Leveraging Global Parameters for Flow-based Neural Posterior Estimation

Pedro Luiz Coelho Rodrigues, Thomas Moreau, Gilles Louppe, Alexandre Gramfort

Leveraging Global Parameters for Flow-based Neural Posterior Estimation

Pedro L. C. Rodrigues [1] **Thomas Moreau** [1] **Gilles Louppe** [2] **Alexandre Gramfort** [1]

Abstract

Inferring the parameters of a stochastic model based on experimental observations is central to the scientific method. A particularly challenging setting is when the model is strongly indeterminate, i.e., when distinct sets of parameters yield identical observations. This arises in many practical situations, such as when inferring the distance and power of a radio source (is the source close and weak or far and strong?) or when estimating the amplifier gain and underlying brain activity of an electrophysiological experiment. In this work, we present a method for cracking such indeterminacy by exploiting additional information conveyed by an auxiliary set of observations sharing global parameters. Our method extends recent developments in simulation-based inference (SBI) based on normalizing flows to Bayesian hierarchical models. We validate quantitatively our proposal on a motivating example amenable to analytical solutions, and then apply it to invert a well known non-linear model from computational neuroscience.

1. Introduction

Simulation-based inference (SBI) has the potential to revolutionize experimental science as it opens the door to the inversion of arbitrary complex non-linear computer models, such as those found physics, biology or neuroscience (Cranmer et al., 2020). The only requirement is to have access to a simulator. Grounded in Bayesian statistics, recent SBI techniques leverage the deep learning advances to approximate the posterior distributions over the full simulator parameters, hence allowing to quantify uncertainties and therefore to reveal whether certain parameters are not worth scientific interpretation given some observation.

SBI is concerned with the estimation of a conditional distribution over parameters of interest $\boldsymbol{\theta}$. Given some observation x_0, the goal is to compute the posterior $p(\boldsymbol{\theta}|x_0)$. It generally happens that some of these parameters are strongly coupled, leading to very structured posteriors with low dimensional sets of equally likely parameters values. For example, this happens when the data generative process depends only on the products of some parameters: multiplying one of such parameters by a constant and another by its inverse will not affect the output. Performing Bayesian inference on such models naturally leads to a "ridge" or "banana shape" in the posterior landscape, as seen e.g. in Figure 4 of Gonçalves et al. (2020). More formally the present challenge is posed as soon as the model likelihood function is non-injective w.r.t. $\boldsymbol{\theta}$, and is not due to the presence of some random perturbations of the output.

To alleviate the ill-posedness of the estimation problem, one may consider a hierarchical Bayesian model (Gelman & Hill, 2007) where certain parameters are shared among different observations. In other words, the model's parameters $\boldsymbol{\theta}_i$ for an observation $\boldsymbol{x}_i$ are partitioned into $\boldsymbol{\theta}_i = \{\boldsymbol{\alpha}_i, \boldsymbol{\beta}\}$, where $\boldsymbol{\alpha}_i$ is a set of sample specific (or local) parameters, and $\boldsymbol{\beta}$ corresponds to shared (or global) parameters. For this broad class of hierarchical models, the posterior distribution for a set $\mathcal{X} = \{x_1, \dots, x_N\}$ of N observations can be written as (Tran et al., 2017):

$$p(\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N, \boldsymbol{\beta}|\mathcal{X}) \propto p(\boldsymbol{\beta}) \prod_{i=1}^{N} p(x_i|\boldsymbol{\alpha}_i, \boldsymbol{\beta}) p(\boldsymbol{\alpha}_i|\boldsymbol{\beta}).$$

Hierarchical models share statistical strength across observations, hence resulting in sharper posteriors and more reliable estimates of the (global and local) parameters and their uncertainty. Examples of applications of hierarchical models are topic models (Blei et al., 2003), matrix factorization algorithms (Salakhutdinov et al., 2013), including Bayesian non-parametrics strategies (Teh & Jordan, 2010).

In this work, we further assume that the likelihood function $p(x_i|\boldsymbol{\alpha}_i, \boldsymbol{\beta})$ is implicit and intractable, leading to so-called likelihood-free inference (LFI) problems. Several Bayesian LFI algorithms (Papamakarios et al., 2019b; Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019; Hermans et al., 2020; Durkan et al., 2020b) have recently been developed to carry out inference in this scenario. These methods all operate by learning parts of the Bayes' rule, such as the likelihood function, the likelihood-to-evidence ratio, or the posterior itself. Approaches for

[1]Inria, Université Paris-Saclay, France [2]University of Liège, Belgium. Correspondence to: Pedro L. C. Rodrigues <pedro.rodrigues@melix.org>.

Preliminary work. Under review.

LFI in hierarchical models exist, but are limited. Tran et al. (2017) adapt variational inference to hierarchical implicit models, while Brehmer et al. (2019) and Hermans et al. (2020) approach this problem using amortized likelihood ratios. Here, motivated by the posterior estimates of individual samples, we consider a sequential neural posterior estimation approach derived from SNPE-C (Greenberg et al., 2019).

The paper is organized as follows. First, we formalize our estimation problem by introducing the notion of global and local parameters, and instantiate it on a motivating example amenable to analytic posterior estimates allowing for quantitative evaluation. Then, we propose a neural posterior estimation technique based on a pair of normalizing flows and a *deepset* architecture (Zaheer et al., 2017) for conditioning on the set $\mathcal{X}$ of observations sharing the global parameters. Results on an application with time series produced by a non-linear model from computational neuroscience (Ableidinger et al., 2017) demonstrate the gain in statistical power of our approach thanks to the use of auxiliary observations.

2. Hierarchical models with global parameters

2.1. Motivating example

Consider a stochastic model with two parameters, α and β, that generates as output $x = \alpha\beta + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. We assume that both parameters have an uniform prior distribution $\alpha, \beta \sim \mathcal{U}[0, 1]$ and that σ is known and small. Our goal is to obtain the posterior distribution of (α, β) for a given observation $x_0 = \alpha_0\beta_0 + \varepsilon$. This simple example describes common situations where indeterminacy emerges. For instance, x_0 could be the radiation power measured by a sensor, α the intensity of the emitting source, and β the inverse squared distance of the sensor to the source. In this case, a given measurement may have been due to either close weak sources ($\alpha \downarrow$ and $\beta \uparrow$) or far strong ones ($\alpha \uparrow$ and $\beta \downarrow$).

Using Bayes' rule we have that

$$p(\alpha, \beta|x_0) \propto p(x_0|\alpha, \beta)p(\alpha, \beta) \ ,$$

and considering σ small we can write (see Appendix A of the supplementary materials for more details)

$$p(\alpha, \beta|x_0) \approx \frac{e^{-(x_0-\alpha\beta)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \ \frac{\mathbf{1}_{[0,1]}(\alpha)\mathbf{1}_{[0,1]}(\beta)}{\log(1/x_0)} \ , \quad (1)$$

where $\mathbf{1}_{[a,b]}(x)$ is an indicator function that equals one for $x \in [a, b]$ and zero elsewhere. Note that the first term in the product converges to $\delta(x_0 - \alpha\beta)$ as $\sigma \to 0$ and that the joint posterior distribution has an infinite number of pairs (α, β) with the same probability, revealing the parameter indeterminacy of this example. Indeed, for $x \in [0, 1]$ and $\beta \in [x, 1]$, all pair of parameters $(\frac{x}{\beta}, \beta)$ yield the same observations and the likelihood function $p(\cdot|\frac{x}{\beta}, \beta)$ is constant. Thus, the posterior distribution has level sets with a ridge or "banana shape" along these solutions. The top row of Figure 1 on Figure 1 portrays the joint and the marginal posterior distributions when $(\alpha_0, \beta_0) = (0.5, 0.5)$ and $\sigma = 0$.

2.2. Exploiting the additional information in $\mathcal{X}$

Our motivating example illustrates a situation where two parameters are related in such a way that one may not be known without the other. In practice, however, it is possible that one of these parameters is shared with other observations. For instance, this is the case when a single source of radiation is measured with multiple sensors located at different unknown distances. The power of the source is fixed across multiple measurements and its posterior can be better inferred by aggregating the information from all sensors. Our goal in this section is to formalize such setting so as to leverage this additional information and obtain a posterior distribution that 'breaks' parameter indeterminacy. Note that the root cause of the statistical challenge here is not the presence of noise, but rather the intrinsic structure of the observation model.

To tackle the inverse problem of determining the posterior distribution of parameters $(\boldsymbol{\alpha}_0, \boldsymbol{\beta})$ given an observation x_0 of a stochastic model, we consider the following scenario. We assume that the model's structure is such that $\boldsymbol{\alpha}_0$ is a parameter specific to each observation (local), while $\boldsymbol{\beta}$ is shared among different observations (global). Yet both are unknown. We consider having access to a set $\mathcal{X} = \{x_1, \ldots, x_N\}$ of additional observations generated with the same $\boldsymbol{\beta}$ as x_0.

Taking the model's hierarchical structure into account we use Bayes' rule to write

$$\begin{aligned} p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0, \mathcal{X}) &= p(\boldsymbol{\alpha}_0|\boldsymbol{\beta}, x_0, \mathcal{X})p(\boldsymbol{\beta}|x_0, \mathcal{X}) \\ &\propto p(\boldsymbol{\alpha}_0|\boldsymbol{\beta}, x_0)p(x_0, \mathcal{X}|\boldsymbol{\beta})p(\boldsymbol{\beta}) \\ &\propto p(\boldsymbol{\alpha}_0|\boldsymbol{\beta}, x_0) \ \textstyle\prod_{i=0}^{N} p(x_i|\boldsymbol{\beta})p(\boldsymbol{\beta}) \\ &\propto p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0) \ \textstyle\prod_{i=1}^{N} p(\boldsymbol{\beta}|x_i)p(\boldsymbol{\beta}) \end{aligned} \quad (2)$$

which shows how the initial posterior distribution $p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0)$ is modified by additional observations from $\mathcal{X}$ sharing the same $\boldsymbol{\beta}$ as x_0. In Section 3, we present a strategy for approximating such posterior distribution when the likelihood function of the stochastic model of interest is intractable and, therefore, the posterior distributions $p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0)$ and $p(\boldsymbol{\beta}|x_i)$ have to be approximated with conditional density estimators.

2.3. Motivating example with multiple observations

We now detail the effect of $\mathcal{X}$ on the posterior distribution of our motivating example. The $N+1$ observations in $\{x_0\} \cup \mathcal{X}$ are such that $x_i = \alpha_i \beta_0 + \varepsilon$ for $i = 0, \dots, N$ with $\alpha_i \sim \mathcal{U}[0,1]$ drawn from the same prior. The posterior distribution may be written as (see Appendix A of the supplementary materials for more details)

$$p(\alpha, \beta | x_0, \mathcal{X}) \approx p(\alpha, \beta | x_0) \frac{\mathbf{1}_{[\mu,1]}(\beta)}{\beta^N} \frac{N \log(1/x_0)}{(1/\mu^N - 1)} \quad , \quad (3)$$

where $\mu = \max(\{x_0\} \cup \mathcal{X})$. This expression shows how the initial full posterior distribution (1) changes with the extra information conveyed by $\mathcal{X}$. It can be also shown that as $N \to 0$ (no additional observations) the posterior distribution converges back to $p(\alpha, \beta | x_0)$. Besides, each observation x_i gives extra information on β as the marginal posterior probability

$$p(\beta | x_i) = \frac{1}{\log(1/x_i)} \frac{\mathbf{1}_{[x_i,1]}(\beta)}{\beta}$$

is supported on $[x_i, 1]$ (see Appendix A for the details). One can verify that μ converges geometrically towards β_0 as

$$P(\mu < \beta_0(1-\epsilon)) = \prod_{i=1}^{N} P(\alpha_i < (1-\epsilon)) = (1-\epsilon)^N \quad .$$

Therefore, the posterior distribution gets sharper around the ground truth as N increases. Figure 1 portrays the joint and marginal posterior distributions with $N = 10$ and $N = 100$.

3. Neural posterior estimation on hierarchical models with global parameters

When the likelihood function of the stochastic model is intractable, MCMC methods commonly used for posterior estimation are not applicable, since they depend on the evaluation of likelihood ratios, which are not available analytically nor numerically. To bypass such difficulty, we employ tools from likelihood-free inference (LFI) to directly estimate an approximation to the posterior distribution using a conditional neural density estimator trained over simulations of the model. In what follows, we present a neural network architecture for approximating the posterior distribution of a hierarchical model with global parameters based on normalizing flows. We also describe the training procedure for learning the parameters of the network using a multi-round procedure known as sequential neural posterior estimation or SNPE-C (Greenberg et al., 2019).

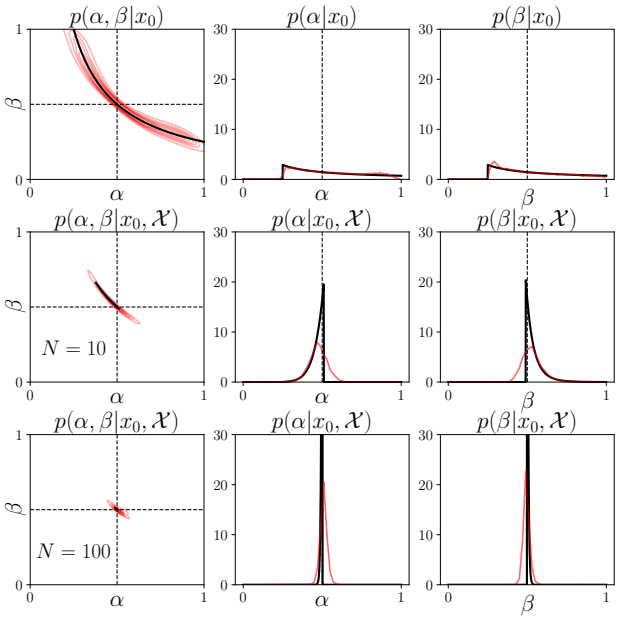

Figure 1. Plots of the analytic (black) and approximated (red) posterior distributions for the motivating example from Section 2. The ground truth values α_0 and β_0 which generate x_0 are indicated with dashed lines. Approximations are obtained using the strategy described in Section 3 with $n = 10^4$ simulations from the model. We observe that adding the $N = 10$ and $N = 100$ observations from $\mathcal{X}$ to the posterior distribution significantly reduces uncertainty over the estimates of α_0 and β_0.

3.1. Approximating the posterior distribution with two normalizing flows

We approximate $p(\boldsymbol{\alpha}_0, \boldsymbol{\beta} | x_0, \mathcal{X})$ based on its factorization (2) as follows:

$$\begin{aligned} p(\boldsymbol{\beta} | x_0, \mathcal{X}) &\approx q_{\boldsymbol{\phi}_1}(\boldsymbol{\beta} | x_0, f_{\boldsymbol{\phi}_3}(\mathcal{X})) \\ p(\boldsymbol{\alpha}_0 | \boldsymbol{\beta}, x_0) &\approx q_{\boldsymbol{\phi}_2}(\boldsymbol{\alpha}_0 | \boldsymbol{\beta}, x_0) \end{aligned} \quad (4)$$

where $q_{\boldsymbol{\phi}_1}$ and $q_{\boldsymbol{\phi}_2}$ are normalizing flows, i.e., invertible neural networks capable of transforming data points sampled from a simple distribution, e.g. Gaussian, to approximate any probability density function (Papamakarios et al., 2019a). The function $f_{\boldsymbol{\phi}_3}$ is a *deepset* neural network (Zaheer et al., 2017) structured as

$$f_{\boldsymbol{\phi}_3}(\mathcal{X}) = g_{\boldsymbol{\phi}_3^{(1)}} \left(\frac{1}{N} \sum_{i=1}^{N} h_{\boldsymbol{\phi}_3^{(2)}}(x_i) \right) , \quad (5)$$

where h is a neural network parametrized by $\boldsymbol{\phi}_3^{(1)}$ that generates a new representation for the data points in $\mathcal{X}$ and g is a network parametrized by $\boldsymbol{\phi}_3^{(2)}$ that processes the average value of the embeddings. Note that this aggregation step is crucial for imposing the invariance to ordering of the neural

network. It would also be possible to choose other permutation invariant operations, such as the maximum value of the set or the sum of its elements, but we have observed more stable performance on our experiments when aggregating the observations by their average. It is possible to show that f_{ϕ_3} is an universal approximator invariant to the ordering of its inputs (Zaheer et al., 2017). Such property is important for our setting because the ordering of the extra observations in $\mathcal{X}$ should not influence our approximation of the posterior distribution. In what follows, we refer to our approximation either by its factors q_{ϕ_1} and q_{ϕ_2} or by $q_{\boldsymbol{\phi}}$ with $\boldsymbol{\phi} = \{\phi_1, \phi_2, \phi_3\}$.

3.2. Estimating ϕ

We estimate the parameters $\boldsymbol{\phi}$ by minimizing the average Kullback-Leibler divergence between the posterior distribution $p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0, \mathcal{X})$ and our approximation $q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0, \mathcal{X})$ for different values of x_0 and $\mathcal{X}$:

$$\min_{\boldsymbol{\phi}} \; \mathbb{E}_{p(x_0,\mathcal{X})}\Big[\mathrm{KL}(p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0, \mathcal{X}) \| q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0, \mathcal{X}))\Big] \;,$$

where $\mathrm{KL}(p\|q_{\boldsymbol{\phi}}) = 0$ if, and only if, $p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0, \mathcal{X}) = q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|x_0, \mathcal{X})$. We may rewrite the optimization problem in terms of each of its parameters to get

$$\min_{\phi_1,\phi_2,\phi_3} \quad \mathcal{L}(\phi_1, \phi_2, \phi_3) \;, \tag{6}$$

where $\mathcal{L} = \mathcal{L}_{\boldsymbol{\alpha}} + \mathcal{L}_{\boldsymbol{\beta}}$ with

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\alpha}} &= -\mathbb{E}_{p(x_0,\mathcal{X})}\mathbb{E}_{p(\boldsymbol{\alpha}_0,\boldsymbol{\beta}|x_0,\mathcal{X})}\left[\log(q_{\boldsymbol{\phi}_2}(\boldsymbol{\alpha}_0|\boldsymbol{\beta}, x_0)\right] \;, \\ &= -\mathbb{E}_{p(x_0,\mathcal{X},\boldsymbol{\alpha}_0,\boldsymbol{\beta})}\left[\log(q_{\boldsymbol{\phi}_2}(\boldsymbol{\alpha}_0|\boldsymbol{\beta}, x_0)\right] \;,\end{aligned}$$

and

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\beta}} &= -\mathbb{E}_{p(x_0,\mathcal{X})}\mathbb{E}_{p(\boldsymbol{\alpha}_0,\boldsymbol{\beta}|x_0,\mathcal{X})}\left[\log(q_{\boldsymbol{\phi}_1}(\boldsymbol{\beta}|x_0, f_{\boldsymbol{\phi}_3}(\mathcal{X})))\right] \\ &= -\mathbb{E}_{p(x_0,\mathcal{X},\boldsymbol{\alpha}_0,\boldsymbol{\beta})}\left[\log(q_{\boldsymbol{\phi}_1}(\boldsymbol{\beta}|x_0, f_{\boldsymbol{\phi}_3}(\mathcal{X})))\right] \;.\end{aligned}$$

3.3. Training from simulated data

In practice, we minimize the objective function in (6) using a Monte-Carlo approximation with data points generated using the factorization

$$p(x_0, \mathcal{X}, \boldsymbol{\alpha}_0, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{i=0}^{N} p(x_i|\boldsymbol{\alpha}_i, \boldsymbol{\beta}) p(\boldsymbol{\alpha}_i|\boldsymbol{\beta}) \;, \tag{7}$$

where $p(\boldsymbol{\alpha}_i, \boldsymbol{\beta}) = p(\boldsymbol{\alpha}_i|\boldsymbol{\beta})p(\boldsymbol{\beta})$ is a prior distribution describing our initial knowledge about the parameters, and $p(x_i|\boldsymbol{\alpha}_i, \boldsymbol{\beta})$ is related to the stochastic output of the simulator for a given pair of parameters $(\boldsymbol{\alpha}_i, \boldsymbol{\beta})$. More concretely, the training dataset is generated as follows:

1. Sample a set of parameters from the prior distribution such that $(\boldsymbol{\alpha}_i^j, \boldsymbol{\beta}^j) \sim p(\boldsymbol{\alpha}_i, \boldsymbol{\beta})$ with $j = 1, \ldots, n$ and $i = 0, \ldots, N$.
2. For each (i, j)-pair, generate an observation from the stochastic simulator $x_i^j \sim p(x|\boldsymbol{\alpha}_i^j, \boldsymbol{\beta}^j)$ so that each observation x_0^j is accompanied by its corresponding N extra observations $\mathcal{X}^j = \{x_1^j, \ldots, x_N^j\}$.

The losses $\mathcal{L}_{\boldsymbol{\alpha}}$ and $\mathcal{L}_{\boldsymbol{\beta}}$ are then approximated by

$$\mathcal{L}_{\boldsymbol{\alpha}}^n = -\frac{1}{n}\sum_{j=1}^{n} \log(q_{\boldsymbol{\phi}_2}(\boldsymbol{\alpha}_0^j|\boldsymbol{\beta}^j, x_0^j)) \tag{8}$$

and

$$\mathcal{L}_{\boldsymbol{\beta}}^n = -\frac{1}{n}\sum_{j=1}^{n} \log(q_{\boldsymbol{\phi}_1}(\boldsymbol{\beta}^j|x_0^j, f_{\boldsymbol{\phi}_3}(\mathcal{X}^j))) \;. \tag{9}$$

3.4. Refining the approximation with multiple rounds

The optimization strategy described above yields a set of parameters $\boldsymbol{\phi}$ for which the KL divergence between the true posterior distribution p and the approximation $q_{\boldsymbol{\phi}}$ is minimized, on average, for all possible values of x_0 and $\mathcal{X}$. This is sometimes called amortization, since the posterior distribution is expected to be well approximated for every possible observation as the number of simulations goes to infinity. However, it might be useful in some cases to focus the capacity of $q_{\boldsymbol{\phi}}$ to better estimate the posterior distribution for a specific choice of $\tilde{x}_0$ and $\tilde{\mathcal{X}}$. This is relevant, for instance, when the observed data is scarce and/or difficult to obtain or simulations of the model are costly.

We target the approximation $q_{\boldsymbol{\phi}}$ to $\tilde{x}_0$ and $\tilde{\mathcal{X}}$ using an adaptation to the sequential neural posterior estimation described in Greenberg et al. (2019), also known as SNPE-C. This algorithm uses a multiround strategy in which the data points used for minimizing the loss function $\mathcal{L}$ and obtaining parameters $\boldsymbol{\phi}^{(r)}$ at round r are obtained from simulations with $\boldsymbol{\alpha}_0, \boldsymbol{\beta} \sim q_{\boldsymbol{\phi}^{(r-1)}}(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|\tilde{x}_0, \tilde{\mathcal{X}})$. At round $r = 0$, parameters $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}$ are generated from their prior distributions, which boils down to the procedure described in Section 3.3. Note that an important point is that for the different rounds, the extra observations $\mathcal{X}$ should be simulated with the parameters $\boldsymbol{\alpha}_i^j$ drawn from the original prior distribution $p(\boldsymbol{\alpha}_i|\boldsymbol{\beta})$, since the posterior distribution returned by the multi-round procedure is only targeted for observation $\tilde{x}_0$. We refer the reader to Greenberg et al. (2019) for further details on the usual SNPE-C procedure, notably a proof of convergence (which extends to our case) of the targeted version of $q_{\boldsymbol{\phi}}$ to the correct posterior density $p(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|\tilde{x}_0, \tilde{\mathcal{X}})$ as the number of simulations per round tends to infinity. Algorithm 1 describes the procedure for obtaining $q(\boldsymbol{\alpha}_0, \boldsymbol{\beta}|\tilde{x}_0, \tilde{\mathcal{X}})$ after R rounds of n simulations.

3.5. Computational aspects

An interesting aspect of the *deepset* architecture of f_{ϕ_3} is that it allows for efficient computations on parallel archi-

Algorithm 1: Sequential posterior estimation for hierarchical models with global parameters

Input : observation $\tilde{x}_0, \tilde{\mathcal{X}}$, prior $p^{(0)}$, simulator $\mathcal{S}$
for *round* $r = 1$ **to** R **do**
  for *sample* $j = 1$ **to** n **do**
    Draw $x_0^j = \mathcal{S}(\boldsymbol{\alpha}_0^j, \boldsymbol{\beta})$ for $(\boldsymbol{\alpha}_0^j, \boldsymbol{\beta}^j) \sim p^{(r-1)}$;
    Draw a set of extra observations $\mathcal{X}^j = \left\{\mathcal{S}(\boldsymbol{\alpha}_i^j, \boldsymbol{\beta}^j) \text{ for } \boldsymbol{\alpha}_i^j \sim p^{(0)}(\cdot|\boldsymbol{\beta}^j)\right\}_{i=1}^N$;
  Train $q_{\boldsymbol{\phi}^{(r)}}$ to minimize $\mathcal{L}_{\boldsymbol{\alpha}}^n + \mathcal{L}_{\boldsymbol{\beta}}^n$;
  Set next proposal $p^{(r)} = q_{\boldsymbol{\phi}^{(r)}}(\cdot|\tilde{x}_0, \tilde{\mathcal{X}})$;
return *posterior* $q_{\boldsymbol{\phi}^{(R)}}(\cdot|\tilde{x}_0, \tilde{\mathcal{X}})$

tectures such as GPUs. Indeed, for a batch of n_b samples $(x^j, \mathcal{X}^j)$, one needs to apply the embedding function $h_{\phi_3^{(2)}}$ to all x_i^j. This can be done efficiently by considering that all these samples form a batch of $n_b \times N$ observations. The outputs of the embedding are then summed and the computational complexity of the following operations does not depend on N anymore. Besides, as the embedding step is often the computational bottleneck of such methods, the use of vectorized batches allows to alleviate the increase in computations caused by adding extra observations.

4. Experiments

All experiments described next are implemented with Python (Python Software Fundation, 2017) and the `sbi` package (Tejero-Cantero et al., 2020) combined with PyTorch (Paszke et al., 2019), Pyro (Bingham et al., 2018) and `nflows` (Durkan et al., 2020a) for posterior estimation [1].

In all experiments, we use the Adam optimizer (Kingma & Ba, 2014) with default parameters, a learning rate of 5.10^{-4} and a batch size of 100.

4.1. Results on the motivating example

To evaluate the impact of leveraging multiple observations when estimating the parameters of a hierarchical model, we use the model presented in Section 2.1, where the observation x_0 is obtained as the product of two parameters α_0 and β_0 with independent uniform prior distributions in $[0, 1]$ (we consider the case where $\sigma = 0$). The set of extra observations $\mathcal{X} = \{x_i\}_{i=1}^N$ is obtained by fixing the same global parameter β_0 for all x_i and sampling local parameters α_i from the prior distribution.

Our approximation to the posterior distribution consists of two conditional neural spline flows of linear order (Durkan et al., 2019), $q_{\boldsymbol{\phi}_1}$ and $q_{\boldsymbol{\phi}_2}$, both conditioned by dense neural networks with two layers and 20 hidden units each. We use neural spline flows because of the highly non-Gaussian aspect of the analytic marginal posterior distributions, which can be well captured by this class of normalizing flows. In general, however, the true posterior distribution is not available, so using other classes of normalizing flows might be justifiable, especially if one's main goal is simply to identify a set of parameters generating a given observation. We set the function $f_{\boldsymbol{\phi}_3}$ to be simply an averaging operation over the elements of $\mathcal{X}$ as the observations in this case are scalar. So the only parameters to be learned in Algorithm 1 are $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$.

We first illustrate in Figure 1 the analytic posterior distribution $p(\alpha, \beta|x_0, \mathcal{X})$ and the approximation $q_{\boldsymbol{\phi}}(\alpha, \beta|x_0, \mathcal{X})$ with no extra observations (*top*; $N = 0$), with ten extra-observations (*middle*; $N = 10$), and one hundred extra-observations (*bottom*; $N = 100$). When only x_0 is available, we observe a ridge shape in the joint posterior distribution, typical of situations with indeterminacies where all solutions $(\frac{x}{\beta}, \beta)$ have the same probability. The addition of a few extra observations resolves this indeterminacy and concentrates the analytic posterior distribution on a reduced support $[x_0, \min(1, \frac{x_0}{\mu})] \times [\mu; 1]$, where $\mu = \max(\{x_0\} \cup \mathcal{X})$. Moreover, on this support, the solutions are no longer equally probable due to the β^{-N} factor that increases the probability of solutions close to μ. Note also that in all three cases the estimated posterior is close to the analytic one.

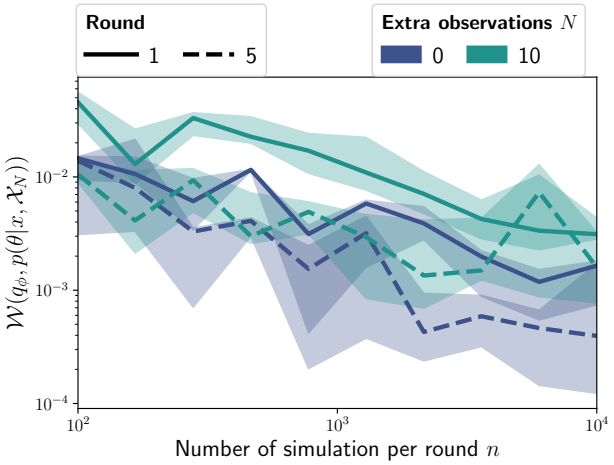

Figure 2. Results on the toy model $x = \alpha\beta$. Evolution of the Sinkhorn divergence $\mathcal{W}_\epsilon$ between the analytic posterior distribution $p(\theta|x_0, \mathcal{X}_N)$ learned posterior q_ϕ and our approximation $q_{\boldsymbol{\phi}}$ trained with an increasing number of simulations per round n. The larger the simulation budget, the closer the learned posterior is to the analytic one for any number of extra observation. Sequential refinement of the posterior (*dash*) helps to capture the posterior.

[1] Code is available upon request and will be made public once the work is published.

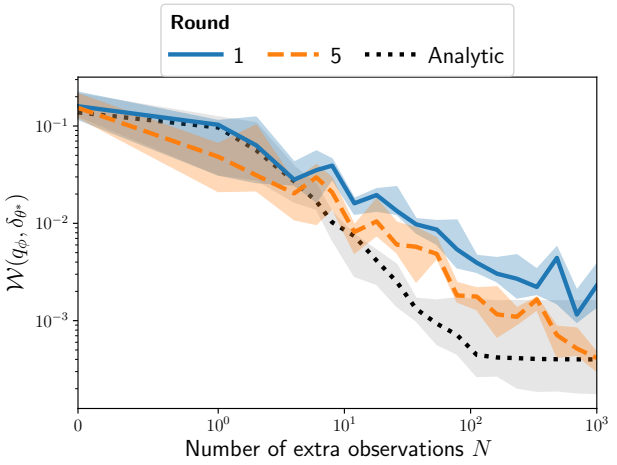

Figure 3. Results on the toy model $x = \alpha\beta$ with $\sigma = 0$. Evolution of the Sinkhorn divergence $\mathcal{W}_\epsilon$ between the learned posterior q_ϕ and the Dirac distribution centered in the true parameters δ_{θ^*} with the number of extra observations N. With more extra observations, all posteriors get more concentrated around the true parameters.

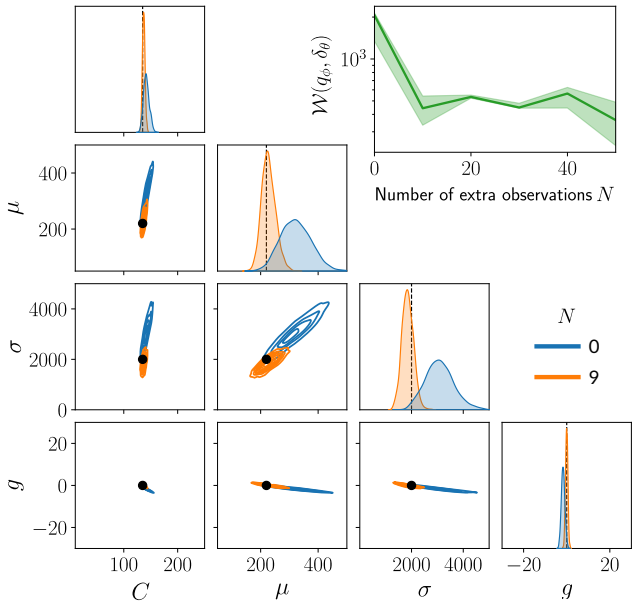

Figure 4. Posterior estimates for the parameters of the neural mass model obtained on 8 s of data sampled at 128 Hz and simulated using $C = 135$, $\mu = 220$, $\sigma = 2000$, and $g = 0$. One can observe that increasing N allows to concentrate the posterior on the correct parameters.

To have a quantitative evaluation of the quality of our approximations q_ϕ, in Figure 2 we display the Sinkhorn divergence (Feydy et al., 2019) $\mathcal{W}_\epsilon$ for $\epsilon = 0.05$ between the analytical posterior $p(\alpha, \beta|x_0, \mathcal{X})$ and our approximation for different numbers of simulations per round (cf. Algorithm 1). The curves display the median value for nine repetitions with different choices of (α_0, β_0) and the transparent area represent the first and the third quartiles. As expected, we note that as the number of simulations per round increases, the approximation gets closer to the analytic solution. The figure also confirms the intuition that, in general, the sequential refinement of multiple rounds leads to better approximations of the true posterior distribution for a fixed observation.

Figure 3 shows how the analytic and estimated posterior distributions tend to concentrate around a given point in the (α, β) space as the number of extra observation N increases. We display the Sinkhorn divergence $\mathcal{W}_\epsilon$ between the learned posterior distribution q_ϕ and the Dirac distribution δ_θ centered in $\theta = (\alpha_0, \beta_0)$. As in the previous figure, the curves represent the median value for nine repetitions with different θ and the transparent areas represent the first and the third quartiles. We see that for both the analytic posterior distribution, as well as its approximation q_ϕ obtained with one and five rounds of training, the distance to the Dirac decreases as more observations are added to $\mathcal{X}$. Here, again, the sequential approach to refine the posterior for the observed data improves the results (*dash*) compared to a single amortized round (*solid*). Note that the performance of the analytic posterior seems to plateau at $N = 100$, due to numerical errors in the computation of the Sinkhorn divergence.

4.2. Inverting a non-linear model from neuroscience

We consider a class of non-linear models from computational neuroscience known as *neural mass models* (Jansen & Rit, 1995) (NMM). These models of cortical columns consist of a set of physiologically motivated stochastic differential equations able to replicate oscillatory electrical signals observed with electroencephalography (EEG) or using intracranial electrodes (Deco et al., 2008). Such models are used in large-scale simulators (Sanz Leon et al., 2013) to generate realistic neural signals oscillating at different frequencies and serve as building blocks for several simulation studies in cognitive and clinical neuroscience (Aerts et al., 2018). In what follows, we focus in the stochastic version of such models presented in Ableidinger et al. (2017) and use the C++ implementation in the supporting code of (Buckwar et al., 2019). In simple terms, the NMM that we consider may be seen as a generative model taking as input a set of four parameters and generating as output a time series x. The parameters of the neural mass model are:

- C, which represents the degree of connectivity between excitatory and inhibitory neurons in the cortical column modelled by the NMM. This connectivity is at the root of the temporal behavior of x and only certain ranges of values generate oscillations.
- μ and σ model the statistical properties of the incoming

oscillations from other neighbouring cortical columns. They drive the oscillations of the NMM and their amplitudes have a direct effect on the amplitude of x.

- g represents a gain factor relating the amplitude of the physiological signal s generated by the system of differential equations for a given set (C, μ, σ), and the electrophysiology measurements x, expressed in Volts.

The reader is referred to Appendix B of the supplementary materials for the full description of the stochastic differential equations defining the neural mass model.

Note that the NMM described above suffers from indeterminacy: the same observed signal x_0 could be generated with larger (smaller) values of g and smaller (larger) values of μ and σ. Fortunately, it is common to record several chunks of signals within an experiment, so other auxiliary signals $x_1, \ldots, x_N$ obtained with the same instrument setup (and, therefore, the same gain g) can be exploited. Using the formalism presented in Section 3, we have that $\boldsymbol{\alpha} = (C, \mu, \sigma)$ and $\boldsymbol{\beta} = g$.

In what follows, we describe the results obtained when approximating the posterior distribution $p(C, \mu, \sigma, g|x_0, \mathcal{X})$ with Algorithm 1 using $R = 2$ rounds and $n = 50000$ simulations per round. Each simulation corresponds to 8 seconds of a signal sampled at 128 Hz, so each simulation outputs a vector of 1024 samples. The prior distributions of the parameters are independent uniform distributions defined as:

$$C \sim \mathcal{U}(10, 250) \quad \mu \sim \mathcal{U}(50, 500)$$
$$\sigma \sim \mathcal{U}(0, 5000) \quad g \sim \mathcal{U}(-30, +30)$$

where the intervals were chosen based on a review of the literature on neural mass models (Jansen & Rit, 1995; David & Friston, 2003; Deco et al., 2008). Note that the gain parameter g is given in decibels (dB), which is a standard scale when describing amplifiers in experimental setups. We have, therefore, that $x(t) = 10^{g/10} s(t)$.

It is standard practice in likelihood-free inference to extract summary features from both simulated and observed data in order to reduce its dimensionality while describing sufficiently well the statistical behavior of the observations. In the present experiment, the summary features consist of the logarithm of the power spectral density (PSD) of each observed time series (Percival & Walden, 1993). The PSD is evaluated in 33 frequency bins between zero and 64 Hz (half of the sampling rate). This leads to a setting with 4 parameters to estimate given observations defined in a 33-dimensional space.

The normalizing flows $q_{\boldsymbol{\phi}_1}$ and $q_{\boldsymbol{\phi}_2}$ used in our approximations are masked autoregressive flows (MAF) (Papamakarios et al., 2017) consisting of five stacked masked autoencoders (MADE) (Germain et al., 2015), each with two hidden layers of 50 units, and a standard normal base distribution as input to the normalizing flow. This choice of architecture provides sufficiently flexible functions capable of approximating complex posterior distributions. We refer the reader to Papamakarios et al. (2019a) for more information on the different types of normalizing flows. We fix function $f_{\boldsymbol{\phi}_3}$ to be a simple averaging operation over the elements of $\mathcal{X}$, so only parameters $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ are learned from data.

Results on simulated data. We first consider a case in which the observed time series x_0 is simulated by the neural mass model with a particular choice of input parameters. In the lower left part of Figure 4, we display the smoothed histograms of the posterior approximation $q_{\boldsymbol{\phi}}$ obtained when conditioning on just x_0 ($N = 0$) or x_0 and $\mathcal{X}$ with $N = 9$. We see that when $N = 0$, parameters μ and σ have large variances and that some of the pairwise joint posterior distributions have a ridge shape that reveals the previously described indeterminacy relation linking g with μ and σ. When $N = 9$, the variances of the parameters decrease and we obtain a posterior distribution that is more concentrated around the true parameters generating x_0. This concentration is explained by the sharper estimation of the g parameter, which is obtained using x_0 and ten auxiliary observations. In the upper right part of Figure 4, we evaluate how our approximation concentrates around the ground truth parameters when N increases, i.e., how the Wasserstein distance of the posterior approximation to the Dirac distribution centered at the true parameters evolves with N. We consider five different choices of ground truth parameters and report the median distance as well as the first and third quartiles. We see that the concentration tends to plateau when $N = 9$ in all cases, indicating that the estimation of the g parameter seems to attain its lowest possible variance and that it no longer improves the knowledge about other parameters.

We have also considered a setting in which the summary statistics of the observed time series are learned from the data instead of being fixed to the log power spectral densities, *i.e.* when $f_{\boldsymbol{\phi}_3}$ is learned. We have used the YuleNet architecture proposed by Rodrigues & Gramfort (2020) on the example with neural mass models and report the results in Section B of the supplementary materials. In all our experiments, we did not see significant changes in the performance of our model so we did not included it in our evaluation as it increased the complexity of the model and its computational burden.

Results on EEG data. Most commonly observed oscillations observed in EEG are known as α waves (Lopes da Silva, 1991). Such waves, characterized by their frequency

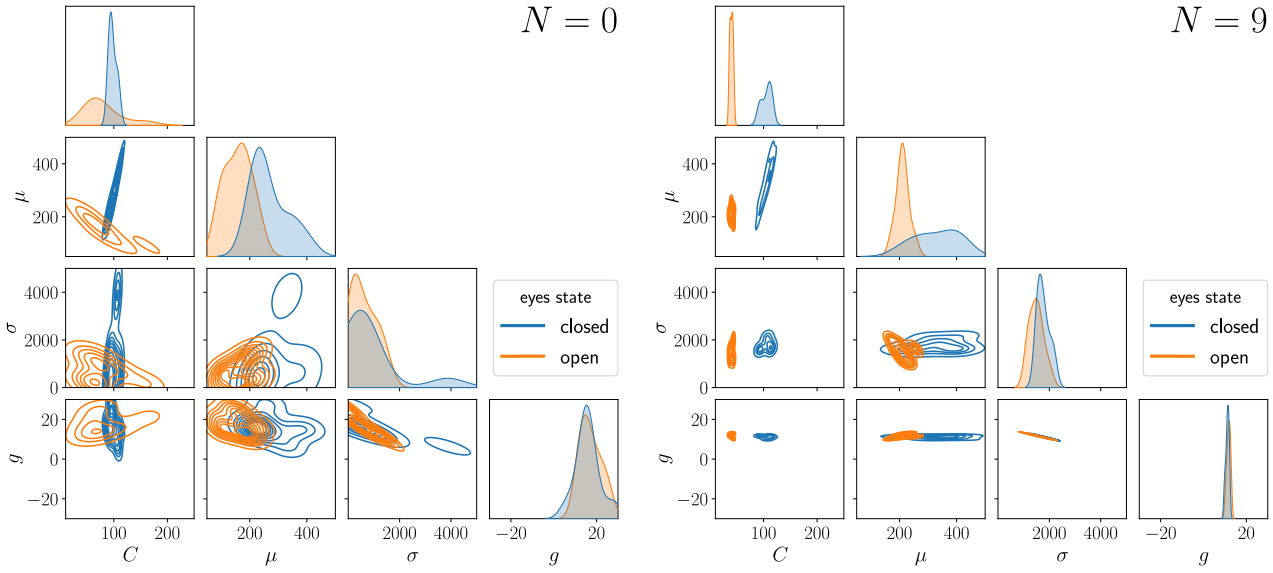

Figure 5. Posterior estimates for the parameters of the neural mass model computed on human EEG signals. Data were collected in two different experimental conditions: eyes closed (in blue) or eyes open (in orange). All signals are 8 s long and recorded at 128 Hz. We see that when $N = 9$ the posterior distributions concentrates, and that the global gain parameter gets similar in both eyes conditions. We observe that the posterior on the 3 parameters of the neural mass model clearly separate between the 2 conditions when $N = 9$.

around 10 Hz, are modulated in amplitude by attention and are typically strengthened when closing our eyes. To relate this phenomenon to the underlying biophysical parameters of the NMM model, we estimated the posterior distribution over the 4 model parameters on EEG signals recorded during short periods of eyes open or eyes closed. Data consists of recordings taken from a public dataset (Cattan et al., 2018) in which subjects were asked to keep their eyes open or closed during periods of 8 s (sampling frequency of 128 Hz). Results for one subject of the dataset are presented in Figure 5 with x_0 being either a recording with eyes closed (in blue) or eyes open (in orange). We consider situations in which no extra-observations are used for the posterior approximation ($N = 0$) or when $N = 9$ additional observations from both eyes-closed and eyes-open conditions are available. When $N = 9$, we observe as expected that the gain parameter, which is global, concentrates for both eyes conditions. More interestingly, we observe that the posterior on the 3 parameters of the neural mass model clearly separate between the 2 conditions when $N = 9$. Looking at parameter C, we see that it concentrates around 130 for the eyes closed data while it peaks around 70 for eyes open. This finding is perfectly inline with previous analysis of the model (Jansen & Rit, 1995). Signals used in this experiment are presented in Appendix C.

Discussion

In this work, we propose a likelihood-free inference approach able to leverage a set of additional observations to boost the estimation of the posterior. This improvement is made possible by a hierarchical model where all available observations share certain global parameters. A dedicated neural network architecture based on normalizing flows is proposed, as well as a training procedure based on simulations from the model. It should be mentioned that although the number of additional observations (N) was fixed in our analysis and experiments, this parameter could be randomized and amortized during learning, at least for the first round. This would enable the posterior approximation to be fed with sets of auxiliary observations of varying sizes, making it more flexible for applications.

Note, also, that each simulated time series $x_0^j = \mathcal{S}(\boldsymbol{\alpha}_0^j, \boldsymbol{\beta}^j)$ in Algorithm 1 is accompanied by N other simulations $x_i^j = \mathcal{S}(\boldsymbol{\alpha}_i^j, \boldsymbol{\beta}^j)$ $(i = 1, \dots, N)$ which are only used as additional observations in the posterior distribution and aggregated by $f_{\boldsymbol{\phi}_3}$. We could improve the efficiency of the training procedure by also considering permutations in which each of these time series x_i^j are also used as the observation x_0. However, this would lead to training batches which are not independent and might cause the model to overfit. We defer this investigation for future work.

Equipped with our posterior approximation for hierarchical models, we demonstrated that it could reliably be applied to neuroscience considering a stochastic model with non-linear differential equations. Very encouraging results on human EEG data open the door to more biologically informed descriptions and quantitative analysis of such non-invasive recordings.

References

Ableidinger, M., Buckwar, E., and Hinterleitner, H. A stochastic version of the Jansen and Rit neural mass model: Analysis and numerics. *The Journal of Mathematical Neuroscience*, 7(1), August 2017. doi: 10.1186/s13408-017-0046-4.

Aerts, H., Schirner, M., Jeurissen, B., Van Roost, D., Achten, E., Ritter, P., and Marinazzo, D. Modeling brain dynamics in brain tumor patients using the virtual brain. *eNeuro*, 5(3), June 2018. ISSN 2373-2822. Society for Neuroscience.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):9931022, March 2003. ISSN 1532-4435.

Brehmer, J., Mishra-Sharma, S., Hermans, J., Louppe, G., and Cranmer, K. Mining for dark matter substructure: Inferring subhalo population properties from strong lenses with machine learning. *The Astrophysical Journal*, 886 (1):49, 2019.

Buckwar, E., Tamborrino, M., and Tubikanec, I. Spectral density-based and measure-preserving ABC for partially observed diffusion processes. an illustration on hamiltonian SDEs. *Statistics and Computing*, 30(3):627–648, November 2019. doi: 10.1007/s11222-019-09909-6.

Cattan, G., Rodrigues, P. L. C., and Congedo, M. EEG alpha waves dataset. December 2018. doi: 10.5281/zenodo.2348892.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. ISSN 0027-8424.

David, O. and Friston, K. J. A neural mass model for MEG/EEG:. *NeuroImage*, 20(3):1743–1755, November 2003. doi: 10.1016/j.neuroimage.2003.07.015.

Deco, G., Jirsa, V. K., Robinson, P. A., Breakspear, M., and Friston, K. The dynamic brain: From spiking neurons to neural masses and cortical fields. *PLOS Computational Biology*, 4(8):1–35, 08 2008. doi: 10.1371/journal.pcbi.1000092.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 7511–7522, 2019.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. nflows: normalizing flows in PyTorch. November 2020a. doi: 10.5281/zenodo.4296287.

Durkan, C., Murray, I., and Papamakarios, G. On contrastive learning for likelihood-free inference. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2771–2781. PMLR, 13–18 Jul 2020b.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouve, A., and Peyré, G. Interpolating between optimal transport and mmd using sinkhorn divergences. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89, pp. 2681–2690. PMLR, 16–18 Apr 2019.

Gelman, A. and Hill, J. *Data analysis using regression and multilevel/hierarchical models*, volume Analytical methods for social research. Cambridge University Press, New York, 2007.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: Masked autoencoder for distribution estimation. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 881–889, Lille, France, 07–09 Jul 2015. PMLR.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., cal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9: e56261, sep 2020. ISSN 2050-084X.

Greenberg, D., Nonnenmacher, M., and Macke, J. Automatic posterior transformation for likelihood-free inference. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2404–2414. PMLR, 09–15 Jun 2019.

Hermans, J., Begy, V., and Louppe, G. Likelihood-free MCMC with amortized approximate ratio estimators. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 4239–4248. PMLR, 13–18 Jul 2020.

Jansen, B. H. and Rit, V. G. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. *Biological Cybernetics*, 73 (4):357–366, September 1995. doi: 10.1007/bf00199471.

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

Lopes da Silva, F. Neural mechanisms underlying brain waves: from neural membranes to networks. *Electroencephalography and Clinical Neurophysiology*, 79(2):81 – 93, 1991. ISSN 0013-4694. doi: https://doi.org/10.1016/0013-4694(91)90044-5.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. Flexible statistical inference for mechanistic models of neural dynamics. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 1289–1299, 2017.

Papamakarios, G. and Murray, I. Fast ϵ-free inference of simulation models with bayesian conditional density estimation. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29, pp. 1028–1036, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 2338–2347. Curran Associates, Inc., 2017.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019a.

Papamakarios, G., Sterratt, D., and Murray, I. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. volume 89 of *Proceedings of Machine Learning Research*, pp. 837–848. PMLR, 16–18 Apr 2019b.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12, Vancouver, BC, Canada, 2019.

Percival, D. B. and Walden, A. T. *Spectral Analysis for Physical Applications*. Cambridge University Press, 1993. doi: 10.1017/CBO9780511622762.

Python Software Fundation. Python Language Reference, version 3.6, 2017.

Rodrigues, P. L. C. and Gramfort, A. Learning summary features of time series for likelihood free inference. *arXiv preprint arXiv:2012.02807*, 2020.

Salakhutdinov, R., Tenenbaum, J. B., and Torralba, A. Learning with hierarchical-deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8):1958–1971, 2013. doi: 10.1109/TPAMI.2012.269.

Sanz Leon, P., Knock, S., Woodman, M., Domide, L., Mersmann, J., McIntosh, A., and Jirsa, V. The virtual brain: a simulator of primate brain network dynamics. *Frontiers in Neuroinformatics*, 7:10, 2013. ISSN 1662-5196. doi: 10.3389/fninf.2013.00010.

Teh, Y. W. and Jordan, M. I. Hierarchical bayesian nonparametric models with applications. *Bayesian nonparametrics*, 1:158–207, 2010.

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonalves, P. J., Greenberg, D. S., and Macke, J. H. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020. doi: 10.21105/joss.02505.

Tran, D., Ranganath, R., and Blei, D. Hierarchical implicit models and likelihood-free variational inference. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 5523–5533. Curran Associates, Inc., 2017.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 3391–3401. Curran Associates, Inc., 2017.

A. Derivations of the posterior distributions for the motivating example

A.1. Single observation

From Bayes' rule we have that

$$p(\alpha, \beta|x_0) \propto p(x_0|\alpha, \beta)p(\alpha, \beta) \, . \tag{10}$$

Since ϵ is Gaussian we can write

$$p(x_0|\alpha, \beta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x_0 - \alpha\beta)^2}{2\sigma^2}\right) \, , \tag{11}$$

so that the posterior is

$$p(\alpha, \beta|x_0) \propto \frac{e^{-(x_0-\alpha\beta)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \mathbf{1}_{[0,1]}(\alpha)\mathbf{1}_{[0,1]}(\beta) \, . \tag{12}$$

We obtain an approximation to the normalization constant of $p(\alpha, \beta|x_0)$ by taking $\sigma \to 0$ and noticing that this makes the Gaussian converge to a Dirac distribution,

$$\begin{aligned} Z(x_0) &= \int_0^1 \int_0^1 \frac{e^{-(x_0-\alpha\beta)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \mathrm{d}\alpha\mathrm{d}\beta \, , \\ &\approx \int_0^1 \int_0^1 \delta(x_0 - \alpha\beta)\mathrm{d}\alpha\mathrm{d}\beta \, . \end{aligned}$$

Doing a change of variables with $\gamma = \alpha\beta$ the integral becomes

$$Z(x_0) \approx \int_0^1 \left[\int_0^\beta \delta(x_0 - \gamma)\frac{\mathrm{d}\gamma}{\beta}\right] \mathrm{d}\beta \, , \tag{13}$$

$$\approx \int_0^1 \frac{1}{\beta} \, \mathbf{1}_{[x_0,1]}(\beta) \, \mathrm{d}\beta = \Big[\log(\beta)\Big]_{x_0}^1 \, , \tag{14}$$

$$\approx \log(1/x_0) \, . \tag{15}$$

The joint posterior distribution is, therefore,

$$p(\alpha, \beta|x_0) \approx \frac{e^{-(x_0-\alpha\beta)^2/2\sigma^2}}{\log(1/x_0)} \frac{\mathbf{1}_{[0,1]}(\alpha)\mathbf{1}_{[0,1]}(\beta)}{\sqrt{2\pi\sigma^2}} \, . \tag{16}$$

The marginal posterior distributions are calculated also using the fact that $\sigma \to 0$,

$$p(\alpha|x_0) = \int p(\alpha, \beta|x_0)\mathrm{d}\beta \, , \tag{17}$$

$$\approx \frac{\mathbf{1}_{[0,1]}(\alpha)}{\log(1/x_0)} \int_0^1 \delta(x_0 - \alpha\beta)\mathrm{d}\beta \, , \tag{18}$$

$$\approx \frac{1}{\log(1/x_0)} \frac{\mathbf{1}_{[x_0,1]}(\alpha)}{\alpha} \, , \tag{19}$$

$$p(\beta|x_0) \approx \frac{1}{\log(1/x_0)} \frac{\mathbf{1}_{[x_0,1]}(\beta)}{\beta} \, . \tag{20}$$

A.2. Multiple observations

Suppose now that we have a set of N observations $x_1, \ldots, x_N$ which all share the same β as x_0 but each have a different α_i, i.e., $x_i = \alpha_i\beta$ for $i = 1, \ldots, N$ (we consider $\sigma \to 0$ and, therefore, $\varepsilon = 0$). Our goal is to use this auxiliary information to obtain a posterior distribution which is sharper around the parameters generating x_0. We have that for $\mathcal{X} = \{x_1, \ldots, x_N\}$ the posterior may be factorized as

$$p(\alpha, \beta|x_0, \mathcal{X}) = p(\alpha|\beta, x_0)p(\beta|x_0, \mathcal{X}) \, . \tag{21}$$

Using Bayes' rule twice to rewrite the second term, we have

$$p(\beta|x_0,\mathcal{X}) \propto p(x_0,\mathcal{X}|\beta)p(\beta)\,, \tag{22}$$
$$\propto \prod_{i=0}^{N} p(x_i|\beta)\,\mathbf{1}_{[0,1]}(\beta)\,, \tag{23}$$
$$\propto \prod_{i=0}^{N} p(\beta|x_i)\mathbf{1}_{[0,1]}(\beta)\,. \tag{24}$$

Therefore,

$$p(\alpha,\beta|x_0,\mathcal{X}) \propto p(\alpha|\beta,x_0)\prod_{i=0}^{N} p(\beta|x_i)\,, \tag{25}$$
$$\propto p(\alpha,\beta|x_0)\prod_{i=1}^{N} p(\beta|x_i)\,, \tag{26}$$

Using expressions (16) and (20) we obtain

$$p(\alpha,\beta|x_0,\mathcal{X}) \propto \frac{\delta(x_0-\alpha\beta)\mathbf{1}_{[x_0,1]}(\alpha)\mathbf{1}_{[x_0,1]}(\beta)\prod_{i=1}^{N}\mathbf{1}_{[x_i,1]}(\beta)}{\log(1/x_0)\prod_{i=1}^{N}(\log(1/x_i)\beta)}\,. \tag{27}$$

which can be simplified to

$$p(\alpha,\beta|x_0,\mathcal{X}) \propto \frac{\delta(x_0-\alpha\beta)\mathbf{1}_{[x_0,1]}(\alpha)\mathbf{1}_{[\mu,1]}(\beta)}{\prod_{i=0}^{N}\log(1/x_i)\beta^n}\,, \tag{28}$$

where $\mu = \max(\{x_0\}\cup\mathcal{X})$. The normalization constant is

$$\begin{aligned}
Z(x_o,\mathcal{X}) &= \iint p(\alpha|\beta,x_0)\prod_{i=0}^{N} p(\beta|x_i)\,\mathrm{d}\alpha\mathrm{d}\beta\,,\\
&= \int\left(\int p(\alpha|\beta,x_0)\mathrm{d}\alpha\right)\prod_{i=0}^{N} p(\beta|x_i)\mathrm{d}\beta\,,\\
&= \int\prod_{i=0}^{N} p(\beta|x_i)\mathrm{d}\beta\,,\\
&= \int\frac{\mathbf{1}_{[\mu,1]}(\beta)}{\prod_{i=0}^{N}\log(1/x_i)\beta^{N+1}}\mathrm{d}\beta\,,\\
&= \frac{1}{\prod_{i=0}^{N}\log(1/x_i)}\left[\frac{-1}{N\beta^N}\right]_{\mu}^{1}\\
&= \frac{\left(1/\mu^N-1\right)}{N\prod_{i=0}^{N}\log(1/x_i)}
\end{aligned}$$

Then, finally, we obtain

$$p(\alpha,\beta|x_0,\mathcal{X}) = \frac{\delta(x_0-\alpha\beta)\mathbf{1}_{[0,1]}(\alpha)\mathbf{1}_{[\mu,1]}(\beta)}{(1/\mu^N-1)}\frac{N}{\beta^N}\,. \tag{29}$$

Simple integrations show that

$$p(\alpha|x_0,\mathcal{X}) = \frac{\mathbf{1}_{[x_0,\min(1,\frac{x_0}{\mu})]}(\alpha)N\alpha^{N-1}}{(1/\mu^N-1)\,x_0^N} \tag{30}$$
$$p(\beta|x_0,\mathcal{X}) = \frac{\mathbf{1}_{[\mu,1]}(\beta)N}{(1/\mu^N-1)\,\beta^{N+1}} \tag{31}$$

B. The neural mass model

B.1. A cortical column as a system of stochastic differential equations

The neural mass model used in our work is the one presented in Ableidinger et al. (2017). This is an extension of the classic Jansen-Rit model (Jansen & Rit, 1995) to make it compatible with a framework based on stochastic differential equations. The model describes the interactions between excitatory and inhibitory interneurons in a cortical column of the brain. In mathematical terms, the model consists of three coupled nonlinear stochastic differential equations of second order, which can be rewritten as a six-dimensional first-order stochastic differential system:

$$\begin{aligned}
\dot{X}_0(t) &= X_3(t) \\
\dot{X}_1(t) &= X_4(t) \\
\dot{X}_2(t) &= X_5(t) \\
\dot{X}_3(t) &= \Big(Aa\big(\mu_3 + \text{Sigm}(X_1(t) - X_2(t)\big) - 2aX_3(t) - a^2X_0(t)\Big) + \sigma_3\dot{W}_3(t) \\
\dot{X}_4(t) &= \Big(Aa\big(\mu_4 + C_2\ \text{Sigm}(C_1X_0(t)\big) - 2aX_4(t) - a^2X_1(t)\Big) + \sigma_4\dot{W}_4(t) \\
\dot{X}_5(t) &= \Big(Bb\big(\mu_5 + C_4\ \text{Sigm}(C_3X_0(t)\big) - 2bX_4(t) - b^2X_2(t)\Big) + \sigma_5\dot{W}_5(t)
\end{aligned} \tag{32}$$

The actual signal that we observe using a EEG recording system is then $X(t) = 10^{g/10}(X_1(t) - X_2(t))$, where g is a gain factor expressed in decibels. According to Jansen & Rit (1995), most physiological parameters in (32) are expected to be approximately constant between different individuals at different experimental conditions, except for the connectivity parameters $(C_1, C_2, C_3, C4)$ and the statistical parameters of the input signal from neighboring cortical columns, modeled by μ_4 and σ_4. Following the setup proposed in Buckwar et al. (2019), we then define our inference problem as that of estimating the parameter vector $\boldsymbol{\theta} = (C, \mu, \sigma, g)$ from an observation $X_{\boldsymbol{\theta}}$, where $\mu = \mu_4$ and $\sigma = \sigma_4$, and the C_i parameters are all related via $C_1 = C, C_2 = 0.8C, C_3 = 0.25, C_4 = 0.25C$.

B.2. Choice of summary statistics

The inference procedure is then carried out not on the time series itself but on a vector of summary statistics. The results described in Section 4.2 were obtained with a fixed choice on the power spectral density of the time series as summary statistics. However, it is possible (and very often preferable) to learn the best summary statistics from data. We have considered this option using the YuleNet proposed in Rodrigues & Gramfort (2020), where a convolutional neural network is jointly learned with the approximation to the posterior distribution. Figure 6 portrays the results obtained with different numbers of auxiliary observations in $\mathcal{X}$. Note that the 'quality' of the approximation seems to stagnate when $N > 10$ as observed also in Figure 4. We did not carry out more experiments on this data-driven setting because of difficulties due to numerical instabilities in the training procedure when N increases and for certain choices of ground truth parameters. Also, the memory consumption using YuleNet with large values of N makes the use of GPU a challenge. We intend to continue investigations with learned summary statistics in future works.

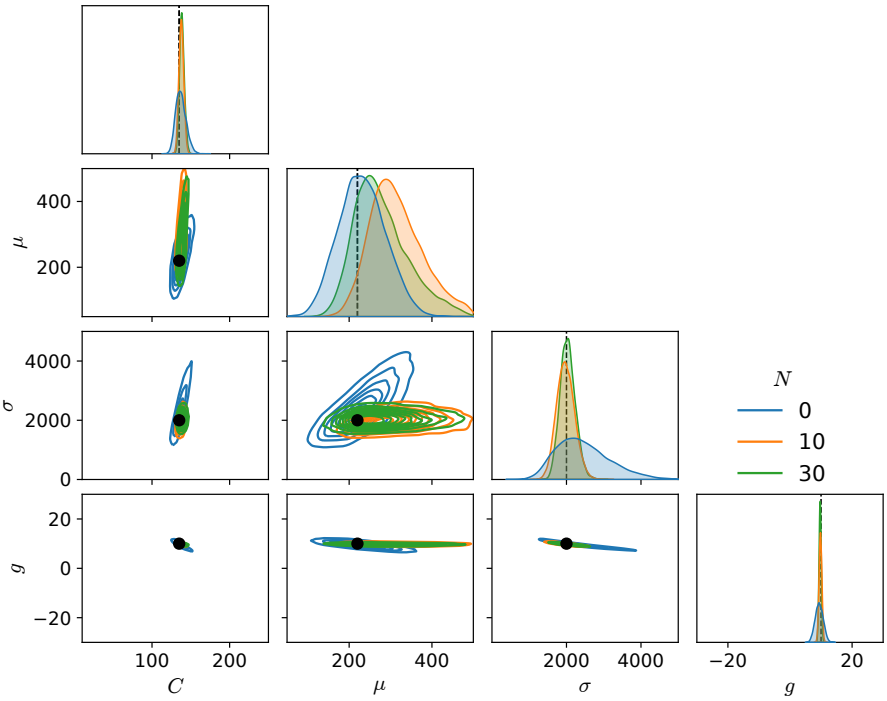

Figure 6. Posterior estimates for the parameters of the neural mass model obtained on 8 s of data sampled at 128 Hz and simulated using $C = 135$, $\mu = 220$, $\sigma = 2000$, and $g = 10$. One can observe that increasing N allows to concentrate the posterior on the correct parameters.

C. EEG data

The EEG signals used for generating the results in Figure 5 are displayed in Figure 7. We have used only the recordings from channel `Oz` because it is placed near the visual cortex and, therefore, is the most relevant channel for the analysis of the open and closed eyes conditions. The signals were filtered between 3 Hz and 40 Hz.

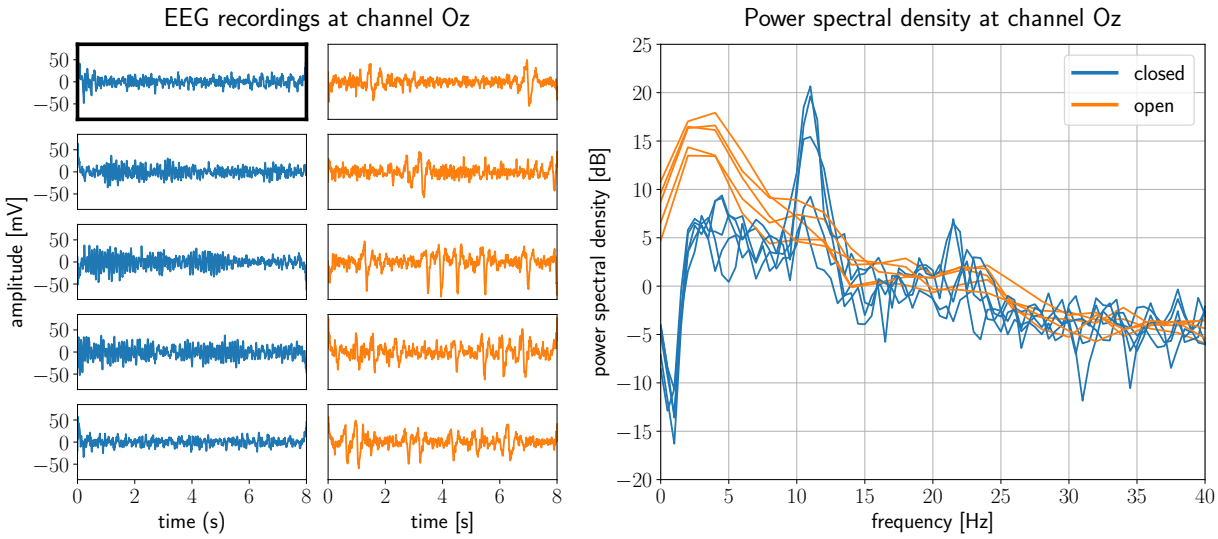

Figure 7. EEG data used on our analysis described in Figure 5. (Left) All ten time series considered in our analysis. The plot with thicker bounding boxes is the observed signal x_0 in the closed eyes state. All other time series belong to $\mathcal{X}$. (Right) Power spectral density of each time series calculated over 33 frequency bins. These are the actual summary features used as input in the approximation of the posterior distribution.