begin{aligned} Model(T_E, b, \mathbf{x}) =& v_I \exp(\frac{-T_E}{T_2^I}) \frac{\sqrt{\pi}\,\mathrm{erf}(\sqrt{b\lambda_\parallel})}{2\sqrt{b\lambda_\parallel}} \\ &+ v_E \exp(\frac{-T_E}{T_2^E}) \exp(-b\lambda_\perp) \frac{\sqrt{\pi}\,\mathrm{erf}(\sqrt{b(\lambda_\parallel - \lambda_\perp)})}{2\sqrt{b(\lambda_\parallel - \lambda_\perp)}} \\ &+ v_{CSF} \exp(\frac{-T_E}{T_2^{csf}}) \exp(-bD_{csf}), \end{aligned}$$

where v_I, v_E, v_{CSF} are the volume fractions of the intra-axonal, extra-axonal, and cerebrospinal fluid (CSF) compartments respectively, $\lambda_\parallel, \lambda_\perp$ are the parallel and perpendicular diffusivities. The likelihood of this model is constructed as in the introduction. In the fitting, we fix the values of $T_2^{csf} = 2s$ and $D_{csf} = 0.003\frac{mm^2}{s}$ at those of free water [11, 16], but we allow v_{CSF} to be free.

Using our proposed sampler, we can bound the T_2's and diffusivities based on prior physical knowledge [11, 16] so we use uniform priors as follows: $T_2^I \sim \mathcal{U}(5ms, 200ms)$, $T_2^E \sim \mathcal{U}(5ms, 100ms)$, $\lambda_\parallel \sim \mathcal{U}(0.0005\frac{mm^2}{s}, 0.003\frac{mm^2}{s})$, $\lambda_\perp \sim \mathcal{U}(0.0001\frac{mm^2}{s}, 0.0005\frac{mm^2}{s})$.

Experimental Setup. We simulate three datasets using three different $T_E's = 50, 75, 100ms$, with three $b = 300, 2150, 4000s/mm^2$ values per dataset, and fit them simultaneously. The ground truth parameters are as follows: $v_I = 0.5, v_E = 0.3, v_{CSF} = 0.2, \lambda_\parallel = 0.0015\frac{mm^2}{s}, \lambda_\perp = 0.0002\frac{mm^2}{s}, T_2^I = 140ms, T_2^E = 70ms$. Since the volume fractions must sum to one, we use a 3D, symmetric Dirichlet prior for the volumes: $(v_I, v_E, v_{CSF}) \sim \mathbf{Dir}(1.0, 1.0, 1.0)$. We generated one hundred signals from the ground truth parameters by adding one hundred realizations of Gaussian noise with a standard deviation of $\sigma = \frac{1}{120}$. We simulated many instances of a typical diffusion acquisition using Dmipy [7] with a mean SNR of 20 on the b_0 data, then performed spherical averaging on each instance. The standard deviation of the resulting signals over the instances was estimated to be around $\frac{1}{120}$, which motivates our setting of σ. We then estimated the parameters over each signal using NUTS, L2HMC, and NNEHMC within the Bayesian framework described in Section 3.2. We also show a fitting using constrained least squares (LSQ). We imposed the same constraints in both the probabilistic and deterministic fittings. We initialize NUTS with a variational inference estimate [13], and use 1000 samples for burn-in and 1000 samples for inference. We initialize L2HMC and NNEHMC with the first 50 samples of the NUTS burn-in, train on batches of 50 samples for 1000 steps, then fix the parameters of the network and use the trained sampler to generate 1000 samples for inference. We set $\lambda = \sigma$, since it is roughly the length scale of the distribution.

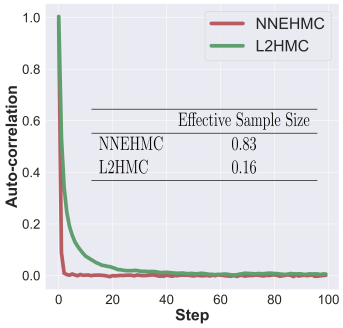

Fig. 2. Plot of the average autocorrelation of 200 chains of length 2000 for L2HMC and NNEHMC with the corresponding effective sample size. The autocorrelation and effective sample size are calculated as in [15]. We see that NNEHMC mixes faster in sampling steps and has a larger effective sample size.

In the results, we report the relative absolute error as follows: letting g denote the ground truth parameter and e as the estimate, the relative absolute error is computed as $|g - e|/g$. We note that we scale b by 10e−2 and the diffusivities by 10e2 in the sampling and results.

4 Results and Discussion

4.1 Strongly Correlated Gaussian

In Fig 2, we show the average autocorrelation of the samples over 50 chains from sampling the strongly correlated Gaussian as a function of steps in the chain as well as a table with the effective sample sizes derived from the autocorrelation. We note that NNEHMC mixes faster and has an effective sample size almost eight times larger than that of L2HMC. Further, on the same computer, NNEHMC requires 179s of computation time while L2HMC requires 1561s. This is mostly because NNEHMC does not use the time reversed dynamics. In cases with tractable distributions and derivatives one can also speed up the sampling by using GPU computation [2].

4.2 Multi Echo Spherical Mean Technique (MESMT)

In Fig. 3, 4, we show the relative absolute error and an example of the marginal posterior probability distributions produced by the MCMC samplers.

We can see that, in general, the MCMC samplers are more accurate and precise than the least squares fitting. However, we see that NNEHMC and L2HMC significantly outperform NUTS in estimating volume fractions and $\lambda_{\|}$, even

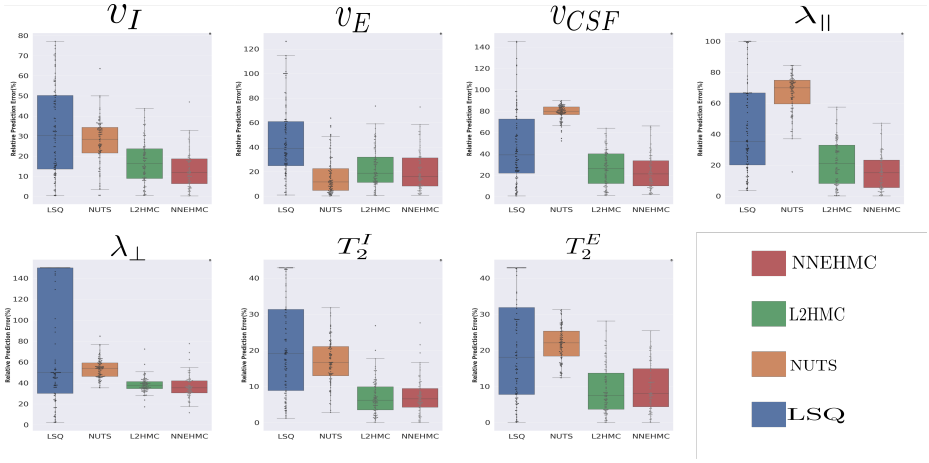

Fig. 3. Box plots of relative absolute errors from ground truth using least squares (blue), NUTS (orange), L2HMC (green), and NNEHMC (red). We note that in general, NNEHMC has the lowest mean error and variance. Further, NUTS has significant issues in the estimation of $\lambda_{||}$ and the volume fractions, which is not observed in NNEHMC or L2HMC.

though they all start from the same initialization. Inspection of the probability distributions reveals that NUTS gives distributions biased away from the ground truth for these parameters. Furthermore, we note that NNEHMC generally outperforms L2HMC regarding the accuracy and variance of the estimates. Unlike in the toy example, where we knew the precise mean and variance of the distribution, we can only compute an approximate autocorrelation analysis in this case. We obtained an effective sample size of 1.5e−3 for NNEHMC and 1.9e−3 for L2HMC. However, the mean computation times for a single signal are $280s$ for NNEHMC and $443s$ for L2HMC.

Furthermore we emphasize that using L2HMC on this model is numerically unstable. By changing the random seed in our implementation, 18 out of the 100 trials with L2HMC either decline to and remain at zero acceptance probability for all chains by the end of training or encounter numerical errors (NaN, infinities). This can happen, for instance, if the proposal samples move too far away. NNEHMC was robust to such changes. We do not consider these results in the analysis since they are invalid for parameter estimation and would artificially bias the results for L2HMC negatively. In order for NUTS not to develop similar numerical issues, we had to set a desired acceptance probability of 99%. It is probable that the poor results of NUTS stem, in part, from inefficient sampling due to a highly curved parameter space which the adaptive tuning could not overcome. Thus, we can see that the parametrization with a neural network can enable efficient sampling of problematic regions in parameter space; however, regularization with an acceptance probability term is needed for stability.

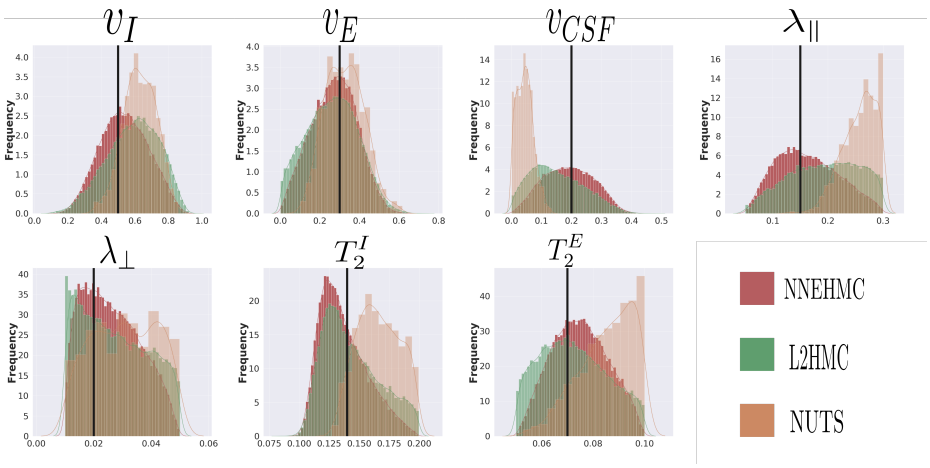

Fig. 4. Representative plots of the marginal probability distributions for each parameter, where the black vertical line denotes the ground truth value. We can see that NNEHMC provides informative posterior distributions from which inference seems justified, while NUTS provides quasi-uniform distributions and distributions biased towards the parameter bounds.

5 Conclusion

In this paper, we have proposed and tested a parametrization of Hamiltonian MCMC with a neural network (NNEHMC) which jointly optimizes sample acceptance probability and distances between successive samples in order to efficiently and stably sample probability distributions, particularly in regions of parameter space with high curvature. Such regions frequently occur in the probabilistic estimation of parameters in bio-physical models since the posterior distributions are parametrized, in part, by highly nonlinear models. We show on a recently proposed MRI model that the neural network enhancement provides parameter estimates which are more accurate and precise than those given by a least squares fitting and the state of the art NUTS and L2HMC samplers; in addition NNEHMC provides more numerically stable sampling than NUTS or L2HMC. Furthermore, we show that the neural network parametrization provides qualitatively different and more informative posterior distributions than those produced from NUTS; NNEHMC can produce posterior distributions which are Gaussian-like centered near the correct parameter values. This highlights the potential of augmenting MCMC methods with neural networks to improve probabilistic estimation of parameters in biophysical models.

Acknowledgements. Thomas Yu is supported by the European Union's Horizon 2020 program under the Marie Sklodowska-Curie project TRABIT (agreement No 765148). Marco Pizzolato is supported by the SNSF under Sinergia

CRSII5_170873. This work is also supported by the Center for Biomedical Imaging (CIBM) of the Universities of Geneva and Lausanne and the EPFL as well as the foundations of Leenaards and Louis-Jeantet.

References

1. Balakrishnan, N., Nevzorov, V.B.: A Primer on Statistical Distributions. John Wiley & Sons (2004)
2. Beam, A.L., Ghosh, S.K., Doyle, J.: Fast Hamiltonian Monte Carlo using GPU Computing. Journal of Computational and Graphical Statistics **25**(2), 536–548 (2016)
3. Betancourt, M.: A conceptual introduction to Hamiltonian Monte Carlo. arXiv preprint arXiv:1701.02434 (2017)
4. Brooks, S., Gelman, A., Jones, G., Meng, X.L.: Handbook of Markov Chain Monte Carlo. CRC press (2011)
5. Canales Rodriguez, E.J., Pizzolato, M., Aleman-Gomez, Y., Kunz, N., Pot, C., Thiran, J.P., Daducci, A.: Unified multi-modal characterization of microstructural parameters of brain tissue using diffusion MRI and multi-echo T2 data. In: Joint Annual Meeting ISMRM-ESMRMB (2018)
6. Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Physics Letters B **195**(2), 216–222 (1987)
7. Fick, R., Wassermann, D., Deriche, R.: Mipy: An Open-Source Framework to improve reproducibility in Brain Microstructure Imaging. In: OHBM 2018-Human Brain Mapping. pp. 1–4 (2018)
8. Girolami, M., Calderhead, B.: Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **73**(2), 123–214 (2011)
9. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications (1970)
10. Hoffman, M.D., Gelman, A.: The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. Journal of Machine Learning Research **15**(1), 1593–1623 (2014)
11. Kaden, E., Kelm, N.D., Carson, R.P., Does, M.D., Alexander, D.C.: Multi-compartment microscopic diffusion imaging. NeuroImage **139**, 346–359 (2016)
12. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic Differentiation Variational Inference. The Journal of Machine Learning Research **18**(1), 430–474 (2017)
14. Ledoux, M.: The Concentration of Measure Phenomenon. No. 89, American Mathematical Soc. (2001)
15. Levy, D., Hoffman, M.D., Sohl-Dickstein, J.: Generalizing Hamiltonian Monte Carlo with Neural Networks. arXiv preprint arXiv:1711.09268 (2017)
16. MacKay, A.L., Laule, C.: Magnetic Resonance of Myelin Water: An in vivo Marker for Myelin. Brain Plasticity **2**(1), 71–91 (2016)
17. Sengijpta, S.K.: Fundamentals of Statistical Signal Processing: Estimation theory (1995)