Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

Master Thesis Informatics

Parameter Inference and Uncertainty Quantification for an intermediate complexity climate model

Benedict Röder

10.08.2022

Reviewers

Dr. Bedartha Goswami
(Machine Learning in Climate Science)
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Prof. Dr. Jakob Macke
(Machine Learning for Science)
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Röder, Benedict:
Parameter Inference and Uncertainty Quantification
for an intermediate complexity climate model
Master Thesis Informatics
Eberhard Karls Universität Tübingen
Thesis period: 01.02.2022-14.08.2022

Abstract

Well-adapted parameters in climate models are essential for making accurate predictions for future projections. In climate science, the record of precise and comprehensive observational data is short, and the parameters of climate models are often hand-tuned or learned from artificially generated data. Most popular algorithms for learning parameters from observational data, like the Kalman inversion approach, only provide point estimates of parameters. However, due to limited and noisy data, Bayesian models are preferable to have access to the uncertainties of the inferred parameters. In this work, we compare two Bayesian parameter inference approaches on various toy problems and apply them to the intermediate complexity model for the El Niño-Southern Oscillation by Zebiak & Cane. The "Calibrate, Emulate, Sample" (CES) approach is an extension of the ensemble Kalman inversion which allows posterior inference by emulating the model via Gaussian processes and thereby enables efficient sampling. The "Simulation-Based Inference" (SBI) approach where the approximate posterior distribution is learned from simulated model data using neural networks. We evaluate the performance of both approaches by comparing their run times and the number of required model evaluations, assess their assumptions and limitations, and examine their posterior distributions.

Abstract

Gut angepasste Parameter sind essenzell für Klimamodellen, um genaue Vorhersagen für zukünftige Entwicklungen zu treffen. In den Klimawissenschaften sind nur wenige Aufzeichnungen von präzisen und flächendeckenden beobachteten Daten vorhanden. Daher werden die Parameter von Klimamodellen oft von Hand bestimmt oder anhand von künstlich erzeugten Daten gelernt. Aufgrund der begrenzten und verrauschten Daten verwenden wir bayessche Modelle, um Zugang zu den Unsicherheiten der inferierten Parameter zu erhalten. Populäre Algorithmen zum Lernen von Parametern von beobachteten Daten, wie zum Beispiel der Kalman-Inversions Ansatz, liefern nur Punktschätzungen der Parameter. In dieser Arbeit vergleichen wir zwei bayessche Parameterinferenzansätze, die auf verschiedene Beispielprobleme und dem Klimamodell mittlerer Komplexität von Zebiak & Cane für das El Niño-Southern Oscillation Phänomen angewendet werden. Der "Calibrate, Emulate, Sample" (CES) Ansatz ist eine Erweiterung der Ensemble Kalman-Inversions Methode und erlaubt Posteriorinferenz, indem das Modell mithilfe von Gauß-Prozessen emuliert wird und daher effizientes Sampling ermöglicht. Der "Simulation-Based Inference" (SBI) Ansatz verwendet neuronale Netzwerke, um eine annähernde Posteriorverteilung anhand von simulierten Modelldaten zu lernen. Wir evaluieren die Leistung beider Ansätze durch den Vergleich der Laufzeiten und der Anzahl an benötigten Modellevaluierungen, beurteilen ihre Annahmen und Grenzen und untersuchen die inferierten Posteriorverteilungen.

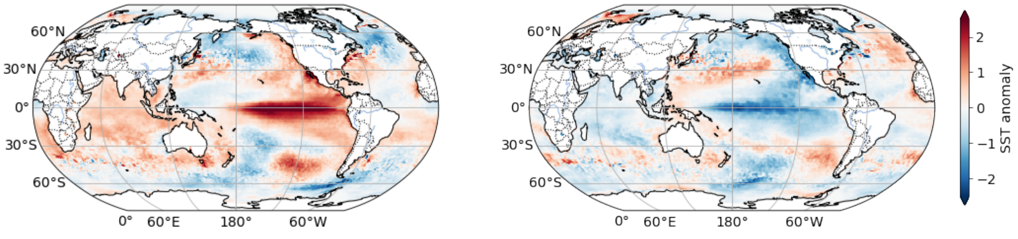
1 Introduction

Natural disasters such as storms, droughts, and floods drastically impact the environment, the economy, and the public health. Last year, these extreme weather events caused damages exceeding 152.6 billion dollars and killed 724 people in the USA alone (NCEI, 2022). As the number of extreme events is projected to increase, climate models are essential to make accurate predictions and can help us to prepare for these scenarios in time.

The largest climate variability on an annual scale is the El Niño-Southern Oscillation (ENSO) which occurs every two to eight years (Latif & Keenlyside, 2009). During the initial phase of an El Niño, the east-to-west trade winds weaken or sometimes even reverse, which allows more warm water to accumulate in the central and eastern Pacific. This weakens the trade winds further, and the system is locked into a positive feedback loop referred to as Bjerknes feedback (Bjerknes, 1969). Clouds, evaporation, and rain also shift eastward following the warm water, affecting global circulation patterns. This alternating atmospheric condition is referred to as Southern Oscillation (Troup, 1965). The converse event is termed La Niña and often has the opposite effects. During La Niña events, the trade winds strengthen and more cold water from the deep sea swells up, leading to colder sea surface temperatures in the central Pacific. The Niño-3.4 index is commonly used to classify El Niño or La Niña events. It is the mean sea surface temperature anomaly (SSTA) in the tropical Pacific basin from 5°N-5°S and 170°W-120°W. When a 5-month running average of this index is above 0.4°C for more than six months, we classify it as an El Niño event. Conversely, when it is below -0.4°C, we refer to a La Niña event. The sea surface temperature anomalies of the 2015/2016 El Niño event and 2007/2008 La Niña are shown in Figure 1. Dilley and Heyman (1995) showed that during ENSO years, droughts are twice as frequent as under normal conditions and floods exhibit significantly higher flood volumes (Ward et al., 2014) worldwide. While this is still an active research topic, the influences of ENSO are likely to strengthen with the global warming trend (Cai et al., 2021; Cai et al., 2022; McGregor et al., 2022). An accurate prediction of the ENSO cycle is crucial for taking early precautions. However, predicting the ENSO cycle remains challenging, and high prediction skill is only achieved for short lead times up to 6 months (Yang et al., 2022). Additionally, forecasts made through spring generally exhibit lower prediction skills which is known as the spring predictability barrier (SPD) (Duan & Wei, 2013).

The first dynamical model to successfully recreate the key characteristics of the ENSO cycle was the Cane-Zebiak model (McPhaden et al., 2020). The Cane-Zebiak model (CZ model) is a coupled ocean-atmosphere model developed in the 1980s by Mark A. Cane & Stephen E. Zebiak (Zebiak, 1984; Zebiak, 1986; Zebiak & Cane, 1987). The CZ model (or ZC model) is also known as *Lamont-Doherty Earth Observation* (LDEO) model. The original CZ model was modified and improved several times (Chen et al., 1995; Chen et al., 1998; Chen et al., 2000; Chen et al., 2004). Even though it is not state of the art anymore, the latest version (LDEO5) is still used as a baseline today in ENSO forecasts. (IRI, 2022).

These models often contain parameterizations that simplify complex physical processes or approximate processes that are too small-scale. Finding appropriate parameter values is essential for making accurate predictions (Wu et al., 2016). However, since the record of precise climate data is short, these parameters have typically been hand-tuned. Learning parameters from data is referred to as



(a)

(b)

Figure 1: (a) El Niño event 2015/2016 and (b) La Niña 2007/2008 event. The shadings depict the average sea surface temperature anomalies (SSTAs) across December, January, and February.

parameter inference or parameter optimization. Derivative-based parameter optimization is often not possible in complex models (such as climate models) due to an intractable likelihood. For these models, a closed-form solution for the likelihood is either unavailable or cannot be evaluated in practice due to an integration over too many variables. Hence, popular optimization techniques such as gradient descent methods (Ruder, 2016) or Newton's method (Polyak, 2007) cannot be applied. Derivative-free optimization algorithms treat the parameterized model as a black box, i.e., no information of the gradients and internal states are available. These black box models receive a set of parameters which are then internally used to produce outputs. Examples of derivative-free optimization algorithms include genetic algorithms (Wright, 1991), simulated annealing (Kirkpatrick et al., 1983) or Kalman filters adapted for parameter estimation (Evensen, 1994; Iglesias et al., 2013; Garbuno-Inigo et al., 2020; Huang et al., 2022). The ensemble Kalman filter has been popular among the climate science community, and Wu et al. (2016), Zhao et al. (2019), and Gao et al. (2021) have used different variants to estimate various parameters of the CZ model. However, these algorithms only return point estimates of the model parameters. In the presence of few and potentially noisy data, we prefer a Bayesian approach to learn distributions over the estimated parameters and quantify their uncertainty. Uncertainty quantification (UQ) allows for better interpretation of the learned parameters, gives us insight into the black box model, and potentially improves forecasts. Approaches for likelihood-free probabilistic parameter estimation include Approximate Bayesian Computation (ABC) (Sisson et al., 2018), Bayesian synthetic likelihood (BSL) (Price et al., 2018), and Bayesian history matching (Craig et al., 1997). These methods are based on sampling input-output pairs of the model. When working with climate models, we must be efficient in the number of required model runs to achieve accurate posterior approximations. Highly complex climate models such as general circulation models (GCMs) model the entire planet in a fine resolution and can be very expensive to run. Having to run these models too often may become computationally infeasible.

The "Calibrate, Emulate, Sample" (CES) approach builds on the success of Kalman methods which require only a few model evaluations, and extends them to allow for probabilistic interpretation (Cleary et al., 2021). Inverse Kalman algorithms are used to calibrate the climate model to a relevant parameter region. In this region, Gaussian processes (GPs) (Rasmussen, 2003) are used to emulate the model output. The GPs yield an accessible likelihood, and posterior parameters can be drawn efficiently via Markov chain Monte Carlo (MCMC) sampling (Metropolis et al., 1953; Hastings, 1970). These samples follow an approximate posterior distribution, allowing us to make statements about their uncertainty. The CES approach has only been applied once by Dunbar et al. (2021) to estimate two parameters of an idealized aquaplanet.

Another approach for estimating the posterior distributions of parameters of black box models is used in the neuroscience community. The algorithms grouped within "Simulation-Based Inference" (SBI) (Cranmer et al., 2020) use artificial neural networks to learn a relationship between parameters and simulated data. The trained neural networks can then be used to make statements about the posterior probability of the parameters. To the best of our knowledge, the SBI methods have never been used on climate models before and are mostly unknown to the climate science community.

In our work, we implement a modified version of the CZ model that allows long artificial simulations of the ENSO cycle. This is the first open-source implementation of the CZ model in Python 3. We apply the probabilistic parameter inference frameworks of CES and SBI to a set of toy problems and, for the first time, to simulated ENSO data from the CZ model. The first toy problem is a linear model for which a closed-form posterior solution is available. This allows us to compare the learned distributions to the true posterior distribution quantitatively. An oscillatory model is used to analyze the algorithms when faced with a multimodal posterior distribution. The third toy problem is a chaotic system described by three ordinary differential equations often used in atmospheric sciences. In our final experiment, we apply the algorithms to simulated ENSO data from our CZ model. Using those experiments, we emphasize the strengths, weaknesses, and limitations of the two approaches and outline their fundamental differences. We evaluate their performances by investigating the learned posterior distributions and analyzing the number of required model evaluations.

This work is structured as follows; in the next chapter, we formulate the problem and present CES and SBI, two probabilistic parameter estimation approaches that can be applied to black box models. Also, we briefly depict the Cane-Zebiak model. In section 3, the algorithms are applied to different toy problems and to the climate model data. In chapter 4, we discuss our results, compare the performance of both algorithms across the tasks, and present possible extensions. Finally, chapter 5 concludes the work.

2 Methods

Observing data can be formulated as the result of a generative process given by

$$y = \mathcal{G}(\theta) + \eta, \tag{1}$$

where $y \in \mathbb{R}^d$ are the observations, $\mathcal{G}$ is called the forward map $\mathcal{G} : \mathbb{R}^p \to \mathbb{R}^d$, $\theta \in \mathbb{R}^p$ are parameters and η is observational noise assumed to follow a Gaussian distribution $\eta \sim \mathcal{N}(0, \Gamma_y)$. The forward map $\mathcal{G}$ could be a climate model that takes parameters θ as input and generates a climate projection $\mathcal{G}(\theta)$. In the SBI literature, a simulator represents this generative process.

Following the Bayesian approach, we are interested in inferring distributions over the parameters θ for the inverse problem 1. However, for complex models, the likelihood typically is intractable because it either has no analytic solution or requires computationally infeasible integrations over all latent variables. Hence, derivative-based methods cannot be applied. Another crucial factor is how many times we have to run the model $\mathcal{G}$ or the simulator since for highly complex climate models running $\mathcal{G}$ many times can be prohibitively expensive. Following, we present two algorithms that allow for likelihood-free probabilistic posterior inference for the inverse problem 1.

2.1 Calibrate, Emulate, Sample (CES)

The "Calibrate, Emulate, Sample" (CES) (Cleary et al., 2021) approach extends ensemble Kalman methods to allow for probabilistic parameter estimation. It consists of three eponymous phases. The first step generates a data set ("Calibrate") on which a surrogate model is trained ("Emulate") from which samples can be efficiently drawn ("Sample") to estimate the posterior distribution.

During the calibration step, we run an ensemble Kalman method adapted for inverse problems. Popular algorithms are the ensemble Kalman inversion (EKI) (Iglesias et al., 2013), ensemble Kalman sampler (EKS) (Garbuno-Inigo et al., 2020), and unscented Kalman inversion (UKI) (Huang et al., 2022). These derivative-free optimization techniques seek to find the optimal parameters for the inverse problem 1. Cleary et al. (2021) showed that ensemble Kalman inversion (EKI) works well in practice, and hence the EKI method will be described in more detail.

The EKI method can be understood as a system of interacting particles. Each particle corresponds to one parameter point estimate. Each particle j in the ensemble gets updated at iteration n according to the following formula:

$$\theta_{n+1}^{(j)} = \theta_n^{(j)} - \frac{\Delta t_n}{J} \sum_{k=1}^{J} \Big\langle \mathcal{G}(\theta_n^{(k)}) - \bar{\mathcal{G}}_n, \Gamma_y^{-1}\Big(\mathcal{G}(\theta_n^{(j)}) - y\Big)\Big\rangle \theta_n^{(k)}, \tag{2}$$

where J is the number of particles, Δt_n is an adaptive timestep or learning rate, the subscripts $n = 1, ..., N$ are the iterations, and the angle brackets denote the Euclidean inner product. $\bar{\mathcal{G}}_n$ corresponds to the mean prediction of the ensemble at iteration n given by

$$\bar{\mathcal{G}}_n = \sum_{k=1}^{J} \mathcal{G}(\theta_n^{(k)}). \tag{3}$$

Intuitively, the first term in the inner product of the update equation 2 drives the particles to consensus and the second term leads them to agree on the observation y we condition them on. This is scaled by the inverse of the noise covariance matrix, which causes a stronger collapse in dimensions with low noise than in dimensions with higher noise. During the Kalman inversion, each particle corresponds to one model run (i.e. simulates data) in every iteration. This update scheme moves particles quickly into a region of high posterior mass. However, their resulting particle spread generally does not offer information about the true variance of the posterior distribution. The algorithm often converges within ten iterations, and as a rule of thumb, the ensemble size can be set to $J = 10 \cdot p$ as a starting point (CliMA, 2022), where p are the dimensions of the parameter. The purpose of the calibration step is to construct a small data set of $\{\theta_n^{(k)}, \mathcal{G}(\theta_n^{(k)})\}_{k=1}^{J}$ pairs for $n = 1, ..., N$ which are mostly located in a region of high posterior mass.

During the emulate step, Gaussian processes (GPs) (Rasmussen, 2003) are used to emulate the true forward model $\mathcal{G}^{(M)} \approx \mathcal{G}$. A subset of the generated $\{\theta_n^{(k)}, \mathcal{G}(\theta_n^{(k)})\}_{k=1,n=1}^{J,N}$ pairs with cardinality

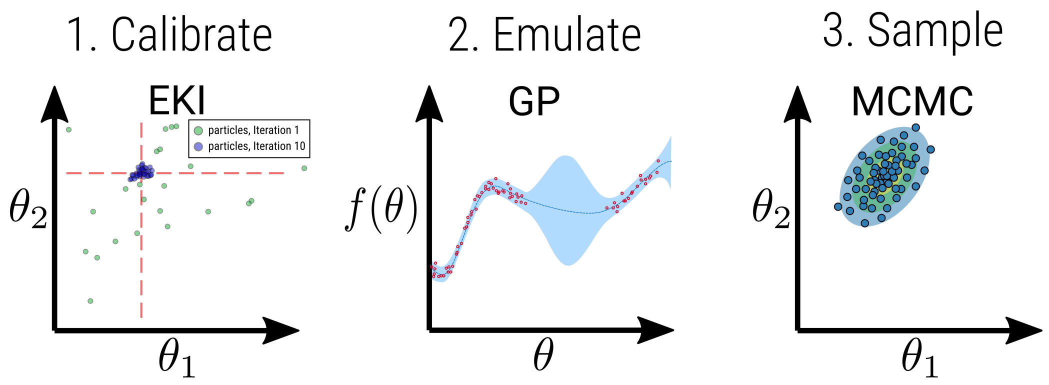


Figure 2: 1. EKI produces input-output pairs $\{\theta_n^{(k)}, \mathcal{G}(\theta_n^{(k)})\}_{k=1}^J$ for $n = 1, ..., N$.
2. Train GPs on input-output pairs to approximate the model $\mathcal{G}^{(M)}(\theta) \approx \mathcal{G}(\theta)$.
3. MCMC sampling from the GPs produces many (cheap) samples $\{\theta_i\}_{i=1}^{I_s}$.

$M \leq JN$ is used to train the GPs. For every output coordinate $l = 1, ..., d$ one GP is trained independently. The individual GPs are stacked together to form the complete emulator $\mathcal{G}^{(M)} \sim \mathcal{N}(m(\theta), \Gamma_{\text{GP}}(\theta))$. The GPs will give an accurate approximation of the model $\mathcal{G}$ in the region where it has many training points and will be imprecise everywhere else. Since the calibration step moved the particles into the region of high posterior probability, the GPs will be a good approximation for the true forward model in this region. The forward model $\mathcal{G}$ implicitly defines a likelihood. This likelihood function is generally unknown, and it would take many simulations to approximate it precisely. However, the trained GPs now give direct access to a negative log-likelihood function defined by[1]

$$\Phi^{(M)}(\theta) = \frac{1}{2}\|y - m(\theta)\|_{\Gamma_y}^2. \tag{4}$$

The final stage is the sampling step. Here, Markov chain Monte Carlo (MCMC) random walk (Metropolis et al., 1953; Hastings, 1970) is used to sample parameters that follow the posterior distribution. For the initial guess θ_0, the mean of the ensemble of the last iteration $\bar{\theta}_N$ of the calibration is taken to ensure a short transient phase. In each iteration, a new proposal parameter is constructed by applying noise to the current parameter. This proposal parameter is accepted with a probability that depends on the likelihood defined by the GPs. The acceptance probability $a(\theta, \theta^*)$ is calculated by

$$a(\theta, \theta^*) = \min\Big\{1, \exp\Big[\Big(\Phi(\theta) + \frac{1}{2}\|\theta\|_{\Gamma_\theta}^2\Big) - \Big(\Phi(\theta^*) + \frac{1}{2}\|\theta^*\|_{\Gamma_\theta}^2\Big)\Big]\Big\}, \tag{5}$$

where Γ_θ is the prior covariance. This approach is iterated until enough samples have been generated. The accepted samples will then follow an approximate posterior distribution. The algorithm is sketched below.

1. Set the initial parameter $\theta_0 = \bar{\theta}_N$.
2. Construct new proposal: $\theta_{i+1}^* = \theta_i + \xi_i$, where $\xi_i \sim \mathcal{N}(0, C(\Theta_N))$.
3. Accept proposal $\theta_{i+1} = \theta_{i+1}^*$ with probability $a(\theta_i, \theta_{i+1}^*)$; else set $\theta_{i+1} = \theta_i$.
4. $i \to i + 1$, return to 2. until enough samples have been generated with $I_s \gg JN$.

The set of particles at iteration n is denoted with $\Theta_n = \{\theta_n^{(k)}\}_{k=1}^J$ and the $p \times p$ covariance matrix of particles is given by

$$C(\Theta_n) = \frac{1}{J}\sum_{k=1}^{J}\left(\theta_n^{(k)} - \bar{\theta}_n\right) \otimes \left(\theta_n^{(k)} - \bar{\theta}_n\right). \tag{6}$$

[1] For any positive-definite matrix A, define $\|x\|_A = \|A^{-\frac{1}{2}}x\|$

Lastly, we can construct histograms from the accepted samples to gain insights into their posterior distribution, their uncertainties, mean values, and maximum a posteriori (MAP) estimates. Generating the MCMC samples is cheap because we only need to evaluate the trained GPs. Hence, for computationally expensive models (like climate models), almost the entire cost of running CES is within the Calibration step as it requires running the forward model. A visualization of the entire pipeline can be seen in Figure 2.

2.2 Simulation-Based Inference (SBI)

Another approach is to approximate the posterior distribution using neural networks. In Bayesian inference, we want to estimate the posterior distribution $p(\theta|x)$ defined by

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int p(x|\theta')p(\theta')d\theta'} = \frac{p(x|\theta)p(\theta)}{p(x)}. \tag{7}$$

That is the distribution over parameters θ given that we have observed data x. The likelihood $p(x|\theta)$ is essential to estimate the posterior. However, this likelihood often is intractable for complex models. Either no closed form solution for the likelihood is known, or we would have to integrate over so many variables that it becomes computationally impractical. Simulation-Based Inference (SBI), also called likelihood-free inference, allows for Bayesian posterior estimation by making use of a simulator. A *simulator* is a program or model that takes parameters θ to produce data x. The generated data follows an implicit likelihood $p(x|\theta)$. A data set of parameters, together with their simulated data, is used to train a conditional neural density estimator, such as a Mixture Density Network (MDN) (Bishop, 1994), a Masked Autoregressive Flow (MAF) (Papamakarios et al., 2017) or a Neural Spline Flow (NSF) (Durkan et al., 2019). To be more efficient in the number of simulations, the networks can also be trained *sequentially*. Instead of drawing all parameters from the prior directly, the neural network is trained on a smaller set of parameters, and it is used in the subsequent iteration to draw parameters from more interesting regions. The training procedure and which distribution the neural network approximates depend on the various approaches within SBI. Some algorithms estimate a synthetic likelihood (SNLE) (Papamakarios et al., 2019), some target the posterior directly (SNPE) (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019) and others model the likelihood ratio (SNRE) (Hermans et al., 1903; Durkan et al., 2020). An overview of the various approaches to likelihood-free inference can be found in Cranmer et al. (2020) and a benchmark in Lueckmann et al. (2021). Once the conditional neural density estimator is trained, it can be used to estimate the posterior distribution $p(\theta|x)$ for a given observation x. When using the inference algorithms (S)NRE and (S)NLE, a Markov chain Monte Carlo (MCMC) (Metropolis et al., 1953; Hastings, 1970) sampling step is necessary to approximate the posterior distribution as these methods do not learn the posterior density directly. One property of the neural density estimators is that they are *amortized*. This means that one can query them for different observations x and get the corresponding estimated posterior distributions $p(\theta|x)$ without needing to repeat the SBI pipeline.

Following the practical guide from Lueckmann et al. (2021), we chose (S)NPE for this work. It performed robustly across most tasks in their benchmark and generally required fewer simulations compared to conventional Approximate Bayesian Computation (ABC) methods (Sisson et al., 2018). It is suitable for high-dimensional data and offers direct access to the posterior distribution.

In our work, we use the SNPE-C implementation of Greenberg et al. (2019). Here, a neural network with weights ϕ learns the probabilistic relationship between simulated data x and their parameters θ. To achieve this, the network learns a mapping from observations x to distribution parameters ψ. For a simple example, the distribution parameters ψ could be the mean and the covariance of a multivariate Gaussian. This mapping from observations x to the distribution parameters ψ depends on the network weights ϕ. Those weights are optimized by maximizing the log-likelihood of the parameters θ for the observations x. The trained network can then be used on a new observation x_o to give us the distribution parameters that construct a posterior probability distribution over all possible parameters θ. Parameter regions with high posterior probability will generate data similar to the observation x_o, whereas parameters with low posterior probability will generate data inconsistent with x_o. The sampling efficiency during training can be increased by preventing too many draws from low posterior probability regions. In order to achieve this, the neural network can be trained over multiple rounds using the estimated posterior of the latest iteration to draw parameters in the subsequent iteration. However, this biases the posterior estimation and complicates the training procedure. To account for

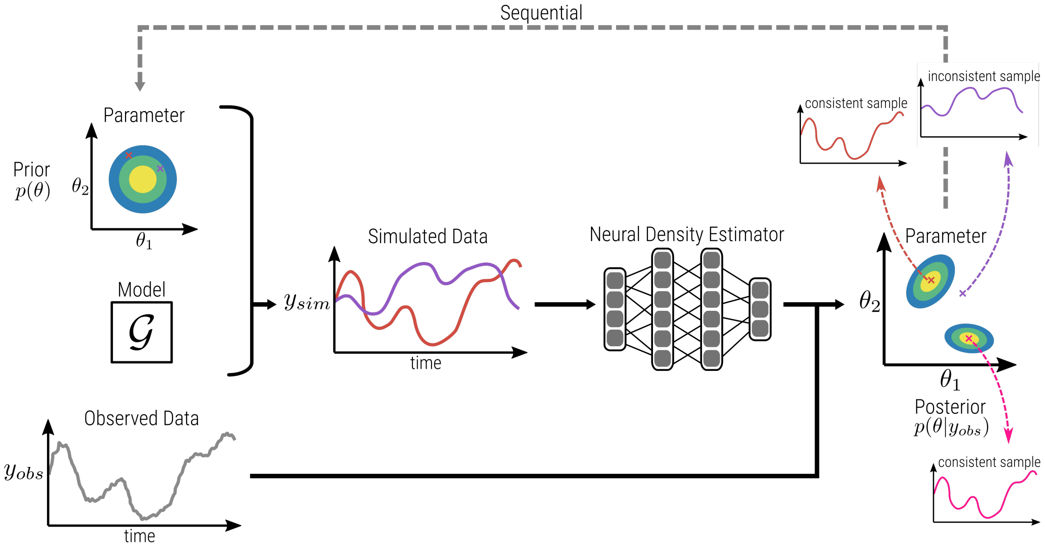


Figure 3: Schematic view of (S)NPE. Parameters are sampled from the prior and run through the simulator (model $\mathcal{G}$) to generate a dataset of simulated data. A neural density estimator learns the approximate posterior distribution from the relationship between parameters and their simulated data. The trained density estimator is queried with observed data and returns a posterior distribution over the parameters. This distribution can also be used *sequentially* as a proposal distribution for parameter draws in the next iteration.

this, Greenberg et al. (2019) use a transformation between proposal posterior and true posterior. First, they optimize the network weights under the proposal posterior and then use the transformation to retrieve the true posterior. This requires the transformation to have a closed-form solution that can be used during training and parameter sampling. Hence, SNPE-C can learn the true posterior even when trained over multiple rounds.

Instead of predicting the distribution parameters ψ of a conventional probability distribution, a conditional neural density estimator is used. Neural density estimators are neural networks that fulfill the properties of basic probability theory. Their outputs must be non-negative and have to integrate to 1 (Papamakarios, 2019). Normalizing flows represent one possible approach for neural density estimation. They can model a complex density through an invertible transformation f of a simple base density (Kobyzev et al., 2020). Typically, the transformations are parameterized by a neural network. In our work, we use Neural Spline Flow (NSF) as the density estimator (Durkan et al., 2019), which uses monotonic rational-quadratic splines instead of affine or additive transformations.

SNPE using NSFs constitutes a powerful approach because NSFs can learn complex distributions, and no knowledge about the black box model is required. An overview of (S)NPE is shown in Figure 3. We also tested (S)NRE, but the overhead of MCMC sampling to reconstruct posterior distributions became infeasible in our experiments.

2.3 The Cane-Zebiak Model

The Cane-Zebiak (CZ) model is an anomaly model that computes anomalies of its internal variables around a prescribed climatological mean state (e.g., average sea surface temperatures for July over the last 30 years). The atmospheric component of the CZ model follows quasi-linear shallow water equations (Gill, 1980), which describe a steady-state response to the sea surface temperature anomalies (SSTAs). Wind stress is generated based on SSTAs and moisture convergence. This wind stress then forces the ocean component. The ocean model is a linear reduced-gravity model that simulates the upper-layer currents and thermocline anomalies driven by the winds from the atmosphere component. Additionally, the ocean dynamics dictate how the temperature in the surface layer changes based on the currents, anomalous cold water upwelling, anomalous thermocline depth, and their respective prescribed climatological mean values. The surface temperatures then affect the

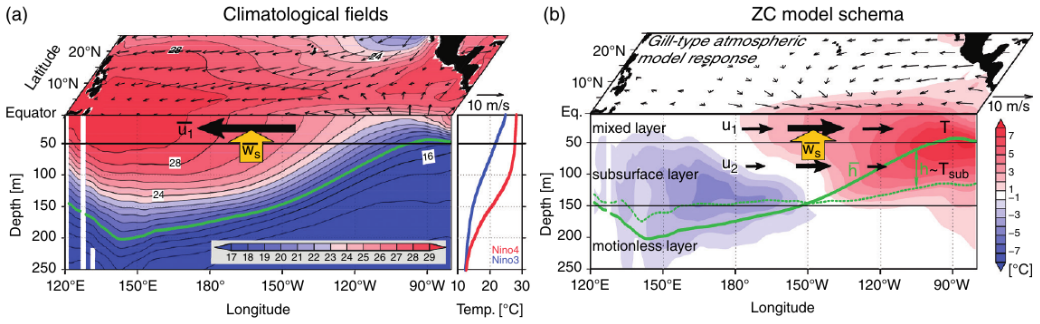


Figure 4: Schematic view of the Cane-Zebiak (CZ) model for the October 1997 - February 1998 El Niño event. (a) Prescribed climatological mean states for the model. Horizontal plane: Colored shadings portray mean SSTs, and vectors indicate surface winds. Vertical plane: mean vertical equatorial ocean temperatures and 18°C isotherm highlighted in green. Niño-4 and Niño-3 indices are depicted in the panel in red and blue, respectively. The bold black arrow displays the mean zonal current in the surface layer, and the orange arrow the mean upwelling. (b) Horizontal plane: Shadings indicate observed sea surface temperature anomalies. Vectors show the simulated Gill-Matsuno wind response. Vertical plane: Shading illustrates observed ocean temperature anomalies along the equator, and the green dashed line is the simulated thermocline depth. Thick black arrows represent the simulated ocean current anomalies in the surface and subsurface layers. Figure taken from Jin et al. (2020).

wind response of the atmosphere model in the next iteration. This ocean-atmosphere interaction is the key feature that drives the CZ model and the ENSO simulation. The domain of the CZ model is the tropical Pacific region from 124°E-80°W and 29°S-29°N. The model has a time step of 10 days between each iteration and assumes a year of 360 days with equally long months. A schematic view of the individual components can be seen in Figure 4. The complete details can be found in Zebiak and Cane (1987), and further information on ENSO modeling can be found in Jin et al. (2020).

Our implementation of the CZ model is motivated by Xie and Jin (2018) based on the adaptations made by Bejarano and Jin (2008). This model was originally modified to generate long synthetic runs (1000 years long) and to statistically analyze the emerging properties of the produced ENSO cycles. It removes the annual cycle of the prescribed climatological values and instead uses the same mean states for the entire year. Furthermore, the model is not spun up with observational data. Hence, this model cannot be used to make real-world forecasts. However, it constitutes a sophisticated artificial task for probabilistic parameter estimation. Additionally, our implementation is the first publicly available open-source version of the CZ model [2]. It is programmed in Python 3 instead of Fortran to allow for easy application of machine learning. We thank Ruihuang Xie for sharing their implementation for testing and comparison.

During the development of the CZ model, we found that the model is sensitive to the precision of the internal variables. For long runs, the dynamics of the model change when we increase the precision of the variables (e.g., change from 32 bit floating points to 64 or 128 bit floating points). This sensitivity could be a crucial finding as Fortran only supports 32 bit floats and is still often used in the climate science community. More information on this can be found in the appendix A1 - Precision Analysis.

3 Results

The results are structured as follows: The algorithms are implemented on a simple linear problem and an oscillatory problem to display their individual characteristics. Then, the algorithms are applied to the Lorenz'63 model, a simple chaotic system often used in atmospheric sciences. Finally, the algorithms are deployed on synthetic ENSO data from our CZ model.

[2] Cane-Zebiak model, Python3 implementation: https://github.com/Backlash1337/cz-model

3.1 Linear Model

To compare CES and (S)NPE to an analytic solution, we estimate the parameters of a polynomial function of third degree given by $ax^3 + bx^2 + cx + d$. In the following we infer the four free parameters a, b, c, d. Even though the function itself is not linear, it is linear with respect to the parameters. Hence, the problem can be formulated as

$$y = G\theta + \eta, \tag{8}$$

with $G \in \mathbb{R}^{d \times p}$ and the observational noise $\eta \sim \mathcal{N}(0, \Sigma_y)$. Since the likelihood of this problem is Gaussian, this problem has the nice property that if we use a Gaussian prior, we can derive an analytic solution. The Gaussian posterior can be calculated in closed-form under the following conditions:

$$\begin{aligned} \eta &\sim \mathcal{N}(0, \Sigma_y) && \text{(Gaussian noise)} \\ p(\theta) &= \mathcal{N}(\theta; m_\theta, \Sigma_\theta) && \text{(Gaussian prior)} \\ p(y|\theta) &= \mathcal{N}(y; G\theta, \Sigma_y), && \text{(Gaussian likelihood)} \\ \text{where we use} \Sigma_y &= \sigma^2 \mathbb{1}. \end{aligned}$$

Then the evidence and posterior distributions are:

$$\begin{aligned} p(y) &= \mathcal{N}(y; Gm_\theta, \Sigma_y + G\Sigma_\theta G^T) \\ p(\theta|y) &= \mathcal{N}(\theta; m_{\theta|y}, \Sigma_{\theta|y}), \quad \text{where} \\ \Sigma_{\theta|y} &= \left[G^T \Sigma_y^{-1} G + \Sigma_\theta^{-1}\right]^{-1} \\ m_{\theta|y} &= \Sigma_{\theta|y} \left[G^T \Sigma_y^{-1} y + \Sigma_\theta^{-1} m_\theta\right] \end{aligned}$$

In our experiment the ground truth parameter is set to $\theta = [0.9, -0.5, -0.1, 0.3]^T$ for a, b, c, d, respectively. We use a prior centered at 0.0 with an identity matrix as covariance. For the observational noise, we set $\sigma = 0.1$. The polynomial is evaluated at the locations $[-1, -0.5, 0, 0.5, 1]$, so the forward map has the dimensions $G \in \mathbb{R}^{5 \times 4}$. We run the calibrate step of CES over ten iterations using the ensemble sizes 5, 10, 20, 30, 50 and 100. For NPE and SNPE, we train the neural networks with 50, 100, 1K, 10K, and 100K simulations. SNPE is trained over 4 rounds.

To evaluate their performance, we draw one million samples from the learned posterior distributions and the ground truth posterior distribution. We use two metrics to compare the marginal distributions of each parameter: (i) the Jensen-Shannon distance and (ii) a histogram overlap. The mean of each metric over all four marginal distributions was taken. Each experiment was repeated ten times to assess the variability between individual runs. The results are displayed in figure 5. The results using a different density estimator for (S)NPE can be found in appendix A2 - Linear Model.

The Jensen-Shannon (JS) distance is a metric between two probability distributions using the Kullback-Leibler (KL) divergence against a mean distribution. In contrast to the KL divergence, it is symmetric and always returns finite values. This makes it more suitable for comparison of empirical distributions where one distribution may have support where the other has no probability mass. This would result in infinite values when using the KL divergence. The JS distance is the square root of the JS divergence defined as

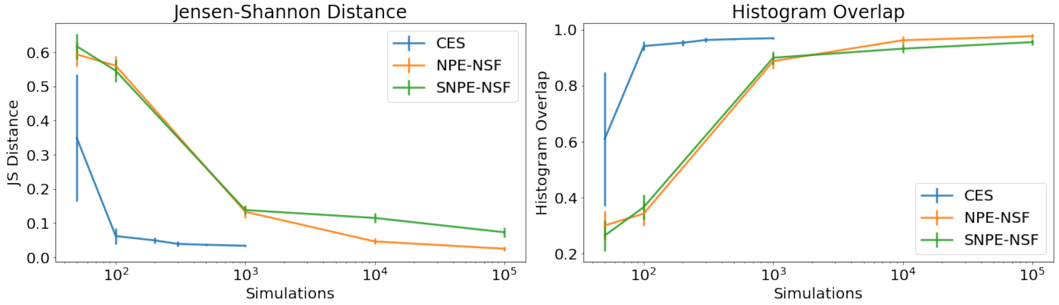
$$\text{JS}_{\text{dist}}(P, Q) = \sqrt{\text{JS}_{\text{div}}(P, Q)} = \sqrt{\frac{D_{KL}(P \| M) + D_{KL}(Q \| M)}{2}}, \tag{9}$$

where M is the mean of both distributions $M = \frac{1}{2}(P + Q)$ and $D_{KL}(P \| M)$ is the KL divergence between P and M. For the histogram overlap, the samples from all distributions are binned into 1000 bins. Then the histogram overlap of each marginal P to the respective ground truth marginal Q is computed as follows:

$$\text{Overlap}(P, Q) = \frac{\sum_{i=1}^{I} min(P_i, Q_i)}{\sum_{i=1}^{I} Q_i}, \tag{10}$$

where i iterates over the individual bins. The sum is normalized by the number of samples in Q. This describes the percentage of histogram overlap that P has with distribution Q.

Figure 5 shows the JS distance and histogram overlap over the number of simulations for CES and (S)NPE. Both algorithms converge towards the ground truth posterior distribution. CES requires



(a) Jensen-Shannon distance

(b) Histogram overlap

Figure 5: Two performance metrics on the linear problem for different numbers of simulations. (a) Jensen-Shannon distance between samples from the ground truth distribution and samples from the learned posterior distribution for different numbers of simulations. 0.0 is the best achievable distance. (b) Histogram overlap computed between an empirical histogram from samples of the ground truth distribution and histograms from the learned posterior distributions. 1.0 is the best possible score. All experiments are repeated ten times. The vertical bars indicate the standard deviation between the trials.

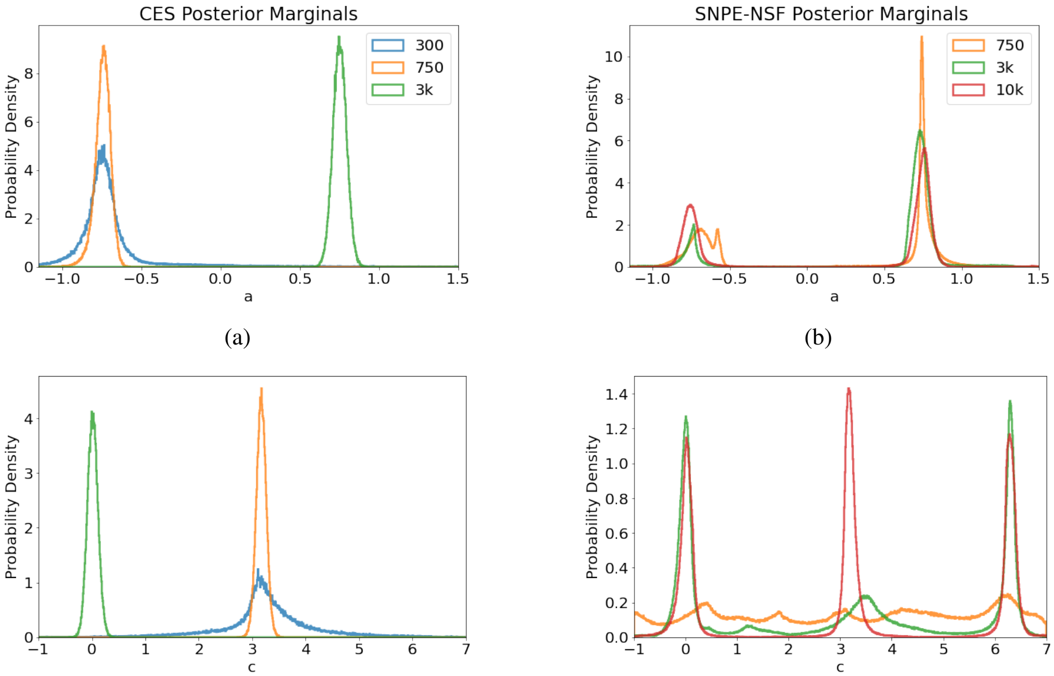
at least an order of magnitude fewer simulations than (S)NPE to converge to the true posterior. It plateaus with an ensemble size of 20 to 30 particles (e.g, 200 and 300 simulations). For the smallest ensemble, CES has the highest variance. However, the following ensemble sizes generally exhibit lower variance when compared to (S)NPE. It is important to note that this is the ideal problem for CES. For the linear case, EKI was proven to converge while the ensemble spread follows the variance of the posterior distribution (Schillings & Stuart, 2018). Hence, the GPs receive perfectly calibrated training data and only have to learn a simple linear relationship of the parameters.

3.2 Oscillatory Model

In this problem, we try to estimate the posterior distribution for the parameters of a sine function $a \cdot \sin(bx + c)$. This oscillatory problem is interesting because different parameter combinations can lead to the same observations (e.g., shifting the function by 2π, or inverting the magnitude a and shifting by π). Hence, the posterior is a multimodal distribution where the modes are far apart from each other. Thus, we investigate how well the algorithms are able to capture these characteristics.

The ground truth parameters are $a = 0.75, b = 1.2, c = 0$. As observations, we extract 11 equally spaced points between 0 and 3π and add observational Gaussian noise with a variance of 0.01 to each point individually. For the priors of CES, we used univariate Gaussian distributions centered at $0.85, 1.1, 3.0$ with standard deviations of $1, 1, 2$ for the parameters a, b, c, respectively. We ran the calibration step for 15 iterations, using the input-output pairs of the last 5 iterations for the emulation phase. For SNPE, we ran uniform priors spanning two standard deviations of the CES priors in either direction from their means. This uniform distribution includes 96% of the probability mass from the Gaussian prior used in CES. We trained SNPE over 4 rounds. We also ran the experiments using the normal distribution prior for (S)NPE (not shown), producing slightly worse results. Theoretically, changing the prior distribution affects the posterior distribution. However, the support region of those two priors chosen here is so similar that for the number of observations we are working with this effect is negligible. The learned marginal posterior distributions for the parameter a and c for different simulation numbers can be seen in Figure 6. The complete posterior pair plots can be found in appendix A3 - Oscillatory Model.

We increased the number of iterations for the calibration step to 15. Due to the multimodal landscape of the parameters a and b, the particles would sometimes jump between two modes, and it would take longer for them to collapse. In figure 6a and 6c, we see that CES always only reconstructs one posterior mode. This is because the particles in the calibration step can only collectively collapse to one individual location. Hence, the trained GPs will only be a good approximation around this mode. It needs around 750 simulations (e.g., ensemble size of 50 particles) to reconstruct one mode well. In contrast, SNPE using 750 simulations starts to recover the two modes of the parameter a but



(a)

(b)

(c)

(d)

Figure 6: Marginal posterior distributions for the parameters a and c of the oscillatory model $a\sin(bx + c)$ for CES (a) and (c) and SNPE (b) and (d), respectively.

not yet the structure of the parameter c, as seen in figure 6b and 6d. For 10K simulations, SNPE is able to reconstruct all modes perfectly. We also applied NPE to this problem (not shown), but it required even more simulations for similar reconstruction quality. Since the modes are so far apart, the sequential approach increases the sampling efficiency notably for this problem.

3.3 Lorenz ’63 Model

The Lorenz system is a set of three parameterized ordinary differential equations describing the chaotic evolution of a particle in three-dimensional space. The equations were published in 1963 by Edward Lorenz to model atmospheric convection (Lorenz, 1963). It was the first example to depict how tiny perturbations of the initial state can lead to vastly diverse developments of the system (i.e., the well-known *butterfly effect*). The three equations are given by:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{11}$$

The Lorenz’63 system is very sensitive to the parameters. Even slightly different parameters can produce drastically different trajectories of the particle. Additionally, we want to investigate the performance of both algorithms when the observations are extracted from a time series.

In our experiments, we start from the initial state $x_0 = 0, y_0 = 1, z_0 = 1.05$ and fix $\sigma = 10$. The ground truth parameters are $\beta = 2.667, \rho = 28$, producing the ’butterfly’ attractor as shown by the ground truth trajectories in Figure 7. We ran the system for 360 time steps with an integration step of 0.01. We apply observational noise $\eta \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.1$ to each point individually on the entire rollout. For the summary statistics, we followed the approach of Cleary et al. (2021). A function $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}^9$ is applied to compute the second moments and the combinations of all first moments at each timestep:

$$\varphi(x, y, z) = (x, y, z, x^2, y^2, z^2, xy, xz, yz) \tag{12}$$

We then use time windows of 10 time steps to compute one mean value, ignoring the simulation's initial 30 time steps as they are almost identical among the trajectories. Hence, the final observation vector consists of 33 points.

For the CES prior, we used a Gaussian distribution centered at 2.5 and 27.5 with a standard deviation of 1 for the parameters β and ρ, respectively. Due to stability reasons, we had to choose a bounded prior for (S)NPE as the simulations of parameters from the tails of the normal distribution would diverge and lead to numerical issues. Hence, we chose a uniform prior for (S)NPE spanning from 1.0 to 4.0 for β and from 26.0 to 29.0 for ρ. This includes around 86.6% of the probability mass of the Gaussian prior used in CES. This issue did not arise for CES, as the few initial ensemble members are unlikely to be located in those low probability regions. We ran each algorithm for 2K simulations: i) CES over 10 iterations with an ensemble of 200 members. ii) NPE with all parameters directly drawn from the prior. iii) SNPE trained over 4 rounds. For the emulation phase in CES, we only used the input-output pairs obtained from the last iteration of the calibration step. The results can be seen in Figure 7.

Due to the sensitivity of this problem, CES required a relatively big ensemble (around 50 to 100 particles) to collapse reliably to the correct posterior region. CES learned a narrow and spiky posterior distribution, indicating that the GPs eventually struggled to emulate the model smoothly. The posterior predictives of CES show no divergence at all from the ground truth trajectory. On the other hand, (S)NPE learned smoother and broader posterior distributions, which contain the ground truth parameter. Two of the posterior predictives of NPE depart from the ground truth trajectory, whereas the posterior predictives of SNPE closely resemble the ground truth trajectory. In general, all three algorithms managed to recover the ground truth parameter from broad prior distributions.

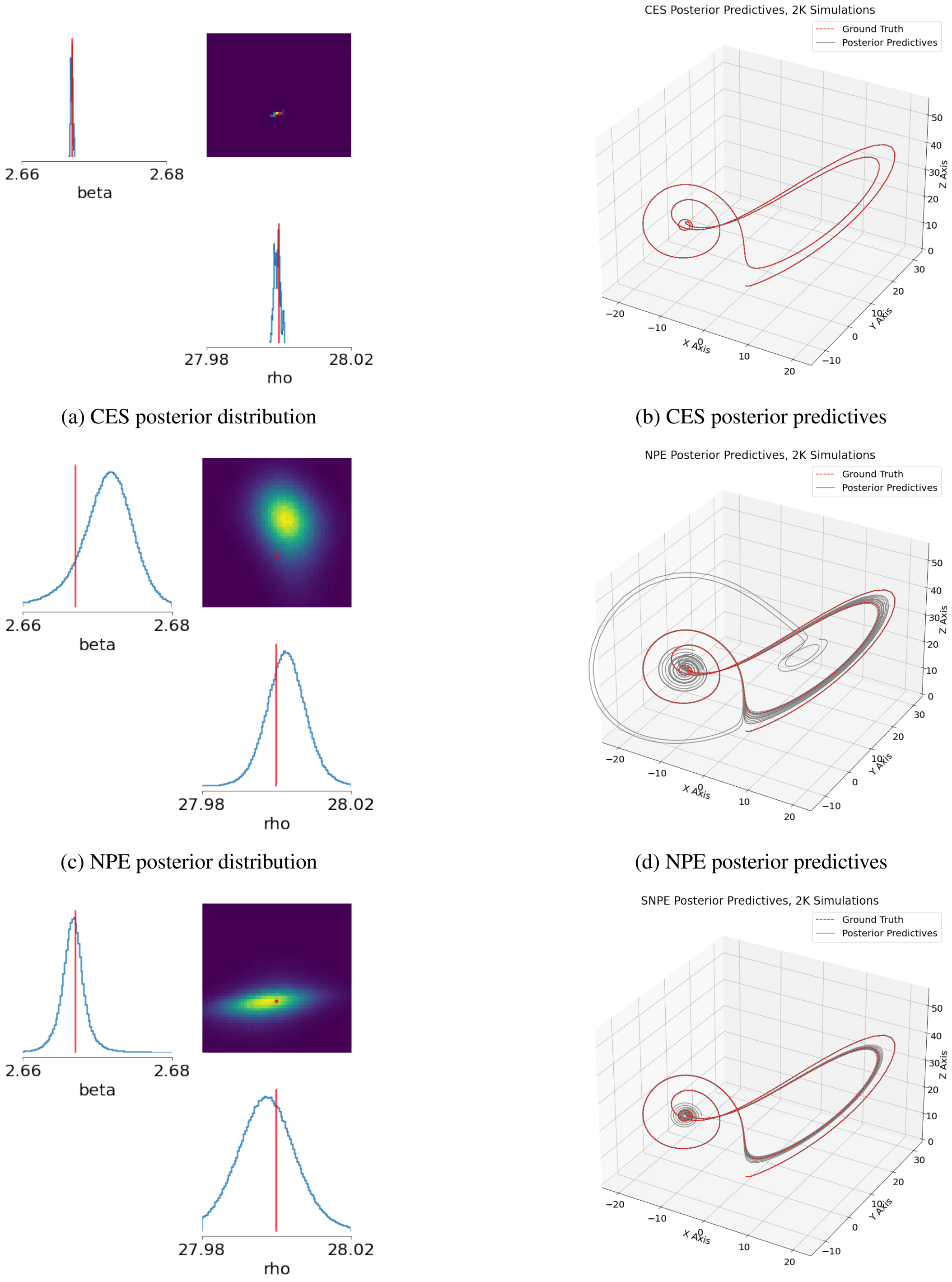
3.4 Cane-Zebiak Model

We ran our CZ model for 20 years, which produces an output of its internal variables every 10 days (e.g., SSTAs, winds, currents, thermocline depth). This produces 30 × 34 fields over 720 time steps. We use the Niño-3.4 time series as it is used to classify ENSO and condenses the dynamics of the CZ model. We add observational Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.1$ to each of the 720 points individually. For the observation vector, we then construct 36 non-overlapping windows spanning 20 time steps and compute the average value for each window. This results in an observation vector of length 36. We ran tests (not shown) to see the effects of different window sizes. Generally, observation vectors of a length of 30 to 80 performed robustly on the task.

In our experiment, we try to recover the Gam2 parameter, which is the most sensitive parameter in the model according to studies from Zhao et al. (2019) and Gao et al. (2021). This parameter controls the strength of anomalous upwelling in the SST update and hence has a fundamental impact on the feedback of the system. The ground truth value of this parameter is 0.75.

For CES, we used a Gaussian prior centered at 0.75 with a standard deviation of 0.15. We ran the calibration step for 10 iterations using ensemble sizes of 20, 30, 50, 100 and 200 particles. For the emulation stage, we took the input-output pairs from the last iteration. For NPE, we used a uniform prior spanning from 0.4 to 1.05, covering 96% of the probability mass of the CES prior. We ran NPE for 500, 1K, 5K, 10K, 20K, 30K and 50K simulations. Each experiment configuration was repeated for 10 trials each producing one posterior distribution. We compute the maximum-a-posteriori (MAP) estimate of each distribution and average them across the 10 trials. Additionally, we extract the smallest and largest MAP estimates among the trials. We repeat the same procedure for the standard deviations (STD) of all the distributions. The results can be seen in Figure 8a and 8b. An example of the 10 posterior distributions for 1K simulations can be seen in Figure 8c for CES and 8d for NPE. Finally, we took the average MAP estimate from the experiments and used them in our ZC model to show if they could reconstruct the same dynamic as the ground truth parameter. Those predictives are shown in Figure 9. We conducted the same procedure for the average of the means of the posterior distributions leading to slightly worse predictives. We also repeated those experiments for SNPE trained over 4 rounds and for CES using the input-output pairs of the last three iterations. Those results can be found in appendix A4 - Cane-Zebiak Results.

In Figure 8a, we see that both algorithms managed to converge to the ground truth parameter. However, when we look at the smallest and largest standard deviations in Figure 8b, we see that CES varies more than NPE. For CES, the smallest standard deviation is always close to 0.0, indicating that at least



(a) CES posterior distribution

(b) CES posterior predictives

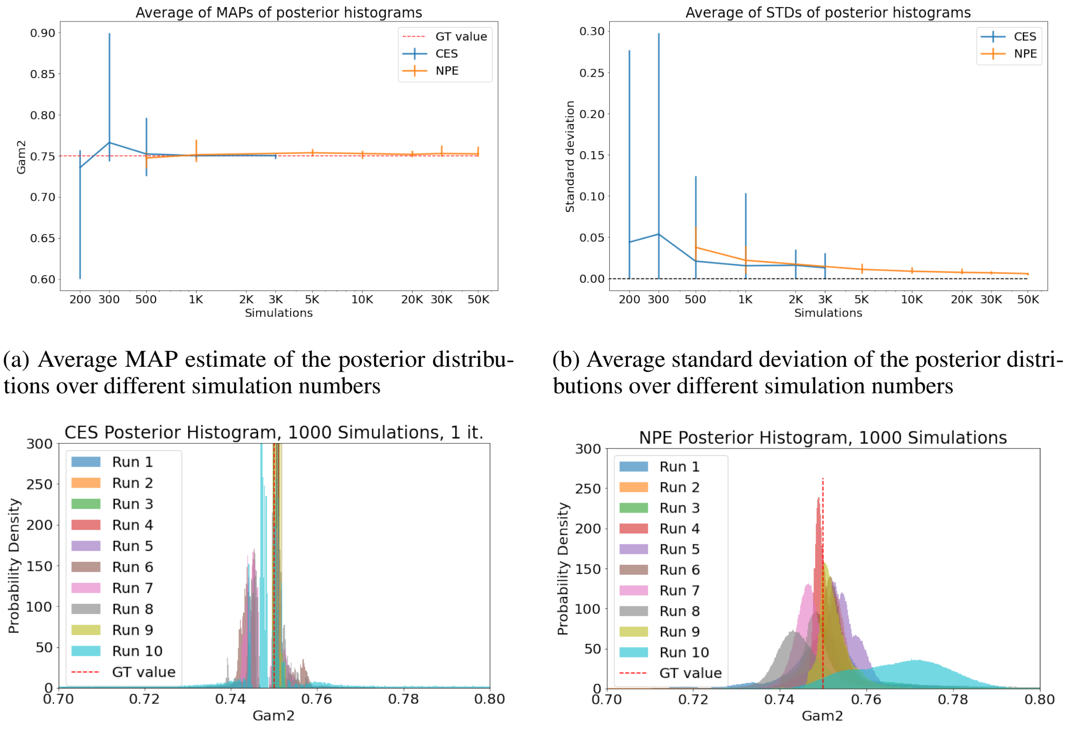
(c) NPE posterior distribution

(d) NPE posterior predictives

(e) SNPE posterior distribution

(f) SNPE posterior predictives

Figure 7: Left column: Marginal and joint posterior distributions for (a) CES, (c) NPE, and (e) SNPE. Right column: 20 posterior predictives for (b) CES, (d) NPE, and (f) SNPE. The posterior predictives are constructed by randomly drawing 20 parameters β and ρ from the learned posterior distributions and using them in the Lorenz ’63 model.



(a) Average MAP estimate of the posterior distributions over different simulation numbers

(b) Average standard deviation of the posterior distributions over different simulation numbers

(c) CES posterior distributions using 1K simulations

(d) NPE posterior distributions using 1K simulations

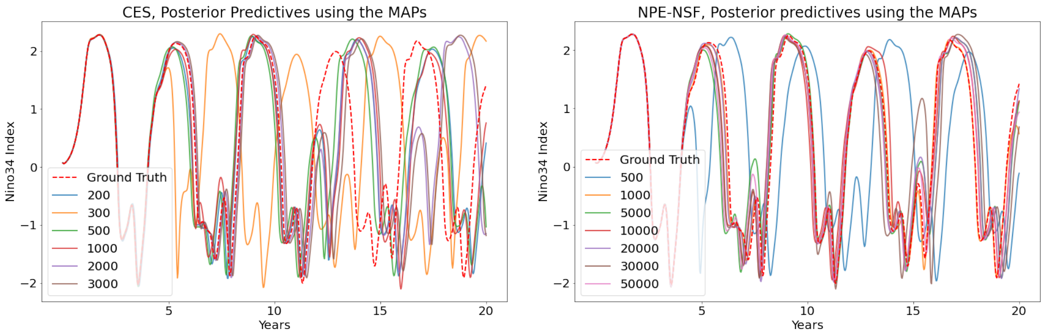
Figure 8: (a) Average MAP and (b) STD of the posterior distributions. The error bars indicate the minimum and maximum MAP and STD of each of the 10 runs. Bottom row: Learned posterior distributions for (c) CES and (d) NPE for 1K simulations.

one posterior distribution is close to a point estimate in these runs. The large error bars imply that the learned posterior distributions differ strongly between the individual runs of CES. In contrast, the error bars for NPE are much smaller and continuously decrease for more simulations. This illustrates that the NPE posterior distributions coincide more across the 10 runs. This deduction is confirmed in the posterior histograms shown in the Figure 8c and 8d for an example of 1K simulations. In general, NPE produced smoother and more well-formed posterior distributions. For CES, the GPs presumably struggle to emulate the coupled dynamics of the model smoothly. The calibration step required at least 100 ensemble members to collapse to the correct region consistently with a sufficient spread of the particles. This explains why the MAP estimates for 200, 300, and 500 simulations (e.g., ensembles of size 20, 30, and 50) have such a big spread. Plots of the evolution of the particles can be found in appendix A5 - Cane-Zebiak Calibration. When we look at the posterior predictives in Figure 9, we see that CES managed to reproduce the dynamics for around 12 years before the time series diverged. Except for 500 simulations, the NPE MAP predictives reconstruct the entire time series well. NPE required more simulations but created more credible posterior distributions, and the posterior predictives resemble the ground truth time series better.

4 Discussion

4.1 Evaluation

The CES approach and the (S)NPE algorithm can be used to learn the posterior distributions of parameters of black box models. In CES, GPs are used to emulate the model and construct an accessible likelihood. This proxy model is then used in MCMC sampling to draw samples that follow the approximate posterior distribution. On the hand, (S)NPE uses a neural density estimator to learn the posterior distribution directly from the relationship between parameters and their simulated data and does not need an additional MCMC sampling step. The MCMC step in CES introduces a



(a)

(b)

Figure 9: Posterior predicitives using the average MAP estimate of different simulation numbers for (a) CES and (b) NPE.

potentially erroneous sampling in comparison to SNPE. Furthermore, (S)NPE is more expressive as any prior can be used, and the neural density estimator can learn any posterior distribution within its capacity In contrast, CES is restricted to Gaussian priors and generally only recovers one mode of the posterior distribution.

Throughout the experiments, we have seen that CES requires fewer simulations, sometimes up to an order of magnitude, compared to (S)NPE. The EKI algorithm moves the particles robustly to the region of high posterior mass when a sufficient ensemble size is chosen. For too small ensemble sizes, EKI would converge to an incorrect region or become instable and jump around in each iteration. For the linear and oscillatory model, this was around 20 to 50 members. For the more sophisticated Lorenz '63 and CZ model, an ensemble of at least 100 particles had to be used to collapse to the correct region consistently. The GPs used in the emulation phase could emulate the linear and oscillatory model well. Hence, the MCMC sampling produced smooth posterior distributions. For the time series problems, the GPs struggled to emulate the correlated outputs, which led to sharp and rough posteriors distributions. The posterior estimates of CES varied more between trials, indicating that the results are sensitive to the data points generated during the calibration step. Using more input-output pairs for training the GPs does not necessarily resolve this issue as it can lead to overfitting. Moreover, using too many input-output pairs from the CZ problem sometimes led to numerical instabilities in the solvers. While in theory, the GPs could emulate a model with a multimodal posterior distribution, in practice, this never happened as the training data is focused around only one optimum, and MCMC random walk sampling generally fails to jump between modes when they are too far apart. This is a fundamental issue of MCMC sampling but potentially more sophisticated sampling methods like Slice Sampling (Neal, 2003) or Hamiltonian Monte Carlo (Duane et al., 1987) could improve the results of CES.

In contrast, (S)NPE worked robustly across all tasks when using enough simulations. It managed to recover multimodal distributions and was able to learn smooth posteriors from time series data. When increasing the number of simulations, the posterior distributions become more consistent among the trials. This can be seen in the standard deviation plot in Figure 8b on the CZ problem. In contrast to the benchmark by Lueckmann et al. (2021), SNPE did not generally outperform NPE in our work. It would often construct narrower posterior distributions which varied more among multiple trials compared to NPE. Eventually, for some tasks, fewer rounds could have performed better.

Since the simulators of our problems can be run fairly quickly, most time was spent in the MCMC sampling for CES and training of the neural network for (S)NPE. The algorithms would run from a few minutes on the linear model to at most four hours for the CZ model using a single CPU on the cluster. However, a direct comparison of the training times would be unfair, as the CES framework is implemented in Julia and the SBI library in Python, and the focus of this work was an algorithmic comparison. Both toolboxes are still being actively developed [3].

[3] CES framework: https://github.com/CliMA/CalibrateEmulateSample.jl
SBI framework: https://github.com/mackelab/sbi

	CES	SBI
Prior	Gaussian prior (may be bounded)	Any prior which can be sampled from and be evaluated
Posterior	Gaussian processes accurate around EKI particles	Any posterior distribution that can be constructed by the density estimator
Learning	Fast particle updates for given observation	Slower neural-network based, amortized
Noise	Gaussian noise with known variance	Gaussian noise
Output	GPs trained on collapsed particles	Trained neural density estimator

Table 1: Summary of the key characteristics of CES and SBI.

A summary of the key characteristics of CES and SBI can be found in Table 1.

4.2 Extensions

A crucial aspect of both algorithms is which features of the data are presented as observations, e.g., the choice of the summary statistics. This is especially important for time series problems since the points are highly correlated. While working with the Lorenz '63 model, it became clear that a well-crafted summary statistic offers a significantly better learning signal. The CZ model produces a broad range of output fields (e.g., SSTA, winds, currents, thermocline depth). In our work, we used averaged windows of the Niño-3.4 index as observations. The Niño-3.4 index is a decent summary statistic used to classify El Niño and La Niña events. However, this work did not focus on a deep examination of summary statistics, and more sophisticated summary statistics certainly exist. The SBI toolbox, for example, allows for the use of neural networks to learn summary statistics from data. Well-crafted summary statistics could reduce the number of simulations necessary to achieve accurate posterior estimates. Crafting sophisticated summary statistics for artificial CZ model data for ENSO could be a direction for future research. These statistics could then also be applied to real-world observational data and eventually improve the parameters of forecast models.

In our experiments on the CZ model, we only tried to infer the Gam2 parameter. For a similar CZ model, Gao et al. (2021) used an ensemble Kalman filter to infer 6 parameters simultaneously. However, they only inferred point estimates and had no probabilistic interpretation. A natural extension of our work would be to expand the algorithms to estimate multiple parameters and compare their performance. Presumably, the advantage in terms of required simulations of CES would increase as EKI was also shown to work well in high-dimensional spaces. On the other hand, the GPs struggled to emulate the CZ model in this work, which potentially worsens in a higher dimensional space.

The hyperparameters of both approaches have not been extensively hypertuned to the problems. Both frameworks offer a broad range of options that have only been lightly explored. The calibration step can be run using other particle update algorithms like the EKS and UKI. For the GPs, different kernel functions can be used. Similarly, SBI provides entirely different inference approaches with SNLE and SNRE. All of these algorithms have their strengths, limitations, and biases. So a comparison of the SBI algorithms on the CZ model could also be interesting.

Evaluating the posterior accuracy is still an open topic for models with an intractable likelihood. Approximate Bayesian Computation (ABC) methods generally require a lot more simulations than SBI and scale badly to high dimensional spaces due to the rejection process. However, our CZ model still allows hundreds of thousands of runs in the order of days on a cluster. Hence, ABC methods could be used to construct another comparison against CES and SBI. Another option is to use Simulation-Based Calibration (SBC) (Talts et al., 2018). SBC allows us to check if our posterior estimates are biased, overdispersed, or underdispersed. These methods could help to make a more quantitative comparison between CES and SBI.

Lastly and most importantly, both approaches have different strengths and weaknesses and are not necessarily mutually exclusive. The SBI methods can be run over multiple rounds to improve the sampling efficiency and to prevent drawing too many parameters from low posterior probability regions. In our experiments, SNPE worked well for some problems but did not consistently outperform NPE. When the objective is to estimate only one posterior mode, ensemble Kalman inversion could be used as a preprocessing step to construct a new prior for (S)NPE. In our experiments, EKI usually converged within at most 1000 simulations. The generated input-output pairs could be added to the training data, and the span of the final ensemble could be utilized to set up a prior in the region of high posterior mass. Hence, the neural density estimator would only be trained in a small region. Due to the fast convergence of EKI this could improve the sampling efficiency compared to running the density estimators sequentially.

5 Conclusion

We implemented the first open-source implementation of a modified Cane-Zebiak model for El Niño-Southern Oscillation simulations in Python 3. We compared two state-of-the-art probabilistic parameter estimation algorithms from different scientific fields, namely SBI used in neuroscience and CES introduced in climate science. We applied them to various toy problems and for the first time to synthetic data from the Cane-Zebiak climate model. We evaluated their performance and investigated the learned posterior distributions. Furthermore, we emphasized the strengths and weaknesses of both algorithms in practice. We have shown that CES performs well when estimating unimodal distributions and the observations are not extracted from time series data. On the other hand, (S)NPE requires more simulations but performs robustly across all tasks, including complex time series problems, and is more expressive.

Acknowledgements

I would like to thank my family for always supporting me throughout the entirety of my studies, both financially and emotionally. I would like to show special gratitude to Jakob Schlör who has been a fantastic supervisor, giving valuable advice and always taking his time for thorough discussions. Thanks to Ingeborg Wenger for proof reading and comments on the thesis. Special thanks to Dr. Bedartha Goswami to give me the opportunity to work in his group on such a fascinating topic and to Prof. Dr. Jakob Macke for co-supervising my thesis. I also thank Ruihuang Xie for sharing their implementation of the CZ model. I thank the developers of `sbi` (Tejero-Cantero et al., 2020), `CES` (CliMA, 2022), `matplotlib` (Hunter, 2007) for providing essential toolboxes for my thesis.

References

Bejarano, L., & Jin, F.-F. (2008). Coexistence of equatorial coupled modes of enso. *Journal of Climate*, *21*(12), 3051–3067.

Bishop, C. M. (1994). Mixture density networks.

Bjerknes, J. (1969). Atmospheric teleconnections from the equatorial pacific. *Monthly weather review*, *97*(3), 163–172.

Cai, W., Ng, B., Wang, G., Santoso, A., Wu, L., & Yang, K. (2022). Increased enso sea surface temperature variability under four ipcc emission scenarios. *Nature Climate Change*, *12*(3), 228–231.

Cai, W., Santoso, A., Collins, M., Dewitte, B., Karamperidou, C., Kug, J.-S., Lengaigne, M., McPhaden, M. J., Stuecker, M. F., Taschetto, A. S., et al. (2021). Changing el niño–southern oscillation in a warming climate. *Nature Reviews Earth & Environment*, *2*(9), 628–644.

Chen, D., Cane, M. A., Kaplan, A., Zebiak, S. E., & Huang, D. (2004). Predictability of el niño over the past 148 years. *Nature*, *428*(6984), 733–736.

Chen, D., Cane, M. A., Zebiak, S. E., Canizares, R., & Kaplan, A. (2000). Bias correction of an ocean-atmosphere coupled model. *Geophysical Research Letters*, *27*(16), 2585–2588.

Chen, D., Cane, M. A., Zebiak, S. E., & Kaplan, A. (1998). The impact of sea level data assimilation on the lamont model prediction of the 1997/98 el nino. *Geophysical research letters*, *25*(15), 2837–2840.

Chen, D., Zebiak, S. E., Busalacchi, A. J., & Cane, M. A. (1995). An improved procedure for ei nino forecasting: Implications for predictability. *Science*, *269*(5231), 1699–1702.

Cleary, E., Garbuno-Inigo, A., Lan, S., Schneider, T., & Stuart, A. M. (2021). Calibrate, emulate, sample. *Journal of Computational Physics*, *424*, 109716.

CliMA. (2022). *Calibrateemulatesample.jl*. Retrieved July 5, 2022, from https://clima.github.io/CalibrateEmulateSample.jl/dev/

Craig, P. S., Goldstein, M., Seheult, A. H., & Smith, J. A. (1997). Pressure matching for hydrocarbon reservoirs: A case study in the use of bayes linear strategies for large computer experiments. *Case studies in bayesian statistics* (pp. 37–93). Springer.

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

Dilley, M., & Heyman, B. N. (1995). Enso and disaster: Droughts, floods and el niño/southern oscillation warm events. *Disasters*, *19*(3), 181–193.

Duan, W., & Wei, C. (2013). The 'spring predictability barrier'for enso predictions and its possible mechanism: Results from a fully coupled model. *International Journal of Climatology*, *33*(5), 1280–1292.

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, *195*(2), 216–222.

Dunbar, O. R., Garbuno-Inigo, A., Schneider, T., & Stuart, A. M. (2021). Calibration and uncertainty quantification of convective parameters in an idealized gcm. *Journal of Advances in Modeling Earth Systems*, *13*(9), e2020MS002454.

Durkan, C., Bekasov, A., Murray, I., & Papamakarios, G. (2019). Neural spline flows. *Advances in neural information processing systems*, *32*.

Durkan, C., Murray, I., & Papamakarios, G. (2020). On contrastive learning for likelihood-free inference. *International Conference on Machine Learning*, 2771–2781.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, *99*(C5), 10143–10162.

Gao, Y., Tang, Y., Song, X., & Shen, Z. (2021). Parameter estimation based on a local ensemble transform kalman filter applied to el niño–southern oscillation ensemble prediction. *Remote Sensing*, *13*(19), 3923.

Garbuno-Inigo, A., Hoffmann, F., Li, W., & Stuart, A. M. (2020). Interacting langevin diffusions: Gradient structure and ensemble kalman sampler. *SIAM Journal on Applied Dynamical Systems*, *19*(1), 412–441.

Gill, A. E. (1980). Some simple solutions for heat-induced tropical circulation. *Quarterly Journal of the Royal Meteorological Society*, *106*(449), 447–462.

Greenberg, D., Nonnenmacher, M., & Macke, J. (2019). Automatic posterior transformation for likelihood-free inference. *International Conference on Machine Learning*, 2404–2414.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

Hermans, J., Begy, V., & Louppe, G. (1903). Likelihood-free mcmc with amortized approximate likelihood ratios. i.

Huang, D. Z., Schneider, T., & Stuart, A. M. (2022). Iterated kalman methodology for inverse problems. *Journal of Computational Physics*, *463*, 111262.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Iglesias, M. A., Law, K. J., & Stuart, A. M. (2013). Ensemble kalman methods for inverse problems. *Inverse Problems*, *29*(4), 045001.

IRI. (2022). *Iri enso forecasts*. Retrieved November 5, 2022, from https://iri.columbia.edu/our-expertise/climate/forecasts/enso/current/?enso_tab=enso-sst_table

Jin, F.-F., Chen, H.-C., Zhao, S., Hayashi, M., Karamperidou, C., Stuecker, M. F., Xie, R., & Geng, L. (2020). Simple enso models. *El Niño Southern Oscillation in a changing climate*, 119–151.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671–680.

Kobyzev, I., Prince, S. J., & Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, *43*(11), 3964–3979.

Latif, M., & Keenlyside, N. S. (2009). El ni&#xf1;o/southern oscillation response to global warming. *Proceedings of the National Academy of Sciences*, *106*(49), 20578–20583. https://doi.org/10.1073/pnas.0710860105

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of atmospheric sciences*, *20*(2), 130–141.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., & Macke, J. (2021). Benchmarking simulation-based inference. *International Conference on Artificial Intelligence and Statistics*, 343–351.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., & Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, *30*.

McGregor, S., Cassou, C., Kosaka, Y., & Phillips, A. S. (2022). Projected enso teleconnection changes in cmip6. *Geophysical Research Letters*, *49*(11), e2021GL097511.

McPhaden, M. J., Santoso, A., & Cai, W. (2020). *El niño southern oscillation in a changing climate* (Vol. 253). John Wiley & Sons.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087–1092.

NCEI. (2022). Us billion-dollar weather and climate disasters. https://doi.org/10.25921/stkw-7w73

Neal, R. M. (2003). Slice sampling. *The annals of statistics*, *31*(3), 705–767.

Papamakarios, G. (2019). Neural density estimation and likelihood-free inference. *arXiv preprint arXiv:1910.13233*.

Papamakarios, G., & Murray, I. (2016). Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, *29*.

Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, *30*.

Papamakarios, G., Sterratt, D., & Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *The 22nd International Conference on Artificial Intelligence and Statistics*, 837–848.

Polyak, B. T. (2007). Newton's method and its use in optimization. *European Journal of Operational Research*, *181*(3), 1086–1096.

Price, L. F., Drovandi, C. C., Lee, A., & Nott, D. J. (2018). Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, *27*(1), 1–11.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. *Summer school on machine learning*, 63–71.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Schillings, C., & Stuart, A. M. (2018). Convergence analysis of ensemble kalman inversion: The linear, noisy case. *Applicable Analysis*, *97*(1), 107–123.

Sisson, S. A., Fan, Y., & Beaumont, M. A. (2018). Overview of abc. *Handbook of approximate bayesian computation* (pp. 3–54). Chapman; Hall/CRC.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*.

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

Troup, A. (1965). The 'southern oscillation'. *Quarterly Journal of the Royal Meteorological Society*, *91*(390), 490–506.

Ward, P. J., Jongman, B., Kummu, M., Dettinger, M. D., Sperna Weiland, F. C., & Winsemius, H. C. (2014). Strong influence of el niño southern oscillation on flood risk around the world. *Proceedings of the National Academy of Sciences*, *111*(44), 15659–15664.

Wright, A. H. (1991). Genetic algorithms for real parameter optimization. *Foundations of genetic algorithms* (pp. 205–218). Elsevier.

Wu, X., Han, G., Zhang, S., & Liu, Z. (2016). A study of the impact of parameter optimization on enso predictability with an intermediate coupled model. *Climate Dynamics*, *46*(3), 711–727.

Xie, R., & Jin, F.-F. (2018). Two leading enso modes and el niño types in the zebiak–cane model. *Journal of Climate*, *31*(5), 1943–1962.

Yang, Y., Hu, X., Liao, G., Cao, Q., Chen, S., Gao, H., & Wei, X. (2022). Improved enso and pdo prediction skill resulting from finer parameterization schemes in a cgcm. *Remote Sensing*, *14*(14), 3363.

Zebiak, S. E. (1986). Atmospheric convergence feedback in a simple model for el niño. *Monthly weather review*, *114*(7), 1263–1271.

Zebiak, S. E., & Cane, M. A. (1987). A model el niñ–southern oscillation. *Monthly Weather Review*, *115*(10), 2262–2278.

Zebiak, S. E. (1984). *Tropical atmosphere-ocean interaction and the el niño/southern oscillation phenomenon* (Doctoral dissertation). Massachusetts Institute of Technology.

Zhao, Y., Liu, Z., Zheng, F., & Jin, Y. (2019). Parameter optimization for real-world enso forecast in an intermediate coupled model. *Monthly Weather Review*, *147*(5), 1429–1445.

Appendix

A1 - Precision Analysis

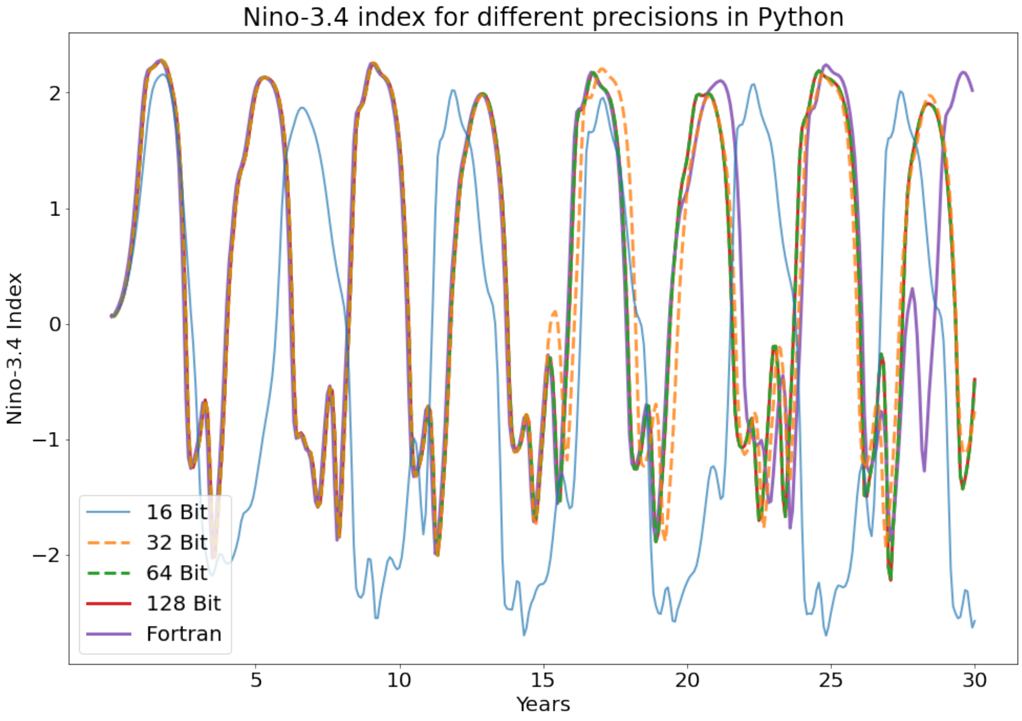
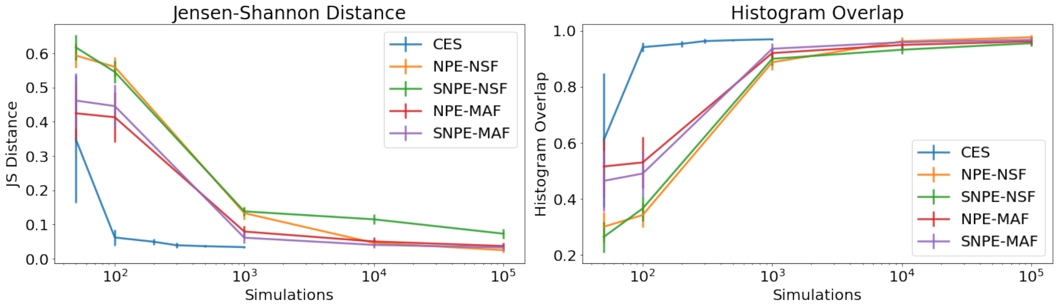


Figure 10: The Niño-3.4 time series for different floating point precisions in Python as well as the time series from the Fortran model.

All four lines in Figure 10 use the same Python code. The only difference is that NumPy arrays containing internal variables either use `np.float16`, `np.float32`, `np.float64`, or `np.float128`. Constant values and single float variables are still represented in the native 64-Bit of Python, which cannot be modified easily. In Figure 10, we see that the 16-Bit curve instantly diverges from the other curves. After around 15 years, the 32-Bit curve slightly diverges from the rest. Moreover, at around 20 years, the 64-Bit and 128-Bit time series depart from the Fortran series. Even though the values of the 64-Bit and the 128-Bit run do not match exactly, their difference is so tiny that it does not significantly influence the output of the model. This shows how infinitesimal disparities of the internal fields (due to different precision) can aggregate and change the model prediction over long runs, similar to the *butterfly effect*. However, this also suggests that when using a higher precision this effect does not appear or at least decelerated. Fortran uses 32-Bit for all its computations, and climate models today are still often implemented in this language due to historical reasons. Our finding could be a property that only our modified CZ model exhibits. However, it would be a relevant topic for future investigation as it could change how much we trust the predictions of future climate projections of models using only low precision variables.

A2 - Linear Model



(a) Jensen-Shannon distance

(b) Histogram overlap

Figure 11: The same plots as in Figure 5 but including results using MAFs as density estimators.

The diagrams in Figure 5 also show the results for using Masked Autoregressive Flows (MAFs) as density estimators for the linear problem. They even outperformed NSFs on this problem. Possibly, MAFs are better at reshaping a multivariate Gaussian into another multivariate Gaussian than NSFs. However, posterior distributions are rarely Gaussian for more complex problems. On the multimodal posterior of the oscillatory model, the NSFs performed much better. Here, the MAFs would require more simulations and often have artifacts of the transformation between the individual posterior modes. Hence, only NSFs have been used for the further experiments.

A3 - Oscillatory Model

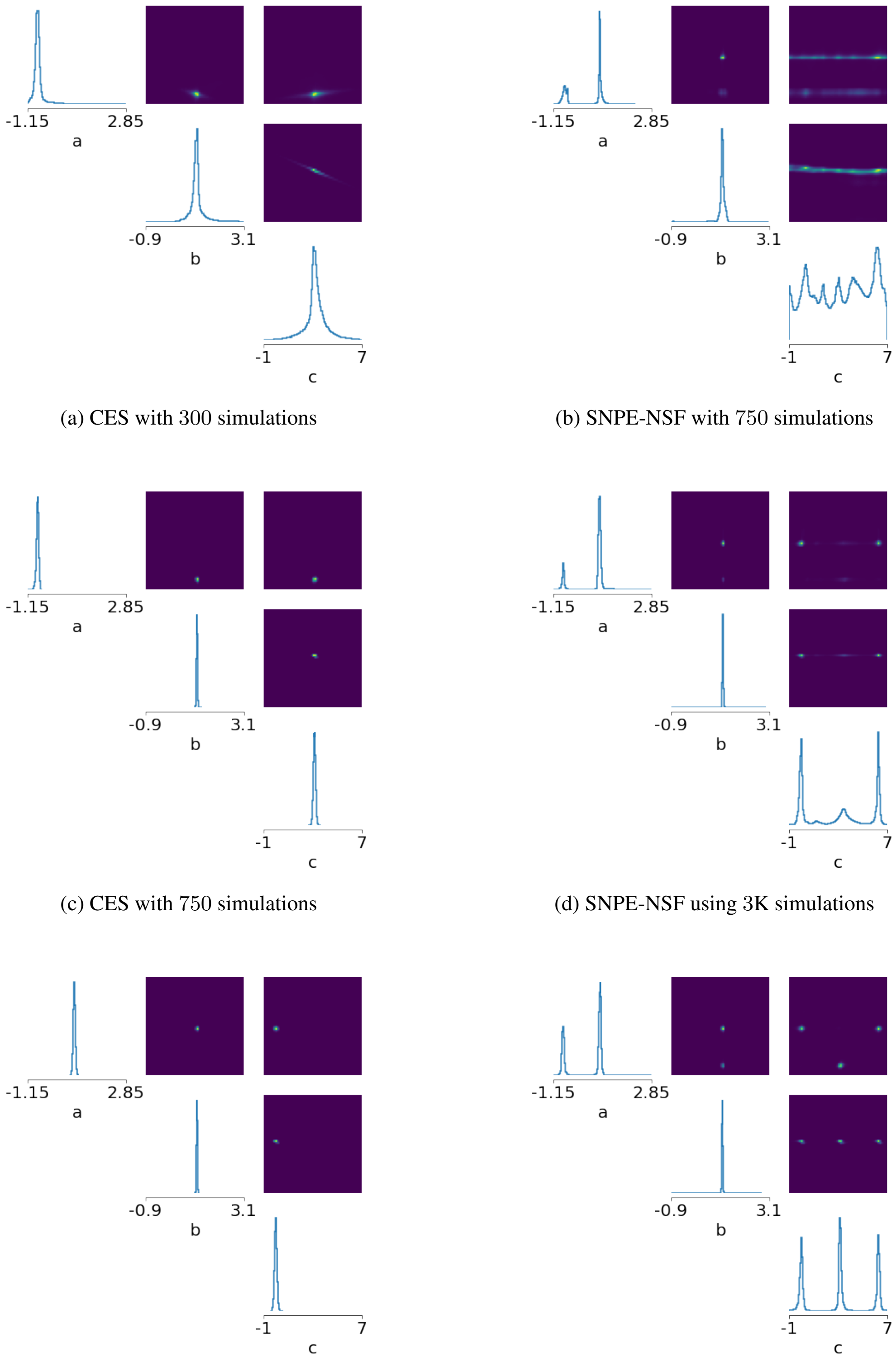
The posterior pair plots for all six runs are displayed in Figure 12. We see that CES using an ensemble of 50 and 200 ensembles was able to reconstruct one posterior mode. SNPE requires more simulations but is able to represent all distinct modes when using enough simulations.

A4 - Cane-Zebiak Results

Even though we increased the number of training points for the GPs, the posterior distributions are not smoother. This can be seen in the example distribution for 1K simulations in Figure 13c. For an ensemble of 50, the posterior distributions all collapsed to almost a point estimate, as indicated by the tiny error bars in Figure 13b. The posterior distributions of SNPE are sharper than the ones of NPE for the same number of simulations. The MAP estimates also consistently align with the ground truth value as shown in Figure 13a. However, the posterior distributions coincide less among the trials, which can be seen on the larger error bars in Figure 13b. The MAP posterior predictives of both algorithms can be found in Figure 14. Interestingly, their MAP estimates are closer to the ground truth value than the ones learned from NPE. Though, their posterior predictives represent the ground truth trajectory worse than the ones from NPE shown in Figure 9.

A5 - Cane-Zebiak Calibration

Examples of the evolution of particles during the calibration step of CES for the Cane-Zebiak model are shown in Figure 15. Each line corresponds to the evolution of one particle over time. The ground truth parameter should be contained within a sufficiently large ensemble spread for a proper calibration. We see that this is only achieved consistently for ensembles with at least 100 particles.



(a) CES with 300 simulations

(b) SNPE-NSF with 750 simulations

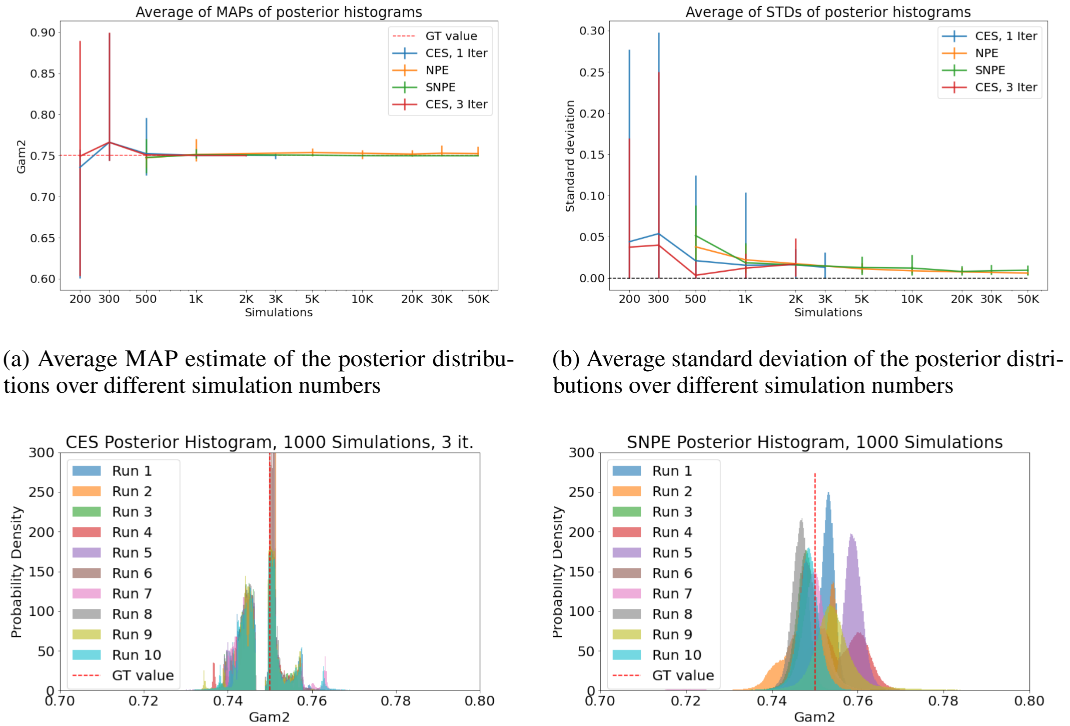
(c) CES with 750 simulations

(d) SNPE-NSF using 3K simulations

(e) CES with 3K simulations

(f) SNPE-NSF using 10K simulations

Figure 12: The complete posterior pair plots for the marginals displayed in Figure 6 for the oscillatory problem. Left column: CES. Right column: SNPE trained over 4 rounds.



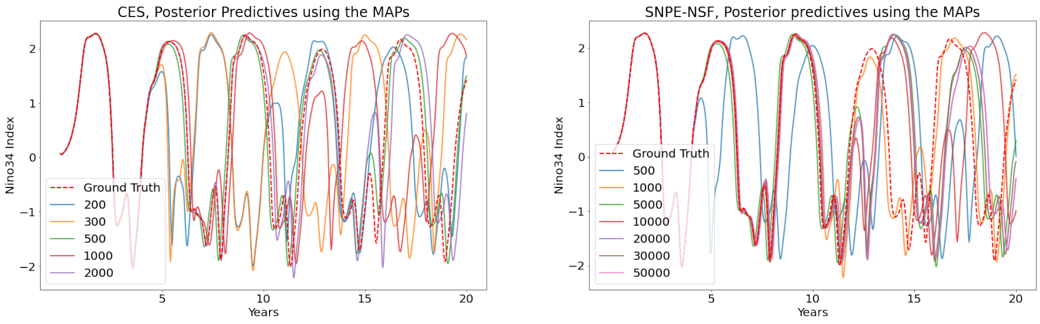
(a) Average MAP estimate of the posterior distributions over different simulation numbers

(b) Average standard deviation of the posterior distributions over different simulation numbers

(c) CES posterior distributions using 1K simulations

(d) SNPE posterior distributions using 1K simulations

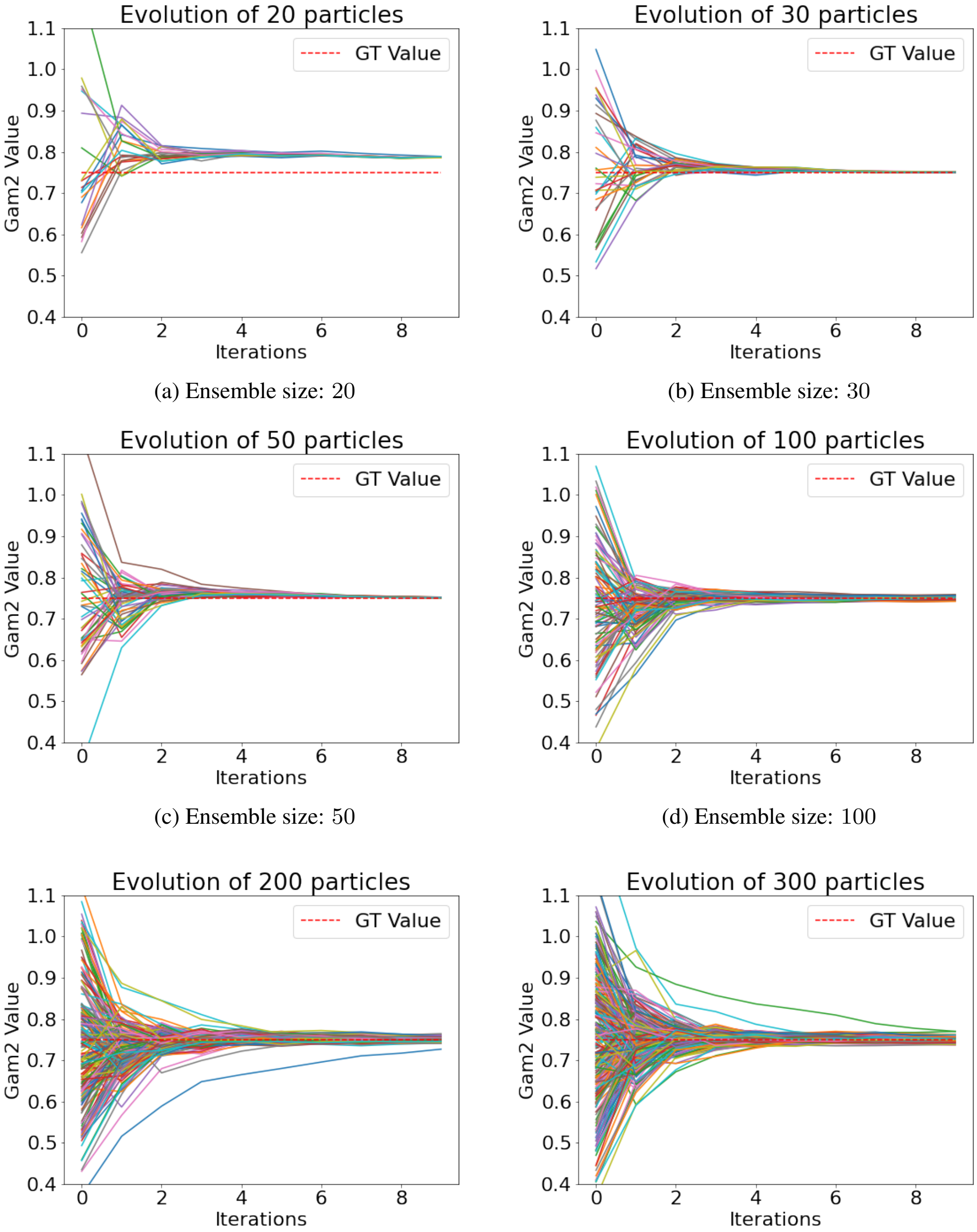
Figure 13: (a) Average MAP and (b) STD of the posterior distributions. The error bars indicate the minimum and maximum MAP and STD of each of the 10 runs. Example posterior distributions for (c) CES using input-output pairs of the last three iterations and (d) SNPE for 1K simulations trained over 4 rounds.



(a)

(b)

Figure 14: Posterior predicitives using the average MAP estimate of different simulation numbers for (a) CES using input-output pairs of the last three iterations and (b) SNPE for 1K simulations trained over 4 rounds.



(a) Ensemble size: 20

(b) Ensemble size: 30

(c) Ensemble size: 50

(d) Ensemble size: 100

(e) Ensemble size: 200

(f) Ensemble size: 300

Figure 15: Visualization of the evolution of particles for different ensemble sizes during the calibration step for the Cane-Zebiak model.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Benedict Röder

Benedict Röder (Matrikelnummer 3999707), August, 10, 2022