Proceedings of Machine Learning Research 96:1–21, 2018. 1st Symposium on Advances in Approximate Bayesian Inference

Likelihood-free inference with emulator networks

Jan-Matthis Lueckmann[1,2] JAN-MATTHIS.LUECKMANN@CAESAR.DE
Giacomo Bassetto[1,2] GIACOMO.BASSETTO@CAESAR.DE
Theofanis Karaletsos[3] THEOFANIS@UBER.COM
Jakob H. Macke[1,2,4] MACKE@TUM.DE

Abstract

Approximate Bayesian Computation (ABC) provides methods for Bayesian inference in simulation-based models which do not permit tractable likelihoods. We present a new ABC method which uses probabilistic neural *emulator* networks to learn synthetic likelihoods on simulated data – both 'local' emulators which approximate the likelihood for specific observed data, as well as 'global' ones which are applicable to a range of data. Simulations are chosen adaptively using an acquisition function which takes into account uncertainty about either the posterior distribution of interest, or the parameters of the emulator. Our approach does not rely on user-defined rejection thresholds or distance functions. We illustrate inference with emulator networks on synthetic examples and on a biophysical neuron model, and show that emulators allow accurate and efficient inference even on problems which are challenging for conventional ABC approaches.

1. Introduction

Many areas of science and engineering make extensive use of complex, stochastic, numerical simulations to describe the structure and dynamics of the processes being investigated (Karabatsos and Leisen, 2017). A key challenge in simulation-based science is linking simulation models to empirical data: Bayesian inference provides a general and powerful framework for identifying the set of parameters which are consistent both with empirical data and prior knowledge. One of the key quantities required for statistical inference, the likelihood of observed data given parameters, $\mathcal{L}(\boldsymbol{\theta}) = p(\mathbf{x}_o|\boldsymbol{\theta})$, is typically intractable for simulation-based models, rendering conventional statistical approaches inapplicable.

Approximate Bayesian Computation (ABC) aims to close this gap (Beaumont et al., 2002), but classical algorithms (Pritchard et al., 1999; Marjoram et al., 2003) scale poorly to high-dimensional non-Gaussian data, and require *ad-hoc* choices (i.e., rejection thresholds, distance functions and summary statistics) which can significantly affect both computational efficiency and accuracy. In *synthetic likelihood* approaches to ABC (Wood, 2010; Ong et al., 2016; Price et al., 2018), one instead uses density estimation to approximate the likelihood $p(s(\mathbf{x}_o)|\boldsymbol{\theta})$ on summary statistics $s(\cdot)$ of simulated data. A recent proposal by Järvenpää et al. (2017), Gutmann and Corander (2016) uses a Gaussian process ($\mathcal{GP}$) to approximate the

1. Computational Neuroengineering, Department of Electrical and Computer Engineering, Technical University of Munich, Germany
2. Neural Systems Analysis, Research Center caesar, an associate of the Max Planck Society, Bonn, Germany
3. Uber AI Labs, Uber Technologies, Inc., San Francisco, CA
4. Part of this work was done while J.H.M was at the Centre for Cognitive Science, Technische Universität Darmstadt, Germany

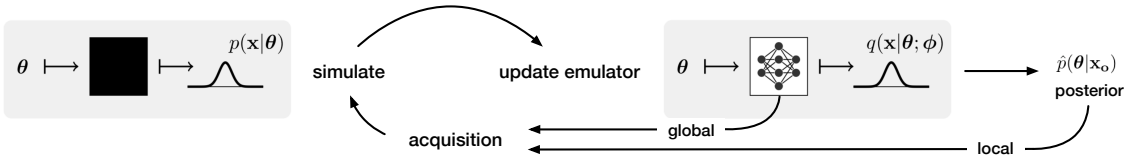

Figure 1: **Likelihood-free inference with emulator networks.** Our goal is to perform approximate Bayesian inference on simulator-models, i.e. models from which we can generate samples, but for which we can not evaluate likelihoods. We learn a tractable probabilistic emulator $q(\mathbf{x}|\boldsymbol{\theta};\phi)$ approximating the simulator $p(\mathbf{x}|\boldsymbol{\theta})$. The emulator then serves as a synthetic likelihood to obtain an approximate posterior. To train the emulator using a low number of simulations, we use active learning to select informative samples: The acquisition rule is either based on the current posterior estimate (if observed data $\mathbf{x}_o$ is given, 'local' learning), or on our uncertainty about the weights of the emulator network ('global' learning).

distribution of the discrepancy $d(s(\mathbf{x}), s(\mathbf{x}_o))$ as a function of $\boldsymbol{\theta}$, and Bayesian Optimization to propose new parameters. While this approach can be very effective even with a small number of simulations, it still requires summary statistics, choice of a distance function $d(\cdot,\cdot)$, and relies on assuming a homoscedastic $\mathcal{GP}$.

The goal of this paper is to scale synthetic-likelihood methods to multivariate and (potentially) non-Gaussian, heteroscedastic data. We use neural-network based conditional density estimators (which we call 'emulator networks', inspired by classical work on emulation methods; Kennedy and O'Hagan, 2002), to develop likelihood-free inference algorithms which are efficient, flexible, and scale to high-dimensional observations. Our approach does not require the user to specify rejection thresholds or distance functions, or to restrict oneself to a small number of summary statistics.

2. Likelihood-free inference with emulator networks

Our goal is to obtain an approximation to the true posterior $p(\boldsymbol{\theta}|\mathbf{x}_o)$ of a black-box simulator model, i.e. models from which we can generate samples $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$, but for which we cannot evaluate likelihoods $\mathcal{L}(\boldsymbol{\theta})$. To solve this task, we learn a synthetic likelihood function $\hat{\mathcal{L}}(\boldsymbol{\theta})$ by training a conditional density estimator on simulated data. We actively propose parameters for simulations, since simulations are often the dominant cost in ABC: Therefore, we want to keep the number of calls to the simulator as low as possible (Fig. 1).

Core to our approach is an emulator $q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi})$, a conditional density estimator with parameters $\boldsymbol{\phi}$ that approximates the simulator $p(\mathbf{x}|\boldsymbol{\theta})$. Having collected an initial simulated dataset $\mathcal{D}$, e.g. by repeatedly drawing from the prior $p(\boldsymbol{\theta})$ and simulating data, the emulator is trained. We actively select new locations $\boldsymbol{\theta}^*$ for which to simulate new data points $\mathcal{D}^* = \{(\boldsymbol{\theta}^*, \mathbf{x}^*)\}$ to keep the number of calls to the (potentially computationally expensive) simulator low. $\mathcal{D}^*$ is appended to the dataset, the emulator is updated, and the active learning loop repeats. The emulator defines a synthetic likelihood function $\hat{\mathcal{L}}(\boldsymbol{\theta}) = q(\mathbf{x} = \mathbf{x}_o|\boldsymbol{\theta};\boldsymbol{\phi})$

that we use to find an approximate posterior, which is proportional to $\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o) := \hat{\mathcal{L}}(\boldsymbol{\theta})p(\boldsymbol{\theta})$. This approach is summarized in Appendix A in form of an algorithm.

Thus, our approach requires (1) an emulator, i.e., a flexible conditional density estimator, (2) an approach for learning the emulator on simulated data and expressing our uncertainty about its parameters, (3) an acquisition rule for proposing new sampling locations, and (4) an inference procedure for obtaining the posterior distribution from the synthetic likelihood and the prior. We will describe these steps in the following.

2.1. Choice of emulator

We use neural network based emulators $q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi})$: parameters $\boldsymbol{\theta}$ are given as inputs to the network, and the network is trained to approximate $p(\mathbf{x}|\boldsymbol{\theta})$. In contrast to traditional synthetic likelihood approaches (Wood, 2010), we are not restricted to using a (multivariate) normal distribution to approximate the conditional density $p(\mathbf{x}|\boldsymbol{\theta})$. The output form of the emulator is chosen according to our knowledge regarding the conditional density of the simulator. In our second example application, we e.g. model $\mathbf{x}|\boldsymbol{\theta}$ as a binomial distribution over 8-bit integer pixel values, and in the third example we model a categorical distribution. If the noise model of the simulation process is unknown, flexible conditional density estimators such as conditional autoregressive models (Oord et al., 2016; Papamakarios et al., 2018) can be readily used in our approach.

2.2. Inference on the parameters of the emulator

We use probabilistic neural networks, i.e. we represent uncertainty about the parameters $\boldsymbol{\phi}$ of the emulator $q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi})$. We then use these uncertainties to guide the acquisition of training data for the emulator using active learning (as discussed in the next section).

In the Bayesian framework, uncertainty is represented through the posterior distribution. Multiple approaches for estimating the posterior distributions over neural network parameters have been proposed, including MCMC methods to draw samples from the full posterior (Welling and Teh, 2011; Chen et al., 2014) and variational methods, e.g. using factorising posteriors (Blundell et al., 2015) or normalizing flows (Louizos and Welling, 2017). Finally, deep ensemble approaches (Lakshminarayanan et al., 2017) represent predictive distributions through ensembles of networks. They have the advantage of not requiring the choice of a functional form of the approximation, and are simple to set up.

Our approach can be applied with any method that represents uncertainty over network parameters. In our experiments, we use deep ensembles to represent uncertainty about $\boldsymbol{\phi}$, as we found them to combine simplicity with good empirical performance. Instead of training a single emulator network and inferring its posterior distribution, we train an ensemble of M networks with parameters $\{\boldsymbol{\phi}_m\}_{m=1}^M$. From here on, we treat $\boldsymbol{\phi}_m$ as if they were samples from $p(\boldsymbol{\phi}|\mathcal{D})$, the posterior over network parameters given data. (In practice, these samples will describe local maxima of the posterior.) The posterior-predictive distribution is approximated by $\mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\big[q(\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi})\big] \approx \frac{1}{M}\sum_{m=1}^M q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi}_m)$.

Networks are trained supervised with data $\mathcal{D} = \big\{(\boldsymbol{\theta}_n, \mathbf{x}_n)\big\}_{n=1}^N$. During training, the parameters of the networks are optimized subject to the loss $-\sum_{m=1}^M \sum_{n=1}^N \log q(\mathbf{x}_n|\boldsymbol{\theta}_n;\boldsymbol{\phi}_m)$ w.r.t. $\boldsymbol{\phi}$ (a proper scoring rule as discussed in Lakshminarayanan et al., 2017). Networks in the ensemble are initialized differently, and data points are randomly shuffled during training.

2.3. Acquisition rules

We use active learning to selectively acquire new samples. We distinguish between two scenarios: In the first, we have particular observed data $\mathbf{x}_o$ available, and train a *local emulator* which approximates the likelihood near $\mathbf{x}_o$. This approach requires learning a new emulator for each new observed data $\mathbf{x}_o$.

We also consider a second scenario, in which we learn a *global emulator* – which approximates $p(\mathbf{x}|\boldsymbol{\theta})$ globally. Learning a global emulator is more challenging and may potentially require more flexible density estimators. However, once the emulator is learned, we can readily approximate the likelihood for *any* $\mathbf{x}_o$, therefore amortizing the cost of learning the emulator.

The two scenarios call for different acquisition functions for proposing new samples, which we will discuss next.

2.3.1. Acquisitions for local emulator learning

With given $\mathbf{x}_o$, we want to learn a local emulator that allows us to derive a good approximation to the (unnormalized) posterior $\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o) \propto \mathbb{E}_{\phi|\mathcal{D}}\big[q(\mathbf{x}=\mathbf{x}_o|\boldsymbol{\theta};\boldsymbol{\phi})\big]p(\boldsymbol{\theta})$.

As we are interested in increasing our certainty about the posterior, we target its variance, $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o,\boldsymbol{\phi})]$, where $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}$ denotes that we take the variance with respect to the posterior over network weights given data $\mathcal{D}$. Thus, we use an acquisition rule which targets the region of maximum variance in the predicted (unnormalized) posterior,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o,\boldsymbol{\phi})] = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \log\sqrt{\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\hat{\mathcal{L}}(\boldsymbol{\theta})]}. \tag{1}$$

We approximate $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}$ with the sample variance across $\boldsymbol{\phi}_m$ drawn from the posterior over networks. We refer to this rule as the *MaxVar* rule (Järvenpää et al., 2017). We optimize this acquisition rule by using gradient descent, making use of automatic differentiation to take gradients with respect to $\boldsymbol{\theta}$ through the synthetic likelihood specified by the emulator.

2.3.2. Acquisitions for global emulator learning

A global emulator may be used to do inference once $\mathbf{x}_o$ becomes available. Here, the goal for active learning is to bring the emulator $q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi})$ close to the simulator $p(\mathbf{x}|\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$s using as few runs of the simulator as possible. We use a rule based on information theory from the active learning literature (Houlsby et al., 2011; Gal et al., 2017; Depeweg et al., 2017). We refer to the rule

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{I}[\mathbf{x},\boldsymbol{\phi}|\boldsymbol{\theta},\mathcal{D}] = \arg\max_{\boldsymbol{\theta}} \mathbb{H}[\mathbf{x}|\boldsymbol{\theta},\mathcal{D}] - \mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\big[\mathbb{H}[\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi}]\big] \tag{2}$$

as the maximum mutual information rule (*MaxMI*). See Appendix B for details.

2.4. Deriving the posterior distribution from the emulator

Once we have learned the emulator, we use Hamiltonian Monte Carlo (HMC, Neal, 2010) to draw samples from our approximate posterior, using the emulator-based synthetic likelihood. We generate samples of $\boldsymbol{\theta}$ drawn from the distribution $\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o) = \mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\big[q(\mathbf{x}_o|\boldsymbol{\theta})\big]p(\boldsymbol{\theta})$. In practice, we sample $\boldsymbol{\theta}$ from each ensemble member individually and use the union of all

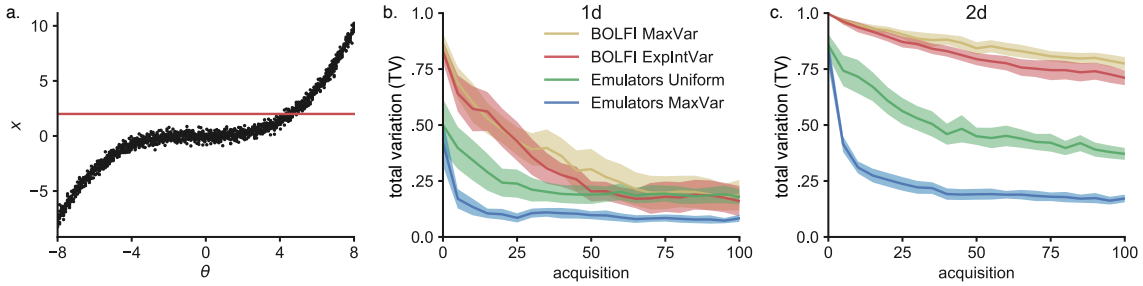

Figure 2: **Inference on simulator with Gaussian noise**. **a.** Data is generated from $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|f(\boldsymbol{\theta}), \boldsymbol{\Sigma})$ with cubic non-linearity. We illustrate posterior inference $p(\boldsymbol{\theta}|\mathbf{x}_o)$ given $\mathbf{x}_o = 2$ (red line at $\mathbf{x}_o$). **b.** In 1-D, emulator-based inference with *MaxVar* acquisitions leads to faster convergence to true posterior than uniform sampling, or BOLFI. Total variation (TV) is measured between true and approximate posteriors. 100 acquisitions starting from $N_{\text{initial}} = 10$ initial points. Lines are means and SEMs from 20 runs. **c.** Same problem, but $\mathbf{x}$ and $\boldsymbol{\theta} \in \mathbb{R}^2$, non-linearity applied point-wise, starting from $N_{\text{initial}} = 25$ points.

samples as a draw from the approximate posterior. We could also obtain the posterior through variational inference, but here prefer to retain full flexibility in the shape of the inferred posterior.

3. Results

We demonstrate likelihood-free inference with emulator networks on three examples: i) we show that emulators are competitive with state-of-the-art on an example with Gaussian observations; ii) we demonstrate the ability of emulators to work with high-dimensional observations while learning to amortize the simulator; iii) we show an application from neuroscience, and infer the posterior over parameters of a biophysical neuron model.

i) Low-dimensional example: Simulator with Gaussian observations

We first demonstrate emulator networks on a non-linear model between parameters and data, corrupted by additive Gaussian observation noise: data is generated according to $\mathbf{x}_i \sim \mathcal{N}(\cdot|f(\boldsymbol{\theta}), \boldsymbol{\Sigma})$, $i = 1 \dots n$, where $f(\boldsymbol{\theta})$ is cubic in $\boldsymbol{\theta}$, $\boldsymbol{\Sigma}$ is fixed, and $\boldsymbol{\theta}$ is distributed uniformly (see Appendix D for complete specification). The goal is to approximate the posterior $p(\boldsymbol{\theta}|\bar{\mathbf{x}}_o)$ from a small number of draws from the generative model (Fig. 2a). We parameterize $q(\mathbf{x}|\boldsymbol{\theta}; \boldsymbol{\phi})$ using a Gaussian distribution whose mean and precision are the output of a neural network with one hidden layer consisting of 10 tanh units.

We will compare our method to BOLFI (Bayesian Optimization for Likelihood-free Inference, Gutmann and Corander, 2016), an ABC method which – given a user-specified discrepancy measure – learns a $\mathcal{GP}$ that models the distribution of discrepancies between summary statistics of $\mathbf{x}$ and $\mathbf{x}_o$. Järvenpää et al. (2017) proposed multiple acquisition rules for BOLFI. The most principled (but also most costly) rule minimizes the expected integrated variance (*ExpIntVar*) of the approximate posterior after acquiring new data. BOLFI is a state-of-the-art method for simulation-efficient likelihood-free inference, and substantially

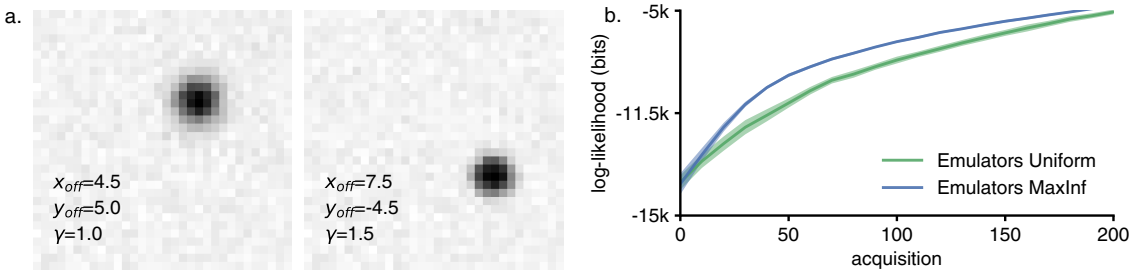

Figure 3: **Inferring location and contrast of a blob**. **a.** Two sample images from the generative model. Parameters are the spatial position and the contrast of the blob. **b.** Acquiring samples using the *MaxMI* rule yield to faster emulator learning than samples acquired uniformly in the parameter space. Performances are reported as log-likelihood of held-out test data. 200 acquisitions starting from $N_{\text{initial}} = 50$ initial points. Lines are means and SEMs from 20 runs.

more efficient than classical rejection-based methods such as rejection-ABC (Pritchard et al., 1999), MCMC-ABC (Marjoram et al., 2003), SMC-ABC (Sisson et al., 2007).

We use the total variation (TV) between true and approximate posterior (evaluated using numerical integration) to quantify performance as a function of the number of acquisitions. The emulator is trained on an initial dataset and updated after each new acquisition. We find that emulators with *MaxVar* sampling work better than uniform sampling (Fig. 2b). Both BOLFI rules (*ExpIntVar* and *MaxVar*) exhibit very similar performance, but require higher number of simulations than emulators to reach low TV values. On a 2-dimensional version of the problem, the qualitative ordering is the same, but the differences between methods are greater (Fig. 2c). We did additional runs of BOLFI *MaxVar* to confirm that it eventually converges towards the correct posterior. However, convergence is slow and the quality of the inferred posterior depends strongly on the choice of the threshold parameter used in BOLFI (see Appendix G).

ii) High-dimensional observations: Inferring the location and contrast of a blob

We show that our method can be applied to estimation problems with high-dimensional observations without having to resort to using summary statistics. We model the rendering of a blob on a 2D image, and learn a global emulator for the forward model.

The forward model takes as inputs three parameters (x_{off}, y_{off} and γ) – which encode horizontal and vertical displacement, and contrast of the blob – and returns per-pixels activation probabilities p_{ij}. The value of each pixel v_{ij} is then generated according to a binomial distribution with total count 255 (8-bits gray-scale image) and probability p_{ij}, resulting in a 32×32 pixel image (Fig. 3a). In this application, we use a multi-layer neural network whose output is, for each pixel, the mean parameter of the binomial distribution (see Appendix E for further details).

Using the *MaxMI* rule to acquire new test points in parameters space results in faster learning of the emulator, compared to uniform random acquisitions. Eventually, both rules converge towards the log-likelihood of the held-out test set, indicating successful global

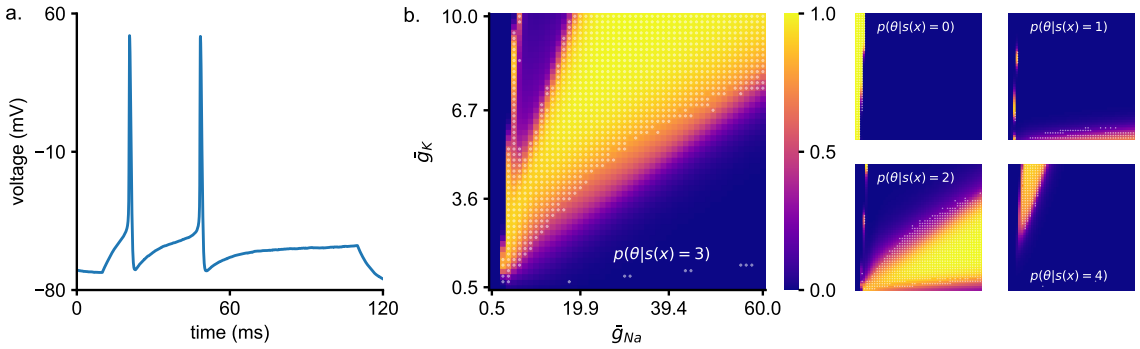

Figure 4: **Hodgkin-Huxley model**. **a.** Example trace from differential equations describing the model. **b.** Posterior inferred for number of spikes as a function of two biophysical parameters. Panels show posteriors for a given number of spikes. The largest panel shows the posterior given three spikes. As a posterior predictive check, we overlay white transparent markers on top of the posteriors where a simulation produced the given number of spikes (and no marker otherwise).

emulation of the forward model (Fig. 3b). We show posteriors distributions and samples in Appendix H. Since alternative approaches for likelihood-free inference (e.g. BOLFI) do not allow one to globally approximate a simulator, no performance benchmark against these methods was performed.

iii) Scientific application: Hodgkin-Huxley model

As an example of a scientific application, we use the Hodgkin-Huxley model (Hodgkin and Huxley, 1952) which describes the evolution of membrane potential in neurons (Fig. 4a). Fitting single- and multi-compartment Hodgkin-Huxley models to neurophysiological data is a central problem in neuroscience, and typically addressed using non-Bayesian approaches based on evolutionary optimization (Druckmann et al., 2007; Van Geit et al., 2016). In contrast to the previous examples, we do not model the raw data $\mathbf{x}$, but summary features derived from them. While this is often done out of necessity, calculating the posterior relative to summary statistics can be of scientific interest (Cornebise and Girolami, 2012). This is indeed the case when fitting biophysical models in neuroscience, which is typically performed with carefully chosen summary statistics representing properties of interest.

Here, we chose to model the number of action potentials (or spikes) in response to a step-current input, and we are interested in the set of parameters that are consistent with the observed number of action potentials. The conditional density of the emulator networks becomes a categorical distribution with 6 classes, modelling the probabilities of exactly 0, 1, ... 4 spikes, and 5 or more spikes (which never occurred under the parameter ranges we explored). Model parameters $\boldsymbol{\theta}$ are the ion-channel conductances $\bar{g}_{\mathrm{Na}}$ and $\bar{g}_{\mathrm{K}}$, controlling the shape and frequency of the spikes (further details in Appendix F).

We trained emulator networks using *MaxMI* to infer the posterior probabilities over $\boldsymbol{\theta}$ generating a given number of observed spikes – the acquisition surface is shown in Appendix I. Resulting posterior distributions are shown in Fig. 4b, along with a posterior predictive check showing that the mapping between parameters and summary features was learned correctly.

4. Discussion

We presented an approach for performing statistical inference on simulation-based models which do not permit tractable likelihood. We learn an 'emulator network', i.e. a probabilistic model that is consistent with the simulation, and for which likelihoods are tractable. The likelihoods of the emulator can then be plugged into any Bayesian inference approach (as in synthetic likelihood approaches Wood, 2010; Ong et al., 2016, 2018) to calculate the posterior. Active learning can be used to adaptively suggest new samples to reduce the number of calls to the simulator. We discussed two acquisition functions for learning 'local' and 'global' emulators, we showed that our approach scales to high-dimensional observation spaces, does not require user-defined distance functions or acceptance thresholds, and is not limited to Gaussian observations – all of which are challenging for conventional ABC approaches.

Our approach uses density estimation to approximate the likelihood. A complementary use of density-estimation in ABC is to directly target the posterior distribution (Papamakarios and Murray, 2017; Lueckmann et al., 2018; Le et al., 2017; Izbicki et al., 2018). This approach can be very useful – however, one advantage of likelihood-based approaches is that they allow one to apply the same synthetic likelihood to multiple priors (without having to retrain), or to pool information from multiple observations (by multiplying the corresponding synthetic likelihoods). More technically, posterior density estimation gives less flexibility in proposing samples – in order to yield the correct posterior, samples have to be drawn from the prior, or approaches such as importance-weighting (Lueckmann et al., 2018) or other post-hoc corrections (Papamakarios and Murray, 2017) have to be applied. We discuss additional related work published concurrently with this manuscript in Appendix C.

There are multiple ways in which our approach can be improved further: First, one could use alternative, and more expressive neural-network based density estimators, e.g. ones based on normalizing flows (Papamakarios et al., 2018). Second, one could use Bayesian posterior estimation (rather than ensembles) to capture parameter uncertainty, and/or use variational inference (rather than HMC) to derive an estimate of the posterior from the synthetic likelihood provided by the emulator. Third, we presented two acquisition functions (one for local and one for global estimation) – it is likely that the approach can be made more simulation-efficient by using different, and more sophisticated acquisition functions. In particular, our *MaxVar* rule targets the parameters with maximal uncertainty, but does not try to predict whether that uncertainty will be effectively reduced. However, evaluating acquisition functions like *ExpIntVar* can be computationally expensive – it will be useful to develop approaches which are sensitive to the relative cost of simulations and proposals, and adaptively adjust the acquisition function used.

Numerical simulations make it possible to model complex phenomena from first principles, and are indispensable tools in many fields in engineering and science. The advent of powerful approaches for statistical inference in simulation-based models (Brehmer et al., 2018) is opening up exciting opportunities for closing the gap between mechanistic, simulation-based and statistical approaches to modelling complex systems. Our Bayesian methodology based on emulators provides a fast, effective surrogate model for the intractable likelihood implied by the simulator, and the active-learning based rules lead to bounded-rational decisions about which simulations to run. In combination, they form a rigorous and resource-efficient basis for data analysis with simulators in the loop.

Acknowledgements

We thank Marcel Nonnenmacher and Pedro J. Gonçalves for discussions, and help with simulation of the Hodgkin-Huxley model. We thank David Greenberg and all members of the Neural Systems Analysis group for comments on the manuscript.

This work was supported by BMBF (FKZ 01IS18052 A-D, Project ADIMEM) and DFG (SFB 1089, SPP 2041, and SFB 1233, Project 'Robust Vision', 276693517) grants and by the caesar foundation.

References

M Beaumont, W Zhang, and D J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4), 2002.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015.

Johann Brehmer, Kyle Cranmer, Gilles Louppe, and Juan Pavez. Constraining Effective Field Theories with Machine Learning. *Physical Review Letters*, 121(11), 2018.

T Chen, E B Fox, and C Guestrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, 2014.

J Cornebise and M Girolami. Inference on summary statistics: a mean or an end? *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 74(3):419–474, 2012.

S Depeweg, J M Hernández-Lobato, F Doshi-Velez, and S Udluft. Decomposition of uncertainty for active learning and reliable reinforcement learning in stochastic systems. *arXiv:1710.07283*, 2017.

S Druckmann, Y Banitt, A Gidon, F Schürmann, H Markram, and I Segev. A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data. *Frontiers in Neuroscience*, 1, 2007.

Y Gal, R Islam, and Z Ghahramani. Deep bayesian active learning with image data. *arXiv:1703.02910*, 2017.

M U Gutmann and J Corander. Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models. *Journal of Machine Learning Research*, 17(125), 2016.

A L Hodgkin and A F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4), 1952.

N Houlsby, F Huszar, Z Ghahramani, and M Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

R Izbicki, A B Lee, and T Pospisil. Abc-cde: Towards approximate bayesian computation with complex high-dimensional data and limited simulations. *arXiv:1805.05480*, 2018.

M Järvenpää, M U Gutmann, A Pleska, Vehtari, A, and P Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *arXiv:1704.00520v2*, 2017.

G Karabatsos and F Leisen. An Approximate Likelihood Perspective on ABC Methods. *arXiv:1708.05341*, 2017.

M C Kennedy and A O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3), 2002.

B Lakshminarayanan, A Pritzel, and C Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, 2017.

T A Le, A G Baydin, R Zinkov, and F D Wood. Using synthetic data to train neural networks is model-based reasoning. *arXiv:1703.00868*, 2017.

C Louizos and M Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on International Conference on Machine Learning*, 2017.

JM Lueckmann, P J Goncalves, G Bassetto, K Öcal, M Nonnenmacher, and J H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems 30*, 2018.

P Marjoram, J Molitor, V Plagnol, and S Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26), 2003.

R M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

V M H Ong, D J Nott, M Tran, S A Sisson, and C C Drovandi. Variational bayes with synthetic likelihood. *arXiv:1608.03069*, 2016.

V MH Ong, D J Nott, and M S Smith. Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27(3), 2018.

A van den Oord, N Kalchbrenner, and K Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016.

G Papamakarios and I Murray. Fast epsilon-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, 2017.

G Papamakarios, T Pavlakou, and I Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, 2018.

L F Price, C C Drovandi, A Lee, and D J Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1), 2018.

J K Pritchard, M T Seielstad, A Perez-Lezaun, and M W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Mol Biol Evol*, 16(12), 1999.

S A Sisson, Y Fan, and M M Tanaka. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6), 2007.

W Van Geit, M Gevaert, G Chindemi, C Rössert, JD Courcol, E B Muller, F Schürmann, I Segev, and H Markram. Bluepyopt: Leveraging open source software and cloud infrastructure to optimise model parameters in neuroscience. *Frontiers in Neuroinformatics*, 10, 2016.

M Welling and Y W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011.

S N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466 (7310), 2010.

Appendix A.

Algorithm 1: ABC via active learning to learn a synthetic likelihood

Input : $p(\boldsymbol{\theta})$, $p(\mathbf{x}|\boldsymbol{\theta})$, $\mathbf{x}_o$ `// prior, stochastic simulator, observed data`
Output : $\hat{p}(\boldsymbol{\theta}|\mathbf{x}_o)$ `// approximate posterior`

$\mathcal{D} \leftarrow \mathcal{D}^{N_{\text{initial}}} = \left\{(\boldsymbol{\theta}_n, \mathbf{x}_n)\right\}_{n=1}^{N_{\text{initial}}} \sim p(\mathbf{x}, \boldsymbol{\theta})$ `//` $\boldsymbol{\theta}_n \sim p(\boldsymbol{\theta})$, $\mathbf{x}_n \sim p(\mathbf{x}|\boldsymbol{\theta}_n)$

do

  Train emulator $q(\mathbf{x}|\boldsymbol{\theta}; \boldsymbol{\phi})$ on $\mathcal{D}$

  Find $\boldsymbol{\theta}^*$ as the maximum of an acquisition function

  Acquire new data point $\mathcal{D}^* = \left\{(\boldsymbol{\theta}^*, \mathbf{x}^*)\right\}$ by simulating for $\boldsymbol{\theta}^*$ `//` $\mathbf{x}^*|\boldsymbol{\theta}^* \sim p(\mathbf{x}|\boldsymbol{\theta}^*)$

  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}^*$

while *not converged*

Find $\hat{p}(\boldsymbol{\theta}|\mathbf{x}_o)$ using the synthetic likelihood $\hat{\mathcal{L}}(\boldsymbol{\theta}) = q(\mathbf{x}_o|\boldsymbol{\theta}; \boldsymbol{\phi})$

Appendix B. Acquisition rule for global emulator learning

For global emulator learning, we use a rule based on information theory from the active learning literature that maximizes information gain (Houlsby et al., 2011; Gal et al., 2017; Depeweg et al., 2017). We refer to the rule

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} \mathbb{I}[\mathbf{x}, \boldsymbol{\phi}|\boldsymbol{\theta}, \mathcal{D}] \\ &= \arg\max_{\boldsymbol{\theta}} \underbrace{\mathbb{H}[\mathbf{x}|\boldsymbol{\theta}, \mathcal{D}]}_{\text{entropy}} - \underbrace{\mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\big[\mathbb{H}[\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\phi}]\big]}_{\text{expected conditional entropy}}\end{aligned} \tag{3}$$

as the maximum mutual information rule (*MaxMI*).

The first term is the entropy of the data under the posterior-predictive distribution implied by the emulator:

$$\mathbb{H}[\mathbf{x}|\boldsymbol{\theta}, \mathcal{D}] = -\int \hat{p}(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D}) \ln \hat{p}(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D}) \mathrm{d}\mathbf{x}, \tag{4}$$

where $\hat{p}(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D})$ is obtained by marginalizing out the emulator's parameters w.r.t. $p(\boldsymbol{\phi}|\mathcal{D})$:

$$\hat{p}(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D}) = \int q(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\mathcal{D}) \mathrm{d}\boldsymbol{\phi}. \tag{5}$$

The expected conditional entropy, $\mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\big[\mathbb{H}[\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\phi}]\big]$, is the average entropy of the output $\mathbf{x}$ for a particular choice of inputs $\boldsymbol{\theta}$ and emulator parameters $\boldsymbol{\phi}$, under the posterior distribution of emulator parameters $p(\boldsymbol{\phi}|\mathcal{D})$. Again, we treat ensemble members $\boldsymbol{\phi}_m$ as if they were draws from $p(\boldsymbol{\phi}|\mathcal{D})$. Houlsby et al. refer to this rule as Bayesian Active Learning by Disagreement (BALD): we query parameters $\boldsymbol{\theta}$ where the posterior predictive is very uncertain about the output (entropy is high), but the emulator, conditioned on the value of its parameters $\boldsymbol{\phi}$, is on average quite certain about the model output (conditional entropy low on average).

For many distributions closed-form expressions of $\mathbb{H}\big[\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\phi}\big]$ are available, but this is in general not true for the entropy of the marginal predictive distribution $\hat{p}(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D})$. To overcome this problem, we derived an upper-bound approximation to the entropy term based on the law of total variance: if we characterize the marginal distribution only in terms of its (co)variance $\Sigma_{\mathcal{D}}(\boldsymbol{\theta})$, then $\mathbb{H}[\mathbf{x}|\boldsymbol{\theta}, \mathcal{D}] \leq \frac{1}{2} \ln\big[(2\pi e)^N |(\Sigma_{\mathcal{D}}(\boldsymbol{\theta}))|\big]$. Using the law of total (co)variance, we get

$$\Sigma_{\mathcal{D}}(\boldsymbol{\theta}|\mathcal{D}) = \mathrm{Cov}[\mathbf{x}|\boldsymbol{\theta}] = \mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\big[\mathrm{Cov}[\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\phi}]\big] + \mathrm{Cov}_{\boldsymbol{\phi}|\mathcal{D}}\big[\mathbb{E}[\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\phi}]\big], \tag{6}$$

where all expectations can be approximated by samples drawn from $p(\boldsymbol{\phi}|\mathcal{D})$.

Note that the density of the forward model, $p(\mathbf{x}|\boldsymbol{\theta})$, does not appear in this rule. By using the upper-bound, we can use gradient-based optimization to find $\boldsymbol{\theta}^*$. Alternatively, entropies could be approximated using sample, which, however, would be slower.

References

S Depeweg, J M Hernández-Lobato, F Doshi-Velez, and S Udluft. Decomposition of uncertainty for active learning and reliable reinforcement learning in stochastic systems. *arXiv:1710.07283*, 2017.

Y Gal, R Islam, and Z Ghahramani. Deep bayesian active learning with image data. *arXiv:1703.02910*, 2017.

N Houlsby, F Huszar, Z Ghahramani, and M Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

Appendix C. Additional related work

Papamakarios et al. (2018), concurrently and independently to our approach (Lueckmann et al., 2018, an earlier preprint version of this work), proposed learning synthetic likelihoods using neural density estimators for likelihood-free inference: They use Masked Autoregressive Flows as synthetic likelihoods and report state-of-the-art performance compared to methods that directly target the posterior. Like our approach, the density estimator is trained on sequentially chosen simulations. Rather than using acquisition functions that take into account uncertainty to guide sampling, they draw samples from the current estimate of the posterior. Their approach corresponds to an alternative way of learning a local emulator.

The recent workshop paper of Durkan et al. (2018) compares Papamakarios et al. (2018) and our approach on three toy problems learning local emulators. On these toy-problems, both methods are similarly efficient (and more efficient than methods directly targeting the posterior), however, the wallclock time of our method is substantially higher, because of the additional cost of evaluating the acquisition function. Whether this additional cost is warranted on a given problem will depend both on any additional gain brought about by the active selection of samples, as well as the cost of the simulator. For expensive simulation costs, additional computational budget should be spent to carefully decide for which parameters to simulate.

References

C Durkan, G Papamakarios, and I Murray. Sequential neural methods for likelihood-free inference. *arXiv:1811.08723*, 2018.

JM Lueckmann, G Bassetto, T Karaletsos, and J H Macke. Likelihood-free inference with emulator networks. *arXiv:1805.09294v1*, 2018.

G Papamakarios, D C Sterratt, and I Murray. Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows. *arXiv:1805.07226*, 2018.

Appendix D. Gaussian simulator example

D.1. Model

Data is generated independently according to $\mathbf{x}_i \sim \mathcal{N}(\cdot|f(\boldsymbol{\theta}), \boldsymbol{\Sigma})$, $i = 1 \dots n$, where $n = 10$, $f(\boldsymbol{\theta}) = (1.5\ \boldsymbol{\theta} + 0.5)^3/200$, $\boldsymbol{\Sigma}_{ii} = 0.1$, $\boldsymbol{\Sigma}_{ij} = 0$ for $i \neq j$, $\bar{\mathbf{x}}_o = \frac{1}{n}\sum_i^n \bar{\mathbf{x}}_o^{(i)} = \mathbf{2}$, and $\boldsymbol{\theta}$ is distributed uniformly in $[-8, 8]^p$ where p is the dimensionality of the problem.

This problem is inspired by the Gaussian example studied in Järvenpää et al. (2017), where f was chosen as $f(\boldsymbol{\theta}) = \boldsymbol{\theta}$. We introduce a nonlinearity in f, since our method with uniform acquisitions would otherwise trivially generalize across the space – we observed that a neural network with the right amount of ReLu units can learn the linear mapping perfectly, independently of where the training samples are acquired.

D.2. Evaluation

We evaluate our method and BOLFI (Järvenpää et al., 2017) on this problem in 1D and 2D. In 1D, algorithms start with $N_{\text{initial}} = 10$ initial samples, in 2D with $N_{\text{initial}} = 25$, and make 100 acquisitions after each of which we evaluate how well the ground truth posterior is recovered.

As performance metric, we calculate total variation (TV) between $\hat{p}(\boldsymbol{\theta}|\mathbf{x}_o)$ and $p(\boldsymbol{\theta}|\mathbf{x}_o)$, defined as

$$\frac{1}{2} \int \left| \hat{p}(\boldsymbol{\theta}|\mathbf{x}_o) - p(\boldsymbol{\theta}|\mathbf{x}_o) \right| \mathrm{d}\boldsymbol{\theta}.$$

D.3. Network architecture and training

Emulator networks model a normal distribution as output, so that the outputs of the network parametrise mean and covariance (Cholesky factor of the covariance matrix). Neural networks have one hidden layer consisting of 10 tanh units. We train an ensemble of $M = 50$ networks using Adam (Kingma and Ba, 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) for SGD, and a learning rate of 0.01.

D.4. BOLFI

BOLFI requires choice of a distance function: We use the the Mahalanobis distance

$$\Delta_{\boldsymbol{\theta}} = \left((\bar{\mathbf{x}} - \bar{\mathbf{x}}_o)^T \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}} - \bar{\mathbf{x}}_o) \right)^{1/2},$$

in line with the distance function used for the Gaussian example studied in Järvenpää et al. (2017). We use the implementation provided by the authors (Lintusaari et al., 2017).

References

M Järvenpää, M U Gutmann, A Pleska, Vehtari, A, and P Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *arXiv:1704.00520v2*, 2017.

D P Kingma and J Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980*, 2014.

J Lintusaari, H Vuollekoski, A Kangasrääsiö, K Skytén, M Järvenpää, M Gutmann, A Vehtari, J Corander, and S Kaski. Elfi: Engine for likelihood free inference. *arXiv:1708.00707*, 2017.

Appendix E. Image example

E.1. Model

Images are generated according to:

$$
\begin{aligned}
I_{xy} &\sim \mathrm{Bin}(\cdot|255, p_{xy}) \\
p_{xy} &= 0.9 - 0.8 \exp^{-0.5\left(r_{xy}/\sigma^2\right)^\gamma} \\
r_{xy} &= (x - x_{\mathrm{off}})^2 + (y - y_{\mathrm{off}})^2,
\end{aligned}
$$

where x and y are coordinates in the image, and $\mathrm{Bin}(\cdot|n, p)$ is the binomial distribution.

Model parameters are x_{off} and y_{off}, which respectively determine the horizontal and the vertical offset of the blob, γ, defining its contrast, and σ^2, determining the width.

For our experiments, we use images of size 32×32 pixels. We choose uniform priors in the range $[-16, 16]$ for x_{off} and y_{off}, and a uniform prior in the range $[0.25, 5]$ for γ. We fix σ to 2.

E.2. Evaluation

We evaluate different acquisition methods by keeping track of the log-likelihood of a test set consisting of 5000 parameters-image pairs over the course of acquisitions (starting from an initial sample of size $N_{\mathrm{initial}} = 50$).

E.3. Network architecture and training

Emulator networks model a binomial distribution as output. Neural networks have two hidden layers (200 units each) with ReLu activation functions. We train an ensemble of $M = 25$ networks using Adam (Kingma and Ba, 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) for SGD with a learning rate of 0.001.

References

D P Kingma and J Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980*, 2014.

Appendix F. Hodgkin-Huxley example

F.1. Model

The dynamic equations describing the evolution of the membrane potential and of the gating variables of the neuron are taken from Pospischil et al. (2008):

$$\begin{aligned} C_m\dot{V} &= -(I_{\text{leak}} + I_{\text{Na}} + I_{\text{K}} + I_{\text{M}} + I_{\text{ext}}) \\ &= g_{\text{leak}}(E_{\text{leak}} - V) + \bar{g}_{\text{Na}} m^3 h (E_{\text{Na}} - V) + \\ &+ \bar{g}_{\text{K}} n^4 (E_{\text{K}} - V) + \bar{g}_{\text{M}} p (E_{\text{K}} - V) + I_{\text{in}}(t), \end{aligned}$$

where C_m is membrane capacitance, V the membrane potential, I_c are ionic currents ($c = \{\text{Na}, \text{K}, \text{M}\}$) and $I_{\text{in}}(t)$ is an externally applied current which we can imagine as the sum of a static bias I_{bias} and a time-varying zero-mean noise signal $\varepsilon(t)$. I_{Na} and I_{K} shape the up- and down-stroke phases of the action potential (spike), I_{M} is responsible for spike-frequency adaptation, and I_{leak} is a leak current describing the passive properties of the cell membrane. Each current is in turn expressed as the product of a maximum conductance ($\bar{g}_c$) and the voltage difference between the membrane potential and the reversal potential for that current(E_c), possibly modulated by zero or more 'gating' variables (m, h, n, p).

Each $x \in \{m, h, n, p\}$ evolves according to first order kinetics in the form:

$$\dot{x} = \frac{1}{\tau_x(V)}\big(x_\infty(V) - x\big)$$

We provide a step current as input.

In our example application, free model parameters are $\bar{g}_{\text{Na}}$ and $\bar{g}_{\text{K}}$. We model uniform priors over these parameters: $\bar{g}_{\text{Na}}$ is between 0.5 and 60 and $\bar{g}_{\text{K}}$ is between 0.5 and 10.

F.2. Evaluation

We evaluate the posterior obtained through the emulator after $t = 250$ acquisitions, starting from an initial sample size $N_{\text{initial}} = 30$. As posterior predictive check, we span a grid over the parameter space and compare simulator outputs to the posterior.

F.3. Network architecture and training

Emulator networks model a categorical distribution with $K = 6$ classes as output. Neural networks have two hidden layer (200 units each) with a ReLu activation functions. We train an ensemble of $M = 25$ networks using Adam (Kingma and Ba, 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) for SGD with a learning rate of 0.001.

References

D P Kingma and J Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980*, 2014.

M Pospischil, M Toledo-Rodriguez, C Monier, Z Piwkowska, T Bal, Y Frégnac, H Markram, and A Destexhe. Minimal hodgkin-huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, 99(4-5), 2008.

Appendix G. BOLFI convergence

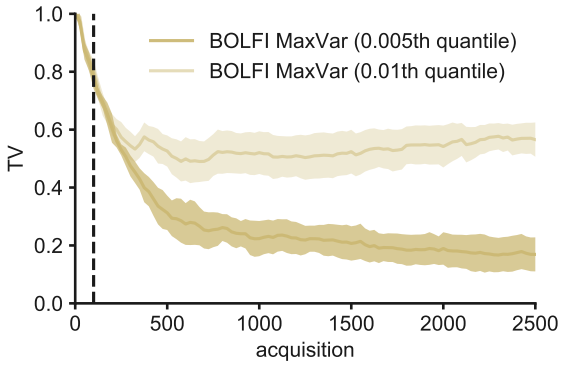

Figure 5: **Convergence of BOLFI *MaxVar*.** In the manuscript, we show performance up to 100 acquisitions (indicated by the dotted line). With additional acquisitions, BOLFI converges. The quality of the inferred posterior strongly depends on the value of the threshold hyperparameter used in BOLFI.

Appendix H. Posteriors and samples for image example

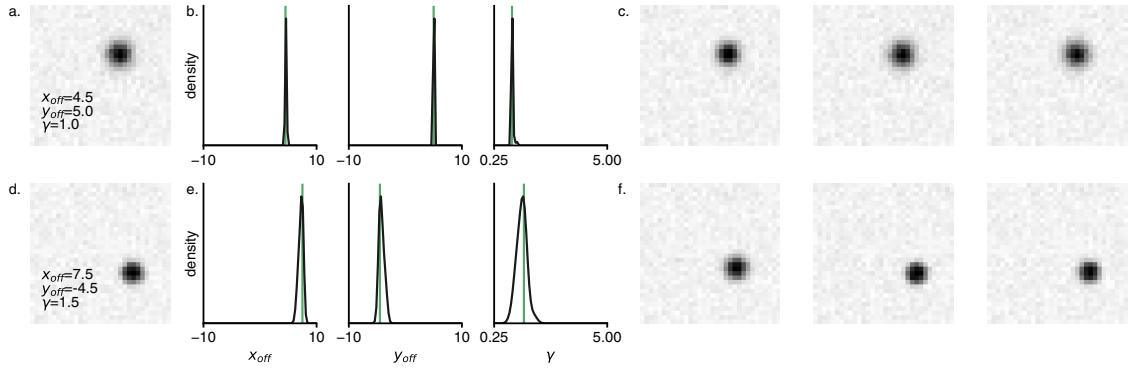

Figure 6: **Posteriors and samples for image example. a.** Observed image. **b.** Inferred posteriors. **c.** Posterior samples. **d.** Another observation, with posteriors in **e.** and samples in **f.**

Appendix I. *MaxMI* acquisition for Hodgkin-Huxley model

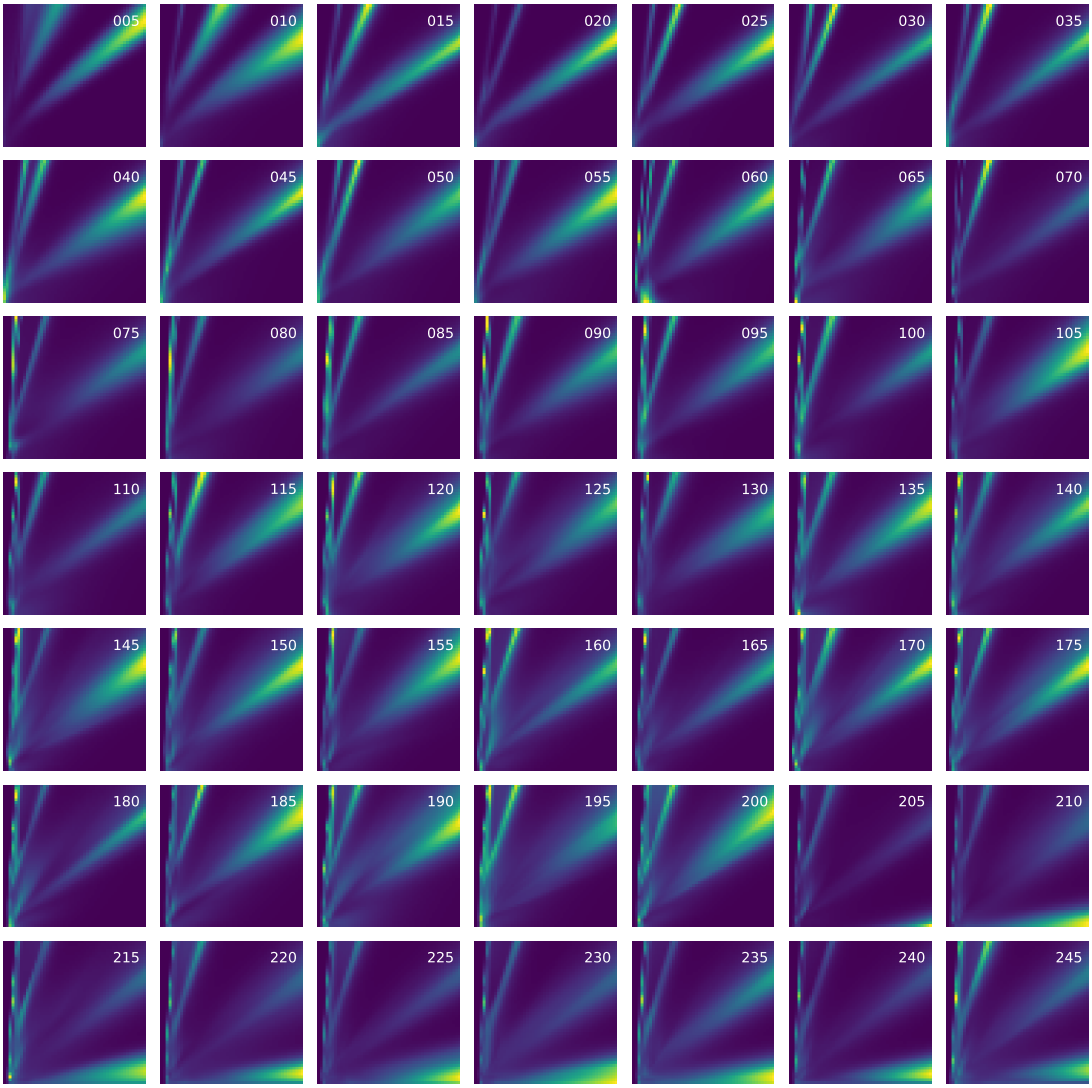

Figure 7: **Acquistion surface for *MaxMI* rule on Hodgkin-Huxley example.** Individual panels show the acquisition surface over $\boldsymbol{\theta}$ as additional samples have been acquired. The acquisition rule proposes datapoints at the decision boundaries of the posterior.