Sequentially Guided MCMC Proposals for Synthetic Likelihoods and Correlated Synthetic Likelihoods

Citation for the original published paper (version of record):
Picchini, U., Simola, U., Corander, J. (2023). Sequentially Guided MCMC Proposals for Synthetic Likelihoods and Correlated Synthetic
Likelihoods. Bayesian Analysis, 18(4): 1099-1129 . http://dx.doi.org/10.1214/22-BA1305

Bayesian Analysis (2023) **18**, Number 4, pp. 1099–1129

Sequentially Guided MCMC Proposals for Synthetic Likelihoods and Correlated Synthetic Likelihoods[*]

Umberto Picchini[†], Umberto Simola[‡], and Jukka Corander[§]

Abstract. Synthetic likelihood (SL) is a strategy for parameter inference when the likelihood function is analytically or computationally intractable. In SL, the likelihood function of the data is replaced by a multivariate Gaussian density over summary statistics of the data. SL requires simulation of many replicate datasets at every parameter value considered by a sampling algorithm, such as Markov chain Monte Carlo (MCMC), making the method computationally-intensive. We propose two strategies to alleviate the computational burden. First, we introduce an algorithm producing a proposal distribution that is sequentially tuned and made conditional to data, thus it rapidly *guides* the proposed parameters towards high posterior density regions. In our experiments, a small number of iterations of our algorithm is enough to rapidly locate high density regions, which we use to initialize one or several chains that make use of off-the-shelf adaptive MCMC methods. Our "guided" approach can also be potentially used with MCMC samplers for approximate Bayesian computation (ABC). Second, we exploit strategies borrowed from the correlated pseudo-marginal MCMC literature, to improve the chains mixing in a SL framework. Moreover, our methods enable inference for challenging case studies, when the posterior is multimodal and when the chain is initialised in low posterior probability regions of the parameter space, where standard samplers failed. To illustrate the advantages stemming from our framework we consider five benchmark examples, including estimation of parameters for a cosmological model and a stochastic model with highly non-Gaussian summary statistics.

Keywords: Bayesian inference, cosmological parameters, intractable likelihoods, likelihood-free.

1 Introduction

Synthetic likelihood (SL) is a methodology for parameter inference in stochastic models that do not admit a computationally tractable likelihood function. That is, similarly to approximate Bayesian computation (ABC, Sisson et al., 2018), SL only requires the

arXiv: 2004.04558

[*] UP is supported by the Swedish Research Council (Vetenskapsrådet 2019-03924) and the Chalmers AI Research Centre (CHAIR). JC was funded by the ERC grant no. 742158. US was funded by Academy of Finland grant no. 320182.

[†] Dept. Mathematical Sciences, Chalmers University of Technology and the University of Gothenburg, Sweden, picchini@chalmers.se

[‡] Department of Mathematics and Statistics, University of Helsinki, Finland, umberto.simola@helsinki.fi

[§] Department of Biostatistics, University of Oslo, Norway, jukka.corander@medisin.uio.no

https://doi.org/10.1214/22-BA1305

ability to generate synthetic datasets from a model simulator, and statistically relevant summary statistics of the data that capture parameter-dependent variation in an adequate manner. The price to pay for its flexibility is that SL can be computationally very intensive, since it is typically embedded into a Markov chain Monte Carlo (MCMC) framework, requiring the simulation of multiple (often hundreds or thousands) synthetic datasets at each proposed parameter. The goal of our work is twofold: (i) we introduce an algorithm that sequentially produces a proposal sampler that is made conditional to data and rapidly enables the identification of high-posterior-density regions, where to initialize MCMC chains using off-the-shelf methods; (ii) we introduce a way to increase the chains mixing, by tweaking methods that have been recently proposed in the correlated particle filters literature. Hence both strategies aim at reducing the computational cost to perform Bayesian inference via SL. We show that our approaches facilitate sampling when the chains are initialised at parameter values in regions of low posterior probability, a case where SL often struggles, see the case studies in Sections 6.2 and 6.3 where the Bayesian synthetic likelihoods (BSL) of Price et al. (2018) fail when using the adaptive MCMC proposal of Haario et al. (2001). For the case study in Section 6.3, having strongly non-Gaussian summary statistics, we show that even a BSL version robustified to non-Gaussian summaries fails to explore the posterior surface when initialized at challenging locations, while our proposal sampler is able to quickly converge towards the high-density region. Our proposal sampler can be beneficial with multimodal targets, to inform the researcher of the existence of multiple modes using a small number of iterations, see Section 6.4. In addition, in Section 5 we inform the reader that for challenging problems where it is difficult to locate appropriate starting parameters, an alternative to our method is Bayesian optimization, which can be efficiently used for kickstarting SL-based posterior sampling (Gutmann and Corander, 2016), and is facilitated by the open-source `ELFI` software (Engine for Likelihood-Free Inference, Lintusaari et al., 2018).

SL is described in detail in Section 2, but here we first review its features with relevant references to the literature. SL was first proposed in Wood (2010) and replaces the analytically intractable data likelihood $p(y|\theta)$ for observed data y with the joint density of a set of summary statistics of the data $s := T(y)$. Here $T(\cdot)$ is a function of the data that has to be specified by the analyst and that can be evaluated for input y, or simulated data y^*. The SL approach is characterized by the assumption that s has a multivariate normal distribution $s \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$ with unknown mean μ_θ and covariance matrix Σ_θ. These can be estimated via Monte Carlo simulations of size M to obtain estimators $\hat{\mu}_{M,\theta}$, $\hat{\Sigma}_{M,\theta}$. The resulting multivariate Gaussian likelihood $p_M(s|\theta) \equiv \mathcal{N}(\hat{\mu}_{M,\theta}, \hat{\Sigma}_{M,\theta})$ can then be numerically maximised with respect to θ, to return an approximate maximimum likelihood estimator (Wood, 2010). It can also be plugged into a Metropolis-Hastings algorithm with flat priors (Wood, 2010), so that MCMC is used as a workhorse to sample from a posterior $\pi_M(\theta|s)$ to ultimately return the posterior mode, and hence a maximum likelihood estimator (a purely Bayesian approach is described below). The introduction of data summaries in the inference has been shown to cope well with chaotic models, where the likelihood would otherwise be difficult to optimize and the corresponding posterior surface may be difficult to explore. More generally, SL is a tool for likelihood-free inference, just like the ABC framework (see reviews Sisson and Fan, 2011; Karabatsos and Leisen, 2018), where the latter can be seen as a nonparametric methodology, while SL uses a parametric distributional assumption on s. SL has found applications in e.g. ecology (Wood, 2010), epidemiology (Engblom et al., 2020; Dehideniya et al., 2019), mixed-effects modeling of tumor growth (Picchini and Forman, 2019). For a recent generalization of the SL family of inference methods using statistical classifiers to directly target estimation of the posterior density, see Thomas et al. (2021) and Kokko et al. (2019).

While ABC is more general than SL, it can sometimes be difficult to tune and it typically suffers from the "curse of dimensionality" when the size of s increases, due to its nonparametric nature. On the other hand, the Gaussianity assumption concerning the summary statistics is the main limitation of SL. At the same time, due to its parametric nature, SL has been shown to perform satisfactorily on problems where $\dim(s)$ is large relative to $\dim(\theta)$ (Ong et al., 2018). Price et al. (2018) framed SL within a pseudo-marginal algorithm for Bayesian inference (Andrieu et al., 2009) and named the method Bayesian SL (BSL). They showed that when s is truly Gaussian, BSL produces MCMC samples from $\pi(\theta|s)$, not depending on the specific choice of M. However, in practice, the inference algorithm does depend on the specific choice of M, since this value affects the chains mixing. Unless the underlying computer model is trivial, producing the M datasets for each θ can be a serious computational bottleneck.

In this work we design a strategy producing a proposal sampler $g(\cdot|s)$ that is conditional to summary statistics of the data, by exploiting the Gaussian assumption for the summary statistics in (B)SL. We call this a *guided sampler*, as it proposes conditionally to data. Moreover, our guided sampler is sequentially built, and we find that our "sequentially Adapted and guided proposal for SL" (named ASL) is easy to construct and adds essentially no overhead, since it exploits quantities that are anyway computed in SL. We stress the importance of rapid convergence to the bulk of the posterior, as while SL may require a large M to get started, once it has approached high posterior probability regions M can be reduced substantially (in Section 6.1 we are forced to start with $M = 1{,}000$ and after a few iterations we can revert to $M = 10$ or 50). Later we briefly discuss how the proposal sampler can also be used in an ABC-MCMC algorithm. We emphasize that our algorithm should be used to rapidly identify the high density region of the posterior, and there initialize other algorithms to produce the actual inference (e.g. using some of the several available adaptive MCMC samplers). We discuss this aspect and suggest possibilities afterwards. In Section 6.4 we show how ASL can be useful with multimodal targets. Furthermore, we correlate log-synthetic likelihoods using a "blockwise" strategy, borrowed from relatively recent advances in pseudo-marginal MCMC literature. This is shown to considerably improve mixing of the chains generated via SL, while not introducing correlation can lead to unsatisfactory simulations when using starting parameter values residing relatively far from the representative ones.

Our paper is structured as follows: in Section 2 we introduce the synthetic likelihood approach. In Section 3 we construct the adaptive proposal distribution via ASL and in Section 4 we construct correlated synthetic likelihoods. In Section 5 we discuss using BOLFI and `ELFI` as an option for SL inference. In Section 6 we discuss four benchmarking simulation studies and a fifth one is in Supplementary Material (Picchini et al., 2022). Code can be found at https://github.com/umbertopicchini/ASL.

2 Synthetic likelihood

We briefly summarize the synthetic likelihood (SL) method as proposed in Wood (2010) and in a Bayesian context in Price et al. (2018) (the latter is detailed in Supplementary Material). The main goal is to produce Bayesian inference for θ, by sampling from (an approximation to) the posterior $\pi(\theta|s) \propto \tilde{p}(s|\theta)\pi(\theta)$ using MCMC, where $\tilde{p}(s|\theta)$ is the density underlying the true (unknown) distribution of s. Wood (2010) proposes a parametric approximation to $\tilde{p}(s|\theta)$, placing the rather strong assumption that $s \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$. The reason for this assumption is that estimators for the unknown mean and covariance of the summaries, μ_θ and Σ_θ respectively, can be obtained straightforwardly via simulation, as described below. As obvious from the notation used, μ_θ and Σ_θ depend on the unknown finite-dimensional vector parameter θ. We denote the synthetic datasets simulated from the assumed model run at a given θ^* with $y_1^*, \ldots, y_M^*$. These are such that $\dim(y_m^*) = \dim(y)$ $(m = 1, \ldots, M)$, with y denoting observed data, and therefore $s \equiv T(y)$. For each dataset it is possible to construct the corresponding (possibly vector valued) summary $s_m^* := T(y_m^*)$, with $\dim(s_m^*) = \dim(s)$. These simulated summaries are used to construct the following estimators:

$$\hat{\mu}_{M,\theta^*} = \frac{1}{M}\sum_{m=1}^{M} s_m^*, \qquad \hat{\Sigma}_{M,\theta^*} = \frac{1}{M-1}\sum_{m=1}^{M}(s_m^* - \hat{\mu}_{\theta^*})(s_m^* - \hat{\mu}_{\theta^*})', \tag{1}$$

with $'$ denoting transposition. By defining $p_M(s|\theta) \equiv \mathcal{N}(\hat{\mu}_{M,\theta}, \hat{\Sigma}_{M,\theta})$, the SL procedure in Wood (2010) samples from the posterior $\pi_M(\theta|s) \propto p_M(s|\theta)\pi(\theta)$, see Algorithm 1. A slight modification of the original approach in Wood (2010) leads to the "Bayesian synthetic likelihood" (BSL) algorithm of Price et al. (2018), which samples from $\pi(\theta|s)$ when s is truly Gaussian, by introducing an unbiased approximation to a Gaussian likelihood. Besides this, the BSL is the same as Algorithm 1. See the Supplementary Material for details about BSL. All our numerical experiments use the BSL formulation of the inference problem. Notice when M is too small or θ^* is implausible, the estimated covariance may mis-behave, e.g. may be not positive-definite: in such case, we attempt a "modified Cholesky factorization" of $\hat{\Sigma}_{M,\theta^*}$, such as the one in Cheng and Higham (1998) (we used the Matlab implementation in Higham, 2015), or we tried to find a "nearest symmetric-positive-definite matrix" (Higham, 1988), using the function by D'Errico (2015).

When the simulator generating the M synthetic datasets is computationally demanding, Algorithm 1 is computer intensive, as it generally needs to be run for a number of iterations R in the order of thousands. The problem is exacerbated by the possibly poor mixing of the resulting chain. The most obvious way to alleviate the problem is to reduce the variance of the estimated likelihoods, by increasing M, but of course this makes the algorithm computationally more intensive. A further problem occurs when the initial θ^* lies far away in the tails of the posterior. This may cause numerical problems when the initial $\hat{\Sigma}_{M,\theta^*}$ is ill-conditioned, possibly requiring a very large M to get the MCMC started, and hence it is desirable to have the chains approach the bulk of the posterior as rapidly as possible.

In the following we propose two strategies aiming at keeping the mixing rate of a MCMC, produced either by SL or BSL, at acceptable levels and also to ease convergence of the chains to the regions of high posterior density. The first strategy results in designing a specific proposal distribution $g(\cdot)$ for use in MCMC via synthetic likelihood: this is a "sequentially Adapted and guided proposal for Synthetic Likelihoods" (shorty ASL) and is described in Section 3. The second strategy reduces the variability in the Metropolis-Hastings ratio α by correlating successive pairs of synthetic likelihoods: this results in "correlated synthetic likelihoods" (CSL) described in Section 4.

Algorithm 1 Synthetic likelihoods MCMC

```
Input: positive integers $M, R$. Observed summaries $s$. Fix starting value $\theta^*$ or generate it from the prior $\pi(\theta)$. Set $\theta_1 := \theta^*$. Define a proposal $g(\theta'|\theta)$. Set $r := 1$.
Output: $R$ correlated samples from $\pi_M(\theta|s)$.
1. Conditionally on $\theta^*$ generate independently from the model $M$ summaries $s^{*1}, \dots, s^{*M}$, compute $\hat{\mu}_{M,\theta^*}$, $\hat{\Sigma}_{M,\theta^*}$ from (1) and $p_M(s|\theta^*) \equiv \mathcal{N}(\hat{\mu}_{M,\theta^*}, \hat{\Sigma}_{M,\theta^*})$.
2. Generate $\theta^\# \sim g(\theta^\#|\theta^*)$. Conditionally on $\theta^\#$ generate independently $s^{\#1}, \dots, s^{\#M}$, compute $\hat{\mu}_{M,\theta^\#}$, $\hat{\Sigma}_{M,\theta^\#}$, and $p_M(s|\theta^\#)$.
3. Generate a uniform random draw $u \sim U(0,1)$, and calculate the acceptance probability
   $$\alpha = \min\left[1, \frac{p_M(s|\theta^\#)}{p_M(s|\theta^*)} \times \frac{g(\theta^*|\theta^\#)}{g(\theta^\#|\theta^*)} \times \frac{\pi(\theta^\#)}{\pi(\theta^*)}\right].$$
   If $u > \alpha$, set $\theta_{r+1} := \theta_r$ otherwise set $\theta_{r+1} := \theta^\#$, $\theta^* := \theta^\#$ and $p_M(s|\theta^*) := p_M(s|\theta^\#)$.
   Set $r := r + 1$ and go to step 4.
4. Repeat steps 2–3 as long as $r \leq R$.
```

3 Guided and sequentially tuned proposals for synthetic likelihoods

In Section 3.1 we illustrate the main ideas of our ASL method. In Section 3.2 we specialize ASL so that we instead obtain a sequence of proposal distributions $\{g_t\}_{t=1}^T$, as detailed in Algorithm 2. What we now introduce in Section 3.1 will also initialize the ASL method, i.e. provide an initial g_0.

3.1 Main idea and initialization

Suppose θ_n^* is a posterior draw generated by some SL procedure (i.e. the standard method from Wood, 2010 or the BSL one from Price et al., 2018) at iteration n, e.g. $\theta_n^* \sim \pi_M(\theta|s)$. Then denote with $\{s_n^{*1}, \dots, s_n^{*M}\}$ a set of M summaries simulated independently from the computer model, conditionally on the same θ_n^*, and define $\bar{s}_n^* = \sum_{m=1}^M s_n^{*m}/M$. By the central limit theorem, for M sufficiently large $\bar{s}_n$ has an approximately Gaussian distribution. Suppose we have at disposal N pairs $\{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$. We set $d_\theta = \dim(\theta)$ and $d_s = \dim(s)$, then $(\theta_n^*, \bar{s}_n^*)$ is a vector having length $d = d_\theta + d_s$. Assume for a moment that the joint vector $(\theta_n^*, \bar{s}_n^*)$ is a d-dimensional Gaussian-distributed vector, with $(\theta_n^*, \bar{s}_n^*) \sim \mathcal{N}_d(m, S)$. We stress that this assumption is made merely to construct a proposal sampler, and does not extend to the actual distribution of (θ, s). We set a d-dimensional mean vector $m \equiv (m_\theta, m_s)$ and the $d \times d$ covariance matrix

$$S \equiv \left[\begin{array}{cc} S_\theta & S_{\theta s} \\ S_{s\theta} & S_s \end{array} \right],$$

where S_θ is $d_\theta \times d_\theta$, S_s is $d_s \times d_s$, $S_{\theta s}$ is $d_\theta \times d_s$ and of course $S_{s\theta} \equiv S'_{\theta s}$ is $d_s \times d_\theta$. We estimate m and S using the N available draws. That is, define $x_n := (\theta_n^*, \bar{s}_n^*)$ then, same as in (1), we have

$$\hat{m} = \frac{1}{N}\sum_{n=1}^{N} x_n, \qquad \hat{S} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \hat{m})(x_n - \hat{m})'. \tag{2}$$

Once $\hat{m}$ and $\hat{S}$ are obtained, it is possible to extract the corresponding entries $(\hat{m}_\theta, \hat{m}_s)$ and $\hat{S}_\theta$, $\hat{S}_s$, $\hat{S}_{s\theta}$, $\hat{S}_{\theta s}$. We can now use well known formulae for conditionals of a multivariate Gaussian distribution, to obtain a proposal distribution (with a slight abuse of notation) $g(\theta|s) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$, with

$$\hat{m}_{\theta|s} = \hat{m}_\theta + \hat{S}_{\theta s}(\hat{S}_s)^{-1}(s - \hat{m}_s), \tag{3}$$

$$\hat{S}_{\theta|s} = \hat{S}_\theta - \hat{S}_{\theta s}(\hat{S}_s)^{-1}\hat{S}_{s\theta}. \tag{4}$$

Hence a new proposal θ^* can be generated as $\theta^* \sim g(\theta|s)$, and is thus "guided" by the summaries of the data s, and gets updated as new posterior draws become available, as further described below. Therefore, this "guided proposal" $g(\theta|s)$ can be employed in place of $g(\theta'|\theta)$ into Algorithm 1, even though we only use this proposal for a limited number of iterations, as clarified below. Clearly the proposal function $g(\theta|s)$ is independent of the last accepted value of θ, hence it is an "independence sampler" (Robert and Casella, 2004), except that its mean and covariance matrix are not kept constant.

The approach outlined so far is essentially step 3 in Algorithm 2, and together with the sequential tuning in Section 3.2, allows for a rapid convergence of the chain towards the high posterior density region. However, this approach does not promote tails exploration. This is not really an issue, as we can let an MCMC incorporating our guided proposal sampler run for a small number of iterations (say 50 iterations, even if we use more iterations for pictorial reasons), where the chain displays a high acceptance rate, and this is useful to collect many accepted draws that we can use to initialize other standard samplers enjoying proven ergodic properties, as detailed in next Section 3.2. Moreover, the next section also illustrates a sampler based on the multivariate Student's distribution.

3.2 Sequential approach

The construction outlined above is only the first step of our guided adaptive sampler for synthetic likelihoods (ASL) methodology, and we now detail it to ease the actual implementation in a sequential way. We define a sequence of $T+1$ "rounds" over which $T+1$

kernels $\{g_t\}_{t=0}^T$ are sequentially constructed. In the first round ($t = 0$), we construct g_0 using the output of $K \gg N$ MCMC iterations, obtained using e.g. a Gaussian random walk. We may consider K as burnin iterations. Once (2)–(3)–(4) are computed using the output $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ of the burnin iterations, we obtain the first adaptive distribution denoted $g_0(\theta|s)$ as illustrated in Section 3.1. We store the draws as $\mathcal{D} := \{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ and then employ g_0 as a proposal density in further N MCMC iterations, after which we perform the following steps: (i) we collect the newly obtained batch of N pairs $\{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$ (where, again, $\theta_n^* \sim \pi_M(\theta|s)$ and $\bar{s}_n^*$ is the sample mean of the *already accepted* simulated summaries generated conditionally to θ_n^*) and add it to the previously obtained ones as $\mathcal{D} := \mathcal{D} \cup \{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$. Then (ii) similarly to (2) we compute the sample mean $\hat{m}^{0:1} = (\hat{m}_\theta^{0:1}, \hat{m}_s^{0:1})$ and corresponding covariance $\hat{S}^{0:1}$, except that here $\hat{m}^{0:1}$ and $\hat{S}^{0:1}$ use the $K + N$ pairs in $\mathcal{D}$. (iii) Update (3)–(4) to $\hat{m}_{\theta|s}^{0:1}$ and $\hat{S}_{\theta|s}^{0:1}$, and obtain $g_1(\theta|s)$. (iv) Use $g_1(\theta|s)$ for further N MCMC moves, stack the new draws into $\mathcal{D} := \mathcal{D} \cup \{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$, and using the $K + 2N$ pairs in $\mathcal{D}$ proceed as before to obtain g_2, and so on until the last batch of N iterations generated using g_T is obtained.

From the procedure we have just illustrated, the sequence of Gaussian kernels has $g_t = g_t(\theta|s) \equiv \mathcal{N}(\hat{m}_{\theta|s}^{0:t}, \hat{S}_{\theta|s}^{0:t})$, with $\hat{m}_{\theta|s}^{0:t}$ and $\hat{S}_{\theta|s}^{0:t}$ the conditional mean and covariance matrix given by

$$\hat{m}_{\theta|s}^{0:t} = \hat{m}_\theta^{0:t} + \hat{S}_{\theta s}^{0:t}(\hat{S}_s^{0:t})^{-1}(s - \hat{m}_s^{0:t}), \tag{5}$$

$$\hat{S}_{\theta|s}^{0:t} = \hat{S}_\theta^{0:t} - \hat{S}_{\theta s}^{0:t}(\hat{S}_s^{0:t})^{-1}\hat{S}_{s\theta}^{0:t}. \tag{6}$$

The proposal function g_t uses all available present and past information, as these are obtained using the most recent version of $\mathcal{D}$, which contains information from the previous $t - 1$ rounds in addition to the latest batch of N draws. Compared to a standard Metropolis random walk, the additional computational effort to implement our method is negligible, as it uses trivial matrix algebra applied on quantities obtained as a by-product of the SL procedure, namely the several pairs $\{\theta_n^*, \bar{s}_n^*\}$. Notice (5)–(6) reduce to $\hat{m}_{\theta|s}^{0:t} \equiv \hat{m}_\theta^{0:t}$ and $\hat{S}_{\theta|s}^{0:t} \equiv \hat{S}_\theta^{0:t}$ respectively as soon as $\hat{m}_s^{0:t} = s$. The latter condition means that the chain is close to the bulk of the posterior and accepted parameters simulate summaries distributed around the observed s. Therefore, when the chain is far from its target, the additional terms in (5)–(6) can help guide the proposals thanks to an explicit conditioning to data.

An alternative to Gaussian proposals are multivariate Student's proposals. We build on the result found in Ding (2016) allowing us to write $\theta_n^* \sim g_t(\theta|s)$, and here $g_t(\theta|s)$ is a multivariate Student's distribution with ν degrees of freedom, and in this case θ_n^* can be simulated using

$$\theta_n^* = \hat{m}_{\theta|s}^{0:t} + \left(\sqrt{\frac{\nu + \delta_n}{\nu + d_s}}(\hat{S}_{\theta|s}^{0:t})^{1/2}\right)\left(Z_n/\sqrt{\frac{\chi^2_{\nu+d_s}}{\nu + d_s}}\right) \tag{7}$$

with $\chi^2_{\nu+d_s}$ an independent draw from a Chi-squared distribution with $\nu + d_s$ degrees of freedom, $\delta_n = (s - \hat{m}_s^{0:t})(\hat{S}_s^{0:t})^{-1}(s - \hat{m}_s^{0:t})'$ and Z_n a d_θ-dimensional standard multivariate Gaussian vector that we simulate at each iteration n and is independent of

$\chi^2_{\nu+d_s}/(\nu+d_s)$. For simplicity, in the following we do not make distinction between the Gaussian and the Student's proposals, and the user can choose any of the two, as they are anyway obtained from the same building-blocks (2)–(6).

As customary in Metropolis-Hastings, when a proposal is rejected at a generic iteration n, the last accepted pair should be stored as $(\theta_n, \bar{s}_n)$. However, should the rejection rate be high (notice we have never incurred into such situation when sampling via ASL), the covariance $\hat{S}^{0:t}_{\theta s}$ would be computed on many identical repetitions of the same $(\theta, \bar{s})$-vectors, this causing numerical instabilities. Therefore, anytime a rejection takes place, we can perform the following when storing the output of the n-th MCMC iteration:

> if proposal $\theta^{\#} \sim g(\theta|s)$ has been rejected at iteration n: resample independently M times with replacement from the last accepted set of summaries $(s^{*1}, \dots, s^{*M})$ (produced from the last accepted θ^*), to obtain the bootstrapped set $(\tilde{s}^{*1}, \dots, \tilde{s}^{*M})$. We use the latter set to compute $\bar{\tilde{s}}^* = \sum_{m=1}^{M} \tilde{s}^{*m}/M$. Hence, at iteration n (and only when proposal $\theta^{\#} \sim g(\theta|s)$ is rejected) we store $\mathcal{D} := \mathcal{D} \cup \{\theta^*_n, \bar{\tilde{s}}^*_n\}$, instead of $\mathcal{D} := \mathcal{D} \cup \{\theta^*_n, \bar{s}^*_n\}$.

This way, the averaged summaries stored in set $\mathcal{D}$ still consist of averages of accepted summaries (as usual), with the benefit that when the acceptance rate is low (which anyway never occurred to us with ASL) $\hat{S}^{0:t}_{\theta s}$ is computed on a set $\mathcal{D}$ that has more varied information, thanks to resampling. This consideration is expressed in step 5 of Algorithm 2. Algorithm 2 constructs the sequence $\{g_t(\theta|s)\}_{t=1}^{T}$ for a SL procedure, and this constitutes our ASL approach. An advantage of ASL is that it is self-adapting.

Algorithm 2 ASL: synthetic likelihoods with a sequentially adapted and guided proposal

```
1: **Input:** $K$ pairs $\{\theta^*_k, \bar{s}^*_k\}_{k=1}^{K}$ from burnin. Positive integers $N$ and $T$. Initialize $\mathcal{D} := \{\theta^*_k, \bar{s}^*_k\}_{k=1}^{K}$.
2: **Output:** $\theta_1, \dots, \theta_T$. Then $\theta_T$ should be used as starting point for another adaptive MCMC algorithm.
3: Construct the proposal density $g_0$ using $\{\theta^*_k, \bar{s}^*_k\}_{k=1}^{K}$ and (2)–(3)–(4) (and optionally propose from (7)). Set $\theta_0 := \theta^*_K$.
4: **for** $t = 1 : T$ **do**
5:     Starting at $\theta_{t-1}$ run $N$ MCMC iterations (SL or BSL) using $g_{t-1}$, producing $\{\theta^*_n, \bar{s}^*_n\}_{n=1}^{N}$. If the current proposal has been rejected at iteration $n$, the $\bar{s}^*_n$ may instead be computed as $\bar{\tilde{s}}^*_n$ (see main text).
6:     Form $\mathcal{D} := \mathcal{D} \cup \{\theta^*_n, \bar{s}^*_n\}_{n=1}^{N}$, compute $(\hat{m}^{0:t}, \hat{S}^{0:t})$ on $\mathcal{D}$, update $(\hat{m}^{0:t}_{\theta|s}, \hat{S}^{0:t}_{\theta|s})$ to construct $g_t$.
7:     Set $\theta_t := \theta^*_N$.
8: **end for**
9: Return $\theta_1, \dots, \theta_T$ to be provided as input to another adaptive MCMC algorithm for BSL or CSL.
```

A disadvantage is that, since the adaptation results into an independence sampler, it does not explore a neighbourhood of the last accepted draw, and newly accepted N draws obtained at stage t might not necessarily produce a rapid change into mean and covariance for the proposal function g_{t+1} (should a rapid change actually be required for optimal exploration of the parameter space). This is why in our applications we always use $N = 1$. That is, the proposal distribution is updated at each iteration by immediately incorporating information provided by the last accepted draw. As clear from the output of Algorithm 2, we recommend to use T iterations of ASL to return i) θ_T which is then used as starting parameter value for a run of BSL (or CSL see Section 4) together with a standard MCMC proposal sampler; and ii) the sequence $\theta_1, \ldots, \theta_T$ (notice this excludes the initial burnin of K iterations) of which we compute the sample covariance matrix, and the latter is used to initiate the adaptive MCMC algorithm of Haario et al. (2001), this one having proven ergodic properties (see the Supplementary Material for details on how this is performed). The above means that the inference results we report are based on draws using Haario et al. (2001) (thanks to the useful initialization via ASL). However, after the T ASL iterations, besides Haario et al. (2001) other adaptive MCMC algorithms with proven ergodic properties could be used: possibilities are e.g. Andrieu and Thoms (2008) or Vihola (2012). Moreover, an interesting use of ASL arises with multimodal targets: if several chains are run in parallel and are initialised at different parameter values, the nature of ASL to rapidly "jump" to high density regions can point the researcher to the existence of multiple modes within few iterations of ASL (this is illustrated in Section 6.4).

In our experiments we use a relatively small number of burnin iterations K (say $K = 200$ or 300), and when ASL is started we immediately observe a large "jump" towards the posterior mode. Importantly, rapid convergence via ASL also helps reducing the computational effort by re-tuning M: in fact, while a large value of M can be necessary when setting θ_0 in tail regions of the posterior, once the chain has converged towards the bulk of the posterior it is possible to reduce M substantially. See the g-and-k example in Section 6.1, where it is necessary to start with $M = 1{,}000$, and after using ASL for a few iterations we can revert to $M = 10$ or 50.

Our ASL strategy is inspired by the sequential neuronal likelihood approach found in Papamakarios et al. (2019). In Papamakarios et al. (2019) N MCMC draws obtained in each of T stages sequentially approximate the likelihood function for models having an intractable likelihood, whose approximation at stage t is obtained by training a neuronal network (NN) on the MCMC output obtained at stage $t-1$. Their approach is more general (and it is aimed at approximating the likelihood, not the MCMC proposal), but has the disadvantage of requiring the construction of a NN, and then the NN hyperparameters must be tuned at every stage t, which of course requires domain knowledge and computational resources. Our approach is framed specifically for inference via synthetic likelihoods, which is a limitation *per-se*, but it is completely self-tuning, with the possible exception of the burnin iterations where an initial covariance matrix must be provided by the user, though this is a minor issue when the number of parameters is limited. Notice, a possible interesting application of our guided sampler could be envisioned with ABC-MCMC algorithms (Marjoram et al., 2003). Even though ABC-MCMC is typically run by simulating a single vector of summary statistics at a given θ (though it is also possible to consider pseudo-marginal versions, as in Picchini and Everitt, 2019), nothing prevents to run ASL for a few iterations in an ABC-MCMC context, by simulating multiple summaries at each θ as in SL, and then revert to simulating a single summary vector once the chain has reached the bulk of the ABC posterior.

4 Correlated synthetic likelihood

Following the success of the pseudo-marginal method (PM), returning exact Bayesian inference whenever a non-negative and unbiased estimate of an intractable likelihood is available (Beaumont, 2003, Andrieu et al., 2009, Andrieu et al., 2010), there has been much research aimed at increasing the efficiency of particle filters (or sequential Monte Carlo) for Bayesian inference in state-space models, see Schön et al. (2018) for an approachable review. A recent important addition to PM methodology, improving the acceptance rate in Metropolis-Hastings algorithms, considers inducing some correlation between the likelihoods appearing in the Metropolis-Hastings ratio. The idea underlying correlated pseudo-marginal methods (CPM), as initially proposed in Dahlin et al. (2015) and Deligiannidis et al. (2018), is that having correlated likelihoods will reduce the stochastic variability in the acceptance ratio. This reduces the stickiness in the MCMC chain, which is typically due to excessively varying likelihood approximations, when these are obtained using a "too small" number of Monte Carlo draws (named "particles"). In fact, while the variability of these estimates can be mitigated by increasing the number of particles, of course this has negative consequences on the computational budget. Instead CPM strategies allow for considerably smaller number of particles when trying to alleviate the stickiness problem, see for example Golightly et al. (2019) for applications to stochastic kinetic models, and Wiqvist et al. (2021) and Botha et al. (2021) for stochastic differential equation mixed-effects models. Interestingly, implementing CPM approaches is trivial. Deligiannidis et al. (2018) and Dahlin et al. (2015) correlate the estimated likelihoods at the proposed and current values of the model parameters by correlating the underlying standard normal random numbers used to construct the estimates of the likelihood, via a Crank-Nicolson proposal. We found particular benefit with the "blocked" PM approach (BPM) of Tran et al. (2016) (see also Choppala et al., 2016 for inference in state-space models), which we now describe in full generality, i.e. regardless of our synthetic likelihoods approach which is instead considered later.

Denote with U the vector of all "auxiliary variables", i.e. pseudorandom numbers (typically standard Gaussian or uniform) that are necessary to produce a non-negative unbiased likelihood approximation $\hat{p}(y|\theta, \mathrm{U})$ at a given parameter θ for data y. Notice U should contain the pseudo-random numbers that are used to "forward simulate" from a model, but can include also other auxiliary variables, for example the pseudo-random numbers that are generated when performing the resampling step in sequential Monte Carlo. In Tran et al. (2016) the set U is divided into G blocks $U = (U_{(1)}, \dots, U_{(G)})$, and one of these blocks is updated jointly with θ in each MCMC iteration as described below. Let $\hat{p}(y|\theta, \mathrm{U}_{(i)})$ be the estimated unbiased likelihood obtained using the ith block of random variates $U_{(i)}$, $i = 1, \dots, G$. Define the joint posterior of θ and U as

$$\pi(\theta, \mathrm{U}|y) \propto \hat{p}(y|\theta, \mathrm{U})\pi(\theta) \prod_{i=1}^{G} p_U(\mathrm{U}_{(i)}), \tag{8}$$

where θ and U are a-priori independent and

$$\hat{p}(y|\theta, \mathrm{U}) := \frac{1}{G} \sum_{i=1}^{G} \hat{p}(y|\theta, \mathrm{U}_{(i)}) \tag{9}$$

is the average of the G unbiased likelihood estimates and hence also unbiased. We then update the parameters jointly with a randomly-selected block $\mathrm{U}_{(K)}$ in each MCMC iteration, with $\Pr(K=k)=1/G$ for any $k=1,\dots,G$. "Updating a randomly selected block" means that only for that picked block $U_{(k)}$ new pseudorandom values are produced (and hence are "refreshed") while for the other blocks these variates are kept fixed to the previously accepted values. Using this scheme, the acceptance probability for a joint move from the current set (θ^c,U^c) to a proposed set (θ^p,U^p) generated using some proposal function $g(\theta^p,\mathrm{U}^p|\theta^c,\mathrm{U}^c)=g(\theta^p|\theta^c)g(\mathrm{U}^p|\mathrm{U}^c)$, is

$$\alpha=\min\left\{1,\frac{\hat{p}\left(y|\theta^p,\mathrm{U}^c_{(1)},\dots,\mathrm{U}^c_{(k-1)},\mathrm{U}^p_{(k)},\mathrm{U}^c_{(k+1)},\dots,\mathrm{U}^c_{(G)}\right)\pi\left(\theta^p\right)}{\hat{p}\left(y|\theta^c,\mathrm{U}^c_{(1)},\dots,\mathrm{U}^c_{(k-1)},\mathrm{U}^c_{(k)},\mathrm{U}^c_{(k+1)},\dots,\mathrm{U}^c_{(G)}\right)\pi\left(\theta^c\right)}\frac{g\left(\theta^c|\theta^p\right)}{g\left(\theta^p|\theta^c\right)}\right\}. \quad (10)$$

Hence in case of proposal acceptance we update the joint vector $(\theta^c,\mathrm{U}^c):=(\theta^p,\mathrm{U}^p)$ and move to the next iteration, where $\mathrm{U}^p=(\mathrm{U}^c_{(1)},\dots,\mathrm{U}^c_{(k-1)},\mathrm{U}^p_{(k)},\mathrm{U}^c_{(k+1)},\dots,\mathrm{U}^c_{(G)})$. The resulting chain targets (8) (Tran et al., 2016). Notice the random variates used to compute the likelihood at the numerator of (10) are the same ones as for the likelihood at the denominator except for the k-th block, hence $G-1$ blocks are shared between the numerator and denominator. Perturbing only a small fraction of the pseudo-random numbers induces beneficial correlation between subsequent pairs of likelihood estimates, as in this case the variance of α gets smaller compared to having all entries in U getting updated at each iteration. Also, we considered $g(\mathrm{U}^p|\mathrm{U}^c)\equiv p_U(U^p_{(k)})$ hence the simplified expression (10). The correlation between $\log\hat{p}\,(y|\theta^p,\mathrm{U}^p)$ and $\log\hat{p}\,(y|\theta^c,\mathrm{U}^c)$ is approximately $\rho=1-1/G$ (Tran et al., 2016), so the larger the number of groups G that can be formed and the higher the correlation (at least theoretically). Also, note that the G approximations $\hat{p}(y|\theta,\mathrm{U}_{(i)})$ can be run in parallel on multiple processors when these likelihoods are approximated using particle filters.

We now consider synthetic likelihoods. Denote with U_j the vector of auxiliary variables employed when producing the j-th model simulation ($j=1,\dots,M$). Denote with $(U_1,\dots,U_M)$ the vector stacking the variates generated across all M model simulations. We distribute those variates across G blocks: assume for simplicity that M is a multiple of G, so that for example the i-th block $U_{(i)}$ could be the collection of the pseudo-random numbers used in a small subset of the M model simulations, so that $\sum_{i=1}^G\dim(U_{(i)})=\dim(U_1,\dots,U_M)$ and $U_{(i)}\bigcap U_{(i')}=\{\emptyset\}$, $i\neq i'$. That is $(U_{(1)},\dots,U_{(G)})$ is a partition of $(U_1,\dots,U_M)$. Same as before, in each MCMC iteration we "refresh" the variates from a randomly sampled block, while the other variates are kept fixed to the previously accepted values. In our synthetic likelihood approach we do not make use of (9) and take instead $p(s|\theta,\mathrm{U})$ without decomposing this into a sum of G contributions. We do not in fact compute separately the $p(s|\theta,\mathrm{U}_{(i)})$, since we found that in order for each $p(s|\theta,\mathrm{U}_{(i)})$ to behave in a numerically stable way, a not too small number of simulations $M_{(i)}$ should be devoted for each sub-likelihood term, or otherwise the corresponding estimated covariance may misbehave (e.g., may result not positive-definite). Therefore, in practice, we just obtain the joint $p(s|\theta,\mathrm{U})$, and (10) becomes

$$\alpha=\min\left\{1,\frac{p\left(s|\theta^p,\mathrm{U}^c_{(1)},\dots,\mathrm{U}^c_{(k-1)},\mathrm{U}^p_{(k)},\mathrm{U}^c_{(k+1)},\dots,\mathrm{U}^c_{(G)}\right)\pi\left(\theta^p\right)}{p\left(s|\theta^c,\mathrm{U}^c_{(1)},\dots,\mathrm{U}^c_{(k-1)},\mathrm{U}^c_{(k)},\mathrm{U}^c_{(k+1)},\dots,\mathrm{U}^c_{(G)}\right)\pi\left(\theta^c\right)}\frac{g\left(\theta^c|\theta^p\right)}{g\left(\theta^p|\theta^c\right)}\right\}, \quad (11)$$

which we therefore call "correlated synthetic likelihood" (CSL) approach. From the analytic point of view our correlated likelihood $p(s|\theta, \mathrm{U})$ is the same unbiased approximation given in Price et al. (2018) (also in Supplementary Material), hence CSL uses the BSL approach, the only difference with BSL being that the numerator and denominator of (11) have $G-1$ blocks in common, while in BSL all pseudo-random numbers are refreshed at each iteration for each new likelihood.

In our experiments we show that using the acceptance criterion (11) into Algorithm 1 (regardless of the use of our ASL proposal kernel) is of benefit to ease convergence and also increase chains mixing. Moreover, it comes with no computational overhead compared to not using correlated synthetic likelihoods. The only potential issue would be some careful extra coding and the need to store $(\mathrm{U}_{(1)}, \dots, \mathrm{U}_{(G)})$ in memory, which could be large dimensional with complex model simulators.

5 Algorithmic initialization using BOLFI and ELFI

This section does not contain novel material, but it is useful to inform modellers using SL approaches of alternative strategies to initialize SL algorithms. We consider the case where obtaining a reasonable starting value θ_1 for θ by trial-and-error is not feasible, due to the computational cost of evaluating the SL density at many candidates for θ_1. At minimum, we need to find a value θ_1 such that the corresponding SL density (the biased p_M or the unbiased one in the sense of Price et al., 2018) has a positive definite covariance matrix $\hat{\Sigma}$. This is not ensured when the summaries are simulated from highly non-representative values of θ, which would result in an MCMC algorithm that halts. The issue is critical, as testing many values θ_1 can be prohibitively expensive, both because the dimension of θ can be large and because the model itself might be slow to simulate from.

An approach developed in Gutmann and Corander (2016) uses Bayesian optimization when the likelihood function is intractable but realizations from a stochastic model simulator are available, which is exactly the framework that applies to ABC and SL. The resulting method, named BOLFI (Bayesian optimization for likelihood-free inference), locates a θ that either minimizes the expected value of $\log \Delta$, where Δ is some discrepancy between simulated and observed summary statistics, say $\Delta = \parallel s^* - s \parallel$ for some distance $\parallel \cdot \parallel$, or alternatively can be used to minimize the negative log-SL expression. For example, $\parallel \cdot \parallel$ could be the Euclidean distance $((s^*-s)'(s^*-s)')^{1/2}$, or a Mahalanobis distance $((s^*-s)'A(s^*-s)')^{1/2}$ for some square matrix A weighting the individual contributions of the entries in s^* and s (see Prangle et al., 2017). The appeal of BOLFI is that (i) it is able to rapidly focus the exploration in those regions of the parameter space where either Δ is smaller, or the SL is larger, and (ii) it is implemented in `ELFI` (Lintusaari et al., 2018), the Python-based open-source engine for likelihood-free inference. Hence, when dealing with expensive simulators, BOLFI is ideally positioned to minimize the number of attempts needed to obtain a reasonable value θ_1, to be used to initialize the synthetic likelihoods approach. BOLFI replaces the expensive realizations from the model simulator with a "surrogate simulator" defined by a Gaussian process (GP, Rasmussen and Williams, 2006). Using simulations from the actual (expensive) simulator to form a collection of pairs such as $(\theta, \log \Delta)$, the GP is trained on the generated pairs and the actual optimization in BOLFI only uses the computationally cheap GP simulator. This means that the optimum returned by BOLFI does not necessarily reflect the best θ generating the observed s. It is possible to use the BOLFI optimum to initialize some other procedure within `ELFI`, such as Hamiltonian Monte Carlo via the NUTS algorithm of Hoffman and Gelman (2014). However, the `ELFI` version of NUTS uses, again, the GP surrogate of the likelihood function. Once the BOLFI optimum is obtained, it can be used to initialise (B)SL MCMC which still uses simulations from the true model, and these may be expensive, but at least are initialised at a θ which should be "good enough" to avoid a long and expensive burnin. Illustrations of BOLFI are in Sections 6.1 and 6.2. A more recent contribution, exploiting GPs to predict a log-SL, is in Järvenpää et al. (2020).

6 Simulation studies

Here follow four simulation studies. A fifth one, using a "perturbed" α-stable model built to pose a challenge to CSL, is in Supplementary Material. In all the considered examples we use $N = 1$, i.e. the ASL proposal kernel is updated at each iteration. Within ASL we always use a Gaussian proposal based on (5)–(6), and never the multivariate Student's one.

6.1 g-and-k distribution

The g-and-k distribution is a standard toy model for case studies having intractable likelihoods (e.g. Allingham et al., 2009; Fearnhead and Prangle, 2012), in that its simulation is straightforward, but it does not have a closed-form probability density function (pdf). Therefore the likelihood is analytically intractable. The g-and-k distributions is a flexibly shaped distribution that is used to model non-standard data through a small number of parameters. It is defined by its quantile function, see Prangle (2017) for an overview. Essentially, it is possible to generate a draw Q from the distribution using the following scheme

$$Q = A + B\left[1 + c\frac{1-\exp(-g\cdot u)}{1+\exp(-g\cdot u)}\right](1+u^2)^k \cdot u, \tag{12}$$

where $u \sim N(0,1)$, A and B are location and scale parameters and g and k are related to skewness and kurtosis. Parameters restrictions are $B > 0$ and $k > -0.5$. We assume $\theta = (A, B, g, k)$ as parameter of interest, by noting that it is customary to keep c fixed to $c = 0.8$ (Drovandi and Pettitt, 2011; Rayner and MacGillivray, 2002). We use the summaries $s(w) = (s_{A,w}, s_{B,w}, s_{g,w}, s_{k,w})$ suggested in Drovandi and Pettitt (2011), where w can be observed and simulated data y and y^* respectively:

$$\begin{aligned} s_{A,w} &= P_{50,w}, \qquad s_{B,w} = P_{75,w} - P_{25,w}, \\ s_{g,w} &= (P_{75,w} + P_{25,w} - 2s_{A,w})/s_{B,w}, \\ s_{k,w} &= (P_{87.5,w} - P_{62.5,w} + P_{37.5,w} - P_{12.5,w})/s_{B,w}, \end{aligned}$$

where $P_{q,w}$ is the qth empirical percentile of w. That is $s_{A,w}$ and $s_{B,w}$ are the median and the inter-quartile range of w respectively. We use the simulation strategy outlined above to generate data y, consisting of 1,000 independent samples from the g-and-k distribution using parameters $\theta = (A, B, g, k) = (3, 1, 2, 0.5)$. We place uniform priors on the parameters: $A \sim U(-30, 30)$, $B \sim U(0, 30)$, $g \sim U(0, 30)$, $k \sim U(0, 30)$.

We run five inference attempts independently, always starting at $\theta_0 = (7.389, 7.389, 2.718, 1.221)$ and using the same data. For all experiments, $M = 1{,}000$ model simulations are produced at each proposed parameter and we found this value of M to be necessary given the used starting parameters, or we would not collect enough parameter moves. However if we instead initialize the simulations close to the true values then a considerably smaller M can be employed (this is discussed later), which shows that our contribution on accelerating convergence to the high posterior probability region is important. We start by running $K = 200$ burnin iterations, during which we advance the chain by proposing parameters using a Gaussian random walk acting on log-scale, i.e. on $\log\theta$, with a constant diagonal covariance matrix having standard deviations given by $[0.025, 0.025, 0.025, 0.025]$ for $(\log A, \log B, \log g, \log k)$ respectively. Given the short burnin, in the first K iterations we implement a Markov-chain-within-Metropolis strategy (MCWM, Andrieu et al., 2009) to increase the mixing of the algorithm before our sequentially guided ASL strategy starts (shortly, MCWM differs from a standard Metropolis-Hastings algorithm in that the stochastic likelihood approximation in the denominator of the Metropolis-Hastings ratio is re-evaluated at the last accepted parameter value, instead of using the value of the previously accepted synthetic likelihood). Notice the use of MCWM is strictly limited to the burnin iterations, since MCWM doubles the execution time and its theoretical properties are not well understood. At iteration $K+1$, our ASL Algorithm 2 starts and is let run for 300 iterations (notice a much smaller number of iterations than 300 can be used, say 50. We chose 300 for pictorial reasons as the effect of ASL gets better noticed in figures). Afterwards BSL inherits the last draw accepted by ASL and reverts to using the adaptive Metropolis random walk proposal of Haario et al. (2001), thereafter denoted “Haario”, which is used for further 2,800 iterations and is adapted as described in Supplementary Material. Therefore the total length of the chain is 3,300 ($K = 200$ iterations, then 300 ASL iterations then 2,800 further iterations). The covariance matrix in the adaptive proposal of “Haario” is updated every 30 iterations. The five independent inference attempts are in Figure 1a. We notice that during the burnin the chains are still quite far from the ground truth. However, as soon as ASL kicks in (iteration 201), we notice a large jump towards the true parameters. The proposal in the ASL algorithm produces a high acceptance rate which although it induces very local moves, it never gets stuck and thus provides useful information to initialize the covariance matrix in “Haario”.

We now avoid using our guided ASL and run BSL using again MCWM during the burnin iterations and the adaptive “Haario” strategy for the remaining iterations, with results in Figure 1b. This shows the difficulty of running BSL when starting parameters are in the tails of the posterior, where several runs completely fail and only occasionally they manage to reach the bulk of the posterior. Furthermore, the patterns are very sticky. This is because the adaptive “Haario” proposal tunes the covariance of the Gaussian random walk sampler on the previous history of the chain, which becomes problematic if many rejections occur, as in this case the covariance shrinks, thus making the recovery difficult. This is why the high acceptance rate of ASL, coupled to the rapid convergence towards the posterior's bulk, helps collecting moves that are useful for the learning of the covariance matrix for the adaptive random walk proposal.

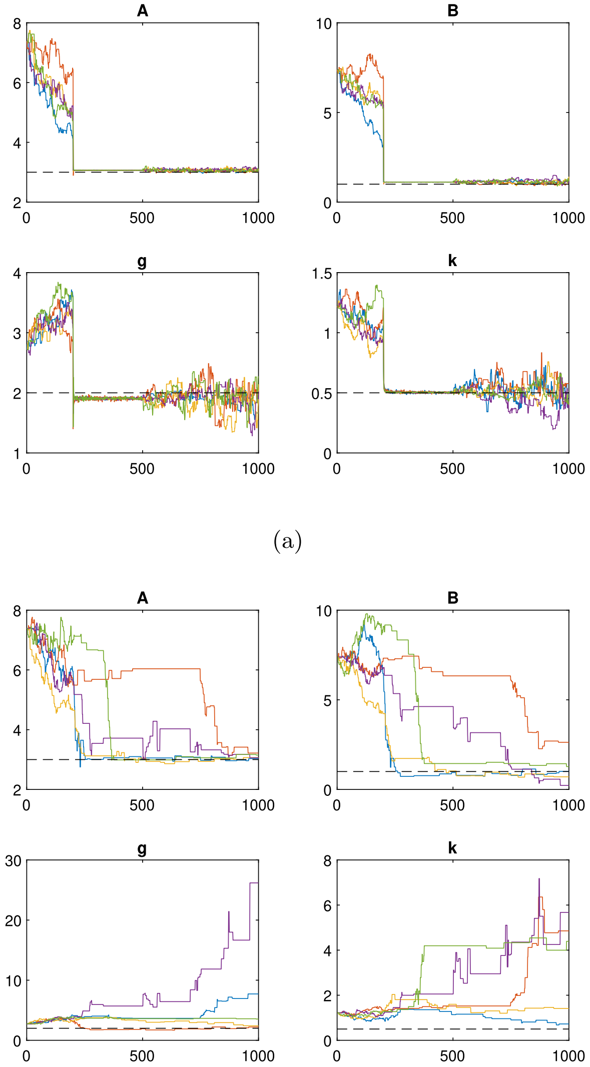
We mentioned that if a starting parameter value is chosen closer to the ground truth value a smaller M can be employed (and recall from Figure 1b that with a standard adaptive proposal method BSL failed even with $M = 1{,}000$). As an example, we take the sample mean of the acceptances produced via ASL from iteration 200 to 500, and use this sample mean to initialize BSL when using the "Haario" proposal with as little as $M = 50$: the mixing of BSL in this case is very satisfactory (results shown in the Supplementary Material) and actually even using $M = 10$ allows good mixing but slightly worse tail performance. Therefore, ASL can be really useful in enabling standard BSL to be used with considerable computational savings (5,000 iterations of BSL can be performed in under a minute when $M = 50$).

Using correlated synthetic likelihood without ASL

Here we consider the correlated synthetic likelihood (CSL) approach outlined in Section 4, without the use of our ASL approach for proposing parameters, to better appreciate the individual effect of using correlated likelihoods. Notice (12) immediately suggests how to implement CSL, since the u appearing in (12) can be thought as a scalar realization of the U variate in Section 4. We initialised parameters at the same starting values as in the previous experiments, across five independent inference attempts. We used CSL throughout, including the burnin phase, that is we do not employ MCWM during the burnin. After 200 burnin iterations with fixed covariance matrix, we propose parameters using "Haario". We illustrate results obtained with $G = 100$ blocks, which should imply a theoretical correlation of $\rho = 1 - 1/100 = 0.99$ between estimated synthetic loglikelihoods, see Figure 2. Figure 2 shows that, while two attempts failed (essentially because the 200 burnin iterations did not produce any acceptance), the remaining attempts managed to reach the ground-truth values. Recall that when using BSL without induced correlation (and employing the same "Haario" proposal sampler) we produced Figure 1b. The benefits of recycling pseudo-random variates are noticeable. Similar plots, but using $G = 50$, are in Supplementary Material. The comparison between the two cases $G = 50$ and $G = 100$ shows that inducing higher correlation (i.e. $G = 100$) allow faster convergence to ground-truth parameters, however at the same time the mixing is reduced due to a reuse of perhaps too many U-variates, whereas with $G = 50$ the chains appear to mix better.

Initialization using ELFI and BOLFI

Here we show results from the BOLFI optimizer (discussed in Section 5) to find a promising area in the posterior region and hence provide a useful starting value for SL. We use the `ELFI` software. In this particular example BOLFI uses a Gaussian Process



(a)

(b)

Figure 1: g-and-k: (a) five inference runs using BSL with ASL sampler employed from iteration 200 to 500. We display the first 1,000 iterations to emphasize the effect of the ASL adaption. The black dashed lines mark ground-truth parameters. (b) five inference runs of BSL (only the first 1,000 iterations are displayed for comparison with Figure 1a). Here BSL uses the sampler of Haario et al. (2001) from iteration 200 onward.

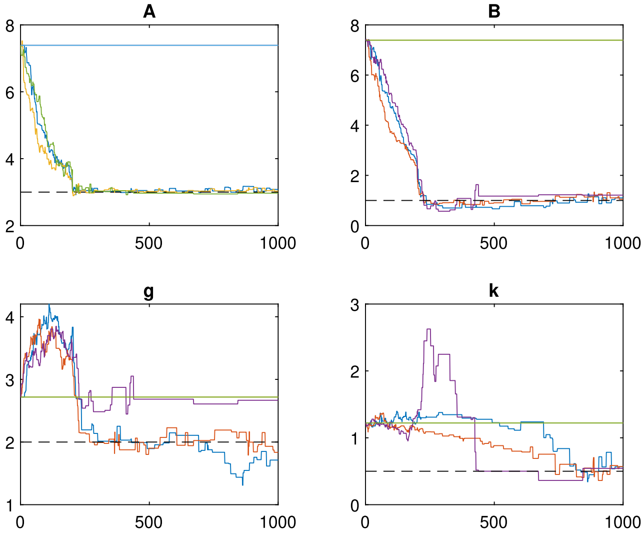
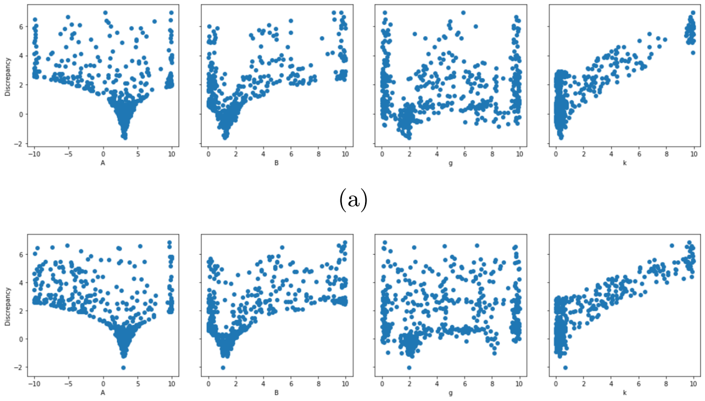


Figure 2: g-and-k: 1,000 iterations from CSL, using $G = 100$ groups. The black dashed lines mark ground-truth parameters. Solid horizontal lines correspond to failed attempts.

(GP) to learn the possibly complex and nonlinear relationship between discrepancies (or log-discrepancies) $\log \Delta$ and corresponding parameters θ. We found that for this specific example, where we set very wide and vague priors, we could not infer the parameters using BOLFI with the LCB (lower confidence bound) acquisition function regardless the value set for J_1. This is because while in previous inference attempts we used MCMC methods to explore the posterior and having very vague priors was still feasible, here having initial samples provided by very uninformative priors is not manageable. In this section we use $A \sim U(-10, 10)$, $B \sim U(0, 10)$, $g \sim U(0, 10)$, $k \sim U(0, 10)$. These priors are narrower than in previous attempts but are still wide and uninformative enough to make this experiment interesting and challenging. Once the J_1 training samples are obtained, BOLFI starts optimizing parameters by iteratively fitting a GP and proposing points $\theta_{(j)}$ such that each $\theta_{(j)}$ attempts at reducing $\log \Delta$, $j = 1, \ldots, J_2$. We first consider $J_2 = 500$ and then $J_2 = 800$. The clouds of points in Figure 3 represent all $J_1 + J_2$ values of log-discrepancies $\log \Delta$ (for $(J_1, J_2) = (20, 500)$ and $(J_1, J_2) = (100, 500)$) and corresponding parameter values. The smallest values of $\log \Delta$ cluster around the ground-truth parameters which we recall are $A = 3$, $B = 1$, $g = 2$, $k = 0.5$. The values of the optimized discrepancies are in Supplementary Material. Even with a very small J_1 the obtained results appear very promising. Also, even though the estimates for k seem to be bounded by the lower limit we set for its prior, we can clearly notice a trend, in that smaller values for k return smaller discrepancies. BOLFI can be an effective tool to initialize an MCMC procedure for synthetic likelihoods.



(a)

(b)

Figure 3: g-and-k: log-discrepancies for the tested parameters using BOLFI with $J_1 = 20$ (top) and $J_1 = 100$ (bottom). From left to right: plots for A, B, g and k respectively.

6.2 Supernova cosmological parameters estimation with twenty summary statistics

We present an astronomical example taken from Jennings and Madigan (2017). There, the ABC algorithm by Beaumont et al. (2009) was used for likelihood-free inference. The algorithm in Beaumont et al. (2009) is a sequential Monte Carlo (SMC) sampler, hereafter denoted ABC-SMC, which propagates many parameter values ("particles") through a sequence of approximations of the posterior distribution of the parameters. The sequence of approximations depends on the sequence of tolerances $\epsilon_{1:T}$, where T is the final iteration of the procedure. The different approaches used in order to create the series of decreasing tolerances (Beaumont et al., 2009; Del Moral et al., 2012), together with the choice for T, can lead to inefficient sampling (Simola et al., 2020). For this reason, rather than using the ABC-SMC algorithm, we employed one of its extensions, the "adaptive ABC Population Monte Carlo" (hereafter aABC-PMC) found in Simola et al. (2020). When using the aABC-PMC algorithm both the series of decreasing tolerances and T are automatically selected, by looking at the online behaviour of the approximations to the posterior distribution (aABC-PMC is also implemented in ELFI). Our goal is to show how synthetic likelihoods may be as well used in order to tackle the inferential problem, and a comparison with aABC-PMC and BOLFI is presented. In Jennings and Madigan (2017) the analysis relied on the SNANA light curve analysis package (Kessler et al., 2009) and its corresponding implementation of the SALT-II light curve fitter presented in Guy et al. (2010). A sample of 400 supernovae with redshift range $z \in [0.5, 1.0]$ are simulated and then binned into 20 redshift bins. However, for this example, we did not use SNANA and data is instead simulated following the procedure in Supplementary Material. The model that describes the distance modulus as a function of redshift z, known in the astronomical literature as Friedmann–Robertson-Model

(Condon and Matthews, 2018), is:

$$\mu_i(z_i; \Omega_m, \Omega_\Lambda, \Omega_k, w_0, h_0) \propto 5 \log_{10}\left(\frac{c(1+z_i)}{h_0}\right) \int_0^{z_i} \frac{dz'}{E(z')}, \tag{13}$$

where $E(z) = \sqrt{\Omega_m(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda e^{3\int_0^z dln(1+z')[1+w(z')]}}$.

The cosmological parameters involved in (13) are five. The first three parameters are the matter density of the universe, Ω_m, the dark energy density of the universe, Ω_Λ and the radiation and relic neutrinos, Ω_k. A constraint is involved when dealing with these three parameters, which is $\Omega_m + \Omega_\Lambda + \Omega_k = 1$ (Genovese et al., 2009; Tripathi et al., 2017; Usmani et al., 2008). The final two parameters are, respectively, the present value of the dark energy equation, w_0, and the Hubble constant, h_0. A common assumption involves a flat universe, leading to $\Omega_k = 0$, as shown in Tripathi et al. (2017); Usmani et al. (2008). As a result, (13) simplifies and in particular $E(z)$ can be written as $E(z) = \sqrt{\Omega_m(1+z)^3 + (1-\Omega_m)e^{3\int_0^z dln(1+z')[1+w(z')]}}$, where we note that $\Omega_\Lambda = 1 - \Omega_m$. Same as in Jennings and Madigan (2017), we work under the flat universe assumption. Concerning the Dark Energy Equation of State (EoS), $w(\cdot)$, we use the parametrization proposed in Chevallier and Polarski (2001) and in Linder (2003):

$$w(z) = w_0 + w_a(1-a) = w_0 + w_a \frac{z}{1+z}. \tag{14}$$

According to (14), w is assumed linear in the scale parameter. Another common assumption relies on w being constant; in this case $w = w_0$. We note that several parametrizations have been proposed for the EoS (see for example Huterer and Turner (2001), Wetterich (2004) and Usmani et al. (2008)). For the present example, ground-truth parameters are set as follow: $\Omega_m = 0.3$, $\Omega_k = 0$, $w_0 = -1.0$ and $h_0 = 0.7$.

In the present study h_0 is assumed known. Similarly to Jennings and Madigan (2017), we aim at inferring the cosmological parameters $\theta = (\Omega_m, w_0)$. The distance function used to compare μ with the "simulated" data $\mu_{sim}(z)$ is:

$$\rho(\mu, \mu_{sim}(z)) = \sum_i (\mu_i - \mu_{sim}(z_i))^2. \tag{15}$$

We recall that the aABC-PMC algorithm in Simola et al. (2020) uses a series of automatically selected decreasing tolerances $\epsilon_{1:T}$, each inducing a better approximation to the true posterior distribution as $t \in [1, T]$ increases. When the stopping rule, based on the improvement between two consecutive posterior distributions is satisfied, the procedure automatically halts. While the ABC posterior based on ϵ_1 uses the prior distribution as proposal function, for $t > 1$ the aABC-PMC uses the previous iteration's ABC posterior to produce candidates, just like regular ABC-PMC or other sequential ABC procedures. In this work, as done also by Jennings and Madigan (2017), we follow Beaumont et al. (2009) regarding the selection of the perturbation kernel, which is a Gaussian distribution centered to the selected particle and having variance equal to twice the weighted sample variance of the particles selected in the previous iteration.

The specifications for the aABC-PMC algorithm are found in Simola et al. (2020). For all experiments, we set priors $\Omega_m \sim Beta(3,3)$, since Ω_m must be in $(0,1)$, and $w_0 \sim \mathcal{N}(-0.5, 0.5^2)$.

Inference

We describe how to forward simulate from the model in Supplementary Material. We take $s = (\mu_1, \ldots, \mu_{20})$ as "observed" summary statistics corresponding to the stochastic input generated as described in Supplementary Material. Notice, in our case s is the trivial summary statistic, in that $(\mu_1, \ldots, \mu_{20})$ is the data itself. We investigate the assumption in the Supplementary Material and find that this is statistically supported, at least for summaries simulated at ground-truth parameter values. However, notice that a different behaviour might occur at other values of θ, for example at those values far from the ground truth. We found it impractical to consider M in the order of thousands, however using a smaller value of, say, $M = 100$ would produce an ill-conditioned covariance matrix. To overcome this problem we found it essential to use a shrinkage estimator of $\hat{\Sigma}_{M,\theta}$, such as the one due to Warton (2008) and employed in a BSL context in Nott et al. (2019). This way we managed to use as little as $M = 100$ model simulations. In this section we denote the BSL approach using shrinkage as "sBSL". We compare sBSL with the correlated synthetic likelihoods approach plugged into ASL, and denote this method "ACSL" (we employed shrinkage also within ACSL). We always use $M = 100$, and within ACSL we experiment with several numbers of blocks, namely $G = 5$ and 10. For all methods, starting parameter values are $(\Omega_m = 0.90, w_0 = -0.5)$, see the Supplementary Material for further details on the MCMC settings. We first note that sBSL is unable to move away from the starting parameter values, and hence this attempt is a failure. Introducing correlation between synthetic loglikelihoods is a key feature for the success of ACSL in this case study.

Traceplots for 11,200 draws from ACSL when $G = 5, 10$ are in Supplementary Material. Having $G > 1$ helps proposals acceptance during the burnin period, so that when ACSL starts it is provided with useful information from the burnin. The output of aABC-PMC is produced by 1,000 particles (the final tolerance that is automatically selected by the algorithm after $T = 9$ iterations is $\epsilon_9 = 30.5$). For comparison with aABC-PMC inference, where the latter is produced by a "cloud" of particles, we thin the output of the single chains of BSL and ACSL: we take the last 10,000 draws from ACSL and sBSL and retain every 10th draw, thus obtaining 1,000 draws that are used to report inference in Table 1. We remind the reader that sBSL fails when initialised at the same starting parameters used for ACSL: therefore to enable some comparison we start sBSL at the ground-truth parameters (this case is denoted $\text{sBSL}_{\text{truth}}$ in the table). Regarding BOLFI, posterior samples were produced by first obtaining 2,000 "acquisition points" in `ELFI` (over which a GP model is fitted), then 10,000 draws are produced via MCMC, and finally chains were thinned to obtain 1,000 draws used for statistical inference. Comparisons between all methods are in Table 1 and Figure 4. Inference results for $\text{sBSL}_{\text{truth}}$, ACSL (both attempts) and BOLFI are similar, however the effective sample size (ESS) for BOLFI is the highest, while the ESS for $\text{sBSL}_{\text{truth}}$ is much lower than for ACSL, as reported in Table 1. While for this case study the exact posterior is unknown, we can speculate that discrepancies in terms of posterior variability between the results obtained with aABC-PMC and those obtained with the other methods can likely be explained by the use of the 20-dimensional summary statistics. For a study on the impact of the summaries dimension in an ABC analysis we refer the reader to Blum et al. (2013). As a further remark, it is important to remember that, unlike standard BSL, ACSL was able to get initialised relatively far from ground-truth parameters and still able to return reasonable inference.

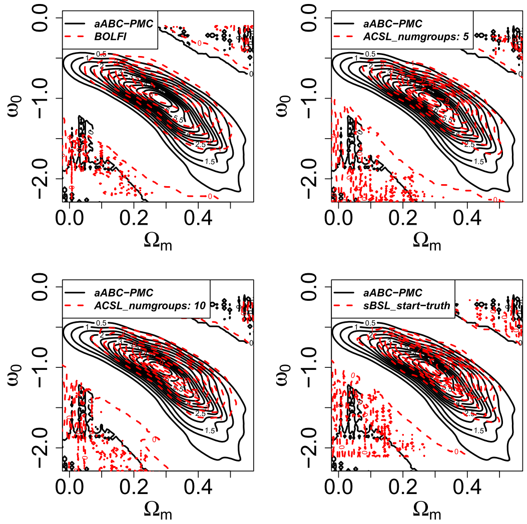


Figure 4: Supernova model. Contour-plot for aABC-PMC method (solid black line), compared with the contour-plots for the remaining methods (dashed red line). In red dashed lines, BOLFI (top left), ACSL with $G = 5$ (top right), ACSL with $G = 10$ (bottom left) and sBSL$_{\text{truth}}$ (bottom right).

6.3 Simple recruitment, boom and bust with highly skewed summaries

Here we consider an example that is discussed in Fasiolo et al. (2018) and An et al. (2020) as it proved challenging due to the highly non-Gaussian summary statistics. The recruitment boom and bust model is a discrete stochastic temporal model that can be used to represent the fluctuation of the population size of a certain group over time. Given the population size N_t and parameter $\theta = (r, \kappa, \alpha, \beta)$, the next value N_{t+1} has the following distribution

$$N_{t+1} \sim \begin{cases} \text{Poisson}(N_t(1+r)) + \epsilon_t, & \text{if} \quad N_t \leq \kappa \\ \text{Binom}(N_t, \alpha) + \epsilon_t, & \text{if} \quad N_t > \kappa \end{cases},$$

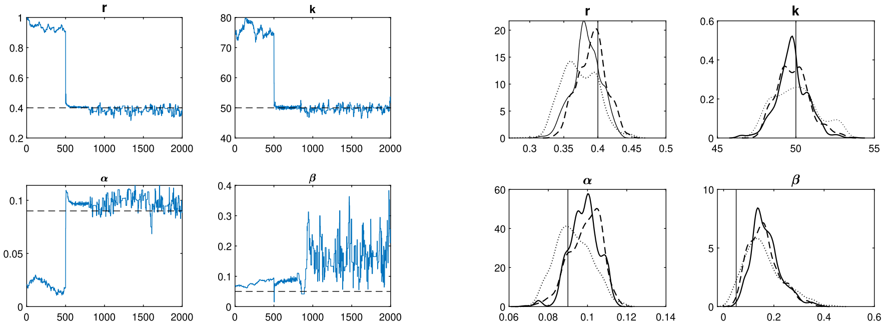
where $\epsilon_t \sim \text{Pois}(\beta)$. The population oscillates between high and low level population sizes for several cycles. Same as in An et al. (2020), true parameters are $r = 0.4$,

	truth	aABC-PMC	sBSL$_{\text{truth}}$	sBSL	ACSL, $G = 5$	ACSL, $G = 10$	BOLFI
Ω_m	0.3	0.31 (0.071, 0.54)	0.313 (0.136, 0.474)	NA	0.316 (0.133, 0.490)	0.317 (0.129, 0.488)	0.289 (0.0765, 0.467)
w_0	-1	-1.05 (-1.95, -0.52)	-1.014 (-1.517, -0.580)	NA	-1.028 (-1.574, -0.607)	-1.047 (-1.502, -0.563)	-0.99 (-1.540, -0.545)
minESS		–	301	NA	781	681	831

Table 1: Supernova model: posterior means (95% high-posterior-density interval) resulting from 1,000 thinned posterior draws from several methods. All chains are initialised at ($\Omega_m = 0.90, w_0 = -0.5$), except for sBSL$_{\text{truth}}$ which is sBSL initialised at ground-truth parameters. The "NA" for sBSL means that the MCMC was unable to move away from the starting location.

$\kappa = 50$, $\alpha = 0.09$ and $\beta = 0.05$ and we assume $N_1 = 10$ a fixed and known constant. This value of β is considered as it gives rise to highly non-Gaussian summaries, and hence it is of interest to test our methodology in such scenario. In fact, the smaller the value of β, the more problematic it is to use synthetic likelihoods. An illustration of the summaries distribution at the true parameters values is in Supplementary Material, together with the prior specifications, the summary statistics employed and other model specifications.

We experiment with two sets of values for the starting parameters: set 1 has $r = 0.8$, $\kappa = 65$, $\alpha = 0.05$, $\beta = 0.07$; set 2 has a more extreme set of values, given by $r = 1$, $\kappa = 75$, $\alpha = 0.02$, $\beta = 0.07$. We always use $M = 200$ (also considered in An et al., 2020). In this case-study we could not experiment with the correlated synthetic likelihoods approach, since the state-of-art generation of Poisson draws requires executing a `while-loop`, where uniform draws are simulated at each iteration. Therefore it is not known in advance how many uniform draws it is necessary to store, and the implementation of correlated SL results inconvenient. When parameters are initialised in set 1, a burnin of 200 iterations aided by MCWM is considered (MCWM is not used after burnin). When initialising from set 2, we use a longer burnin of 500 iterations. During the burnin, as usual we propose parameters using a Gaussian random walk proposal with constant diagonal covariance matrix with diagonal elements $[0.005^2, 0.5^2, 0.001^2, 0.001^2]$. For ASL the burnin was followed by 300 iterations (again this can be set much smaller) using the guided proposals approach, and then further 1,200 iterations using "Haario". BSL was found to diverge to wrong regions of the posterior surface with chains stuck for long periods, for both attempted starting parameters. We therefore implemented the semi-parametric BSL approach from An et al. (2020), thereafter "semiBSL": semiBSL is a robustified version of BSL to address the case of non-Gaussian-distributed summary statistics. However, also semiBSL failed when parameters were initialized in the tails of the posterior (i.e. when using the same starting parameters considered above for ASL), meaning that chains were unable to mix, and were stuck in wrong regions, see the Supplementary Material for details. This shows that even a "robustified" version of synthetic likelihoods can be fragile to bad initializations. Therefore, results we report in Figure 5b for both standard BSL and semiBSL are based on chains initialized at the ground-truth parameter values. With ASL, at the end of the initial 500 burnin iterations we notice the characteristic "jump" towards the true parameter values, see Figure 5a. Therefore, ASL is able to produce inference also when initialised at parameters in the tails of the posterior surface, while BSL and semiBSL cannot, at least for this example. Traces for the failing semiBSL initialised at set 2 are in the Supplementary material. Similarly to the supernova model, we now compute minESS values. Using 1,000 posterior draws from the run initialised at set 2, we have that with ASL minESS is 49. For BSL, when starting at ground-truth, we have minESS = 35 and for semiBSL minESS = 41. Therefore, the values for semiBSL and ASL are quite similar, despite the fact that ASL is initialised at a less favorable location.



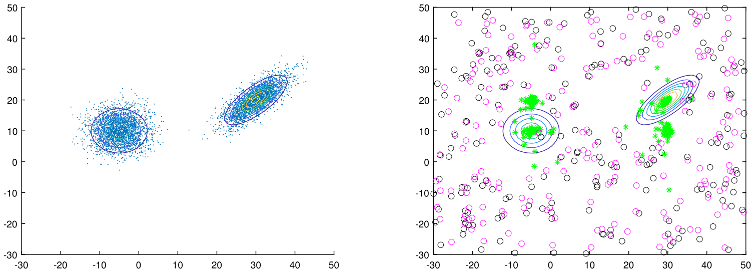
(a) (b)

Figure 5: Boom-and-bust: (a) traces for ASL initialised at set 2. Dashed lines are true parameter values. (b) marginal posteriors from 1,000 draws produced with initialisation via ASL (solid) with starting parameters in set 2; with BSL (dashed) and semiBSL (dotted) both initialised at ground truth parameters (vertical lines).

6.4 A multimodal surface

We show how to run multiple chains employing ASL to rapidly alert the researcher of the existence of multiple modes, at a small computational cost. The toy model is admittedly very simple, but the experiment is expressive enough for our take-home message. We consider a likelihood consisting of a two-components Gaussian mixture $x \sim 0.5\mathcal{N}(\mu_1, \Sigma_1) + 0.5\mathcal{N}(\mu_2, \Sigma_2)$, where each component is two-dimensional. We consider 5,000 observations generated by such mixture with ground truth $\mu_1 = (\mu_1^{(1)}, \mu_1^{(2)}) = (-5, 10)$, $\mu_2 = (\mu_2^{(1)}, \mu_2^{(2)}) = (30, 20)$ and covariance matrices Σ_1 and Σ_2 both having diagonal entries $(4^2, 4^2)$, however Σ_1 is diagonal while Σ_2 has off-diagonal entries both equal to 12. Data are exemplified in Figure 6(a). We assume μ_1 and μ_2 as the only unknowns, and everything else is fixed to ground-truth values. We set independent priors $\mu_1^{(1)} \sim \mathcal{N}(-5, 2^2)$, $\mu_1^{(2)} \sim \mathcal{N}(10, 2^2)$, $\mu_2^{(1)} \sim \mathcal{N}(30, 2^2)$, $\mu_2^{(2)} \sim \mathcal{N}(20, 2^2)$. In our experiments, summary statistics of simulated data are the estimated means of the two mixture components, as obtained by fitting a two-components Gaussian mixture (with known covariances set to ground-truth). Observed summaries are always $s = (-5, 10, 30, 20)$, that is the ground truth means. We used common strategies to get around the well-known "label-switching" issue affecting mixture models: that is whenever during MCMC a vector $(\mu_1^{(1)}, \mu_1^{(2)}, \mu_2^{(1)}, \mu_2^{(2)})$ is proposed, we sort its entries across the mixture components so that the proposed vector has components rearranged to have $\mu_1^{(1)} < \mu_2^{(1)}$ and $\mu_1^{(2)} < \mu_2^{(2)}$. Since we work in the context of synthetic likelihoods, once a simulated dataset is produced at the proposed parameters, we fit a two-components Gaussian mixture to the data as previously mentioned, and the four corresponding estimated means (which are used as summary statistics) are sorted in the same way as the proposed parameters. We use $M = 10$ to approximate the synthetic likelihood and design the following experiment. For a fixed dataset with observed summaries $s = (-5, 10, 30, 20)$

we run 100 independent chains initialised at random locations. We set up a very short burnin consisting of 49 iterations where as usual MCWM is used and a Gaussian random walk sampler is employed, where the noise in the random walk has standard deviation set to 0.2 for each proposed entry in μ_1 and μ_2. This means that during burnin we intentionally induce slow exploration of the posterior surface. We show that, as soon as ASL starts, most chains quickly reach the high-density region of the posterior. Figure 6(b) shows 100 starting values that were randomly sampled uniformly in the 4-dimensional hypercube $[-30, 50]^4$. We notice that after 49 iterations using random walk proposals the draws are still fairly close to the starting value, however one further iteration afterwards, when ASL is initialised, a rapid jump is performed towards the high density region. The clustering of the ASL draws should signal the researcher the existence of more than one mode, and hence inform her of the opportunity to initialise more than one chain for a full-fledged Bayesian inference, by picking the starting values in the clusters determined by ASL.



(a)

(b)

Figure 6: Gaussian-mixture: (a) 5,000 data-points from the likelihood model and contour lines for the latter; (b) contour lines for the likelihood model; black circles are the 100 starting values for the corresponding 100 chains; magenta circles correspond to iteration 49 (last burnin iteration) for each chain; the green asterisks correspond to iteration 50 for each chain, that is the first ASL iteration.

7 Discussion

We have introduced several ways to improve the performance of the computing-intensive synthetic likelihood framework. Firstly, we have developed a sequential strategy to learn a "guided by data" proposal distribution for SL. The resulting sequentially adaptive and guided SL sampler (ASL) helped the chain to rapidly approach the ground truth parameter values. Importantly, for two of the considered case studies (supernova cosmological parameters and recruitment boom-and-bust model), standard SL methods failed when initialized at remote parameter values and when the standard adaptive MCMC strategy by Haario et al. (2001) was employed, whereas ASL helped the chains to rapidly converge to high-posterior regions (remarkably, this happened even for the markedly non-Gaussian summary statistics considered in Section 6.3). In addition, we have shown how to introduce correlation between successive estimates of the synthetic likelihood, calling this approach “correlated synthetic likelihoods”. This should help reducing the variance in the acceptance ratio of Metropolis-Hastings, and indeed we have noticed an increase in the mixing of the chains. We have shown how this correlated SL approach (CSL) can be of help when SL is initialized in the tails of the posterior and how beneficial CSL is in terms of chains mixing. However, CSL is not a silver bullet, and it does not necessarily have to succeed at completely eliminating the possibility for SL getting stuck when badly initialized. However, when it can be implemented, there is no obvious reason to prefer standard SL to CSL. At worst, we conjecture that for very nonlinear transformations of the data following the construction of possibly complex summary statistics (and hence complex transformations of the pseudo-random variates), it may happen that the correlation between successive likelihoods gets destroyed, thus transforming CSL into standard SL. We have challenged CSL with a “perturbed α-stable model” (in Supplementary Material) and even in this case CSL has shown beneficial. Finally, for the g-and-k and supernova examples, we have illustrated how the problem of a difficult initialization for SL can be tackled by using a Bayesian optimization-based approach to likelihood-free inference (Gutmann and Corander, 2016), available in the `ELFI` software (Lintusaari et al., 2018). However, we note further that the BOLFI implementation uses the LCB (lower confidence bound) acquisition function which can be prone to over-explore boundaries of parameter spaces and may in some cases result in a poorly resolved surrogate model. An improved acquisition function based on expected integrated variance introduced by Järvenpää et al. (2019) has been shown to lead to more accurate posterior approximation and it is also available in `ELFI`, although it is typically rather expensive computationally. As a summary, we believe that when a reasonable starting region where to set an initial θ is unknown, BOLFI can likely much more rapidly screen the posterior surface in the search for a promising starting region than a random walk proposal. On the other hand, when the dimension of θ is small as it is often the case in BSL applications (say ≤ 7 parameters), then our approach of producing a small number of random walk proposals followed by a short run of ASL can also be more computationally convenient and generally easy to implement.

The steps taken in this work thus broaden the scope of usage of synthetic likelihood methods and open up new venues for further research on improving applicability of intractable inference.

Supplementary Material

Supplementary Material for “Sequentially guided MCMC proposals for synthetic likelihoods and correlated synthetic likelihoods” (DOI: 10.1214/22-BA1305SUPP; .pdf).

References

Allingham, D., King, R., and Mengersen, K. (2009). “Bayesian estimation of quantile distributions.” *Statistics and Computing*, 19(2): 189–201. MR2486231. doi: https://doi.org/10.1007/s11222-008-9083-x. 1111

An, Z., Nott, D. J., and Drovandi, C. (2020). "Robust Bayesian synthetic likelihood via a semi-parametric approach." *Statistics and Computing*, 30: 543–557. MR4065218. doi: https://doi.org/10.1007/s11222-019-09904-x. 1119, 1121

Andrieu, C., Doucet, A., and Holenstein, R. (2010). "Particle Markov chain Monte Carlo methods." *Journal of the Royal Statistical Society: Series B*, 72(3): 269–342. MR2758115. doi: https://doi.org/10.1111/j.1467-9868.2009.00736.x. 1108

Andrieu, C., Roberts, G. O., et al. (2009). "The pseudo-marginal approach for efficient Monte Carlo computations." *The Annals of Statistics*, 37(2): 697–725. MR2502648. doi: https://doi.org/10.1214/07-AOS574. 1101, 1108, 1112

Andrieu, C. and Thoms, J. (2008). "A tutorial on adaptive MCMC." *Statistics and computing*, 18(4): 343–373. MR2461882. doi: https://doi.org/10.1007/s11222-008-9110-y. 1107

Beaumont, M. A. (2003). "Estimation of population growth or decline in genetically monitored populations." *Genetics*, 164(3): 1139–1160. 1108

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009). "Adaptive approximate Bayesian computation." *Biometrika*, 96(4): 983–990. MR2767283. doi: https://doi.org/10.1093/biomet/asp052. 1116, 1117

Blum, M. G., Nunes, M. A., Prangle, D., Sisson, S. A., et al. (2013). "A comparative review of dimension reduction methods in approximate Bayesian computation." *Statistical Science*, 28(2): 189–208. MR3112405. doi: https://doi.org/10.1214/12-sts406. 1119

Botha, I., Kohn, R., and Drovandi, C. (2021). "Particle methods for stochastic differential equation mixed effects models." *Bayesian Analysis*, 16(2): 575–609. MR4255340. doi: https://doi.org/10.1214/20-ba1216. 1108

Cheng, S. H. and Higham, N. J. (1998). "A modified Cholesky algorithm based on a symmetric indefinite factorization." *SIAM Journal on Matrix Analysis and Applications*, 19(4): 1097–1110. MR1636528. doi: https://doi.org/10.1137/S0895479896302898. 1102

Chevallier, M. and Polarski, D. (2001). "Accelerating universes with scaling dark matter." *International Journal of Modern Physics D*, 10(02): 213–223. MR1889333. doi: https://doi.org/10.1142/S021827180100161X. 1117

Choppala, P., Gunawan, D., Chen, J., Tran, M.-N., and Kohn, R. (2016). "Bayesian Inference for State Space Models using Block and Correlated Pseudo Marginal Methods." *arXiv preprint* arXiv:1612.07072. 1108

Condon, J. and Matthews, A. (2018). "ΛCDM Cosmology for Astronomers." *Publications of the Astronomical Society of the Pacific*, 130(989): 073001. 1117

Dahlin, J., Lindsten, F., Kronander, J., and Schön, T. B. (2015). "Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables." *arXiv preprint* arXiv:1511.05483. 1108

Dehideniya, M., Overstall, A. M., Drovandi, C. C., and McGree, J. M. (2019). "A synthetic likelihood-based Laplace approximation for efficient design of biological processes." *arXiv preprint* arXiv:1903.04168. 1101

Del Moral, P., Doucet, A., and Jasra, A. (2012). "An adaptive sequential Monte Carlo method for approximate Bayesian computation." *Statistics and Computing*, 22(5): 1009–1020. MR2950081. doi: https://doi.org/10.1007/s11222-011-9271-y. 1116

Deligiannidis, G., Doucet, A., and Pitt, M. K. (2018). "The correlated pseudo-marginal method." *Journal of the Royal Statistical Society: Series B*, 80(5): 839–870. MR3874301. doi: https://doi.org/10.1111/rssb.12280. 1108

D'Errico, J. (2015). "nearestSPD." https://www.mathworks.com/matlabcentral/fileexchange/42885-nearestspd, MATLAB Central File Exchange. Retrieved October 8, 2021. 1102

Ding, P. (2016). "On the conditional distribution of the multivariate t distribution." *The American Statistician*, 70(3): 293–295. MR3535516. doi: https://doi.org/10.1080/00031305.2016.1164756. 1105

Drovandi, C. and Pettitt, A. (2011). "Likelihood-free Bayesian estimation of multivariate quantile distributions." *Computational Statistics & Data Analysis*, 55(9): 2541–2556. MR2802334. doi: https://doi.org/10.1016/j.csda.2011.03.019. 1111

Engblom, S., Eriksson, R., and Widgren, S. (2020). "Bayesian epidemiological modeling over high-resolution network data." *Epidemics*, 32. doi: https://doi.org/10.1016/j.epidem.2020.100399. 1101

Fasiolo, M., Wood, S. N., Hartig, F., Bravington, M. V., et al. (2018). "An extended empirical saddlepoint approximation for intractable likelihoods." *Electronic Journal of Statistics*, 12(1): 1544–1578. MR3806432. doi: https://doi.org/10.1214/18-ejs1433. 1119

Fearnhead, P. and Prangle, D. (2012). "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation." *Journal of the Royal Statistical Society: Series B*, 74(3): 419–474. MR2925370. doi: https://doi.org/10.1111/j.1467-9868.2011.01010.x. 1111

Genovese, C. R., Freeman, P., Wasserman, L., Nichol, R. C., and Miller, C. (2009). "Inference for the dark energy equation of state using Type IA supernova data." *The Annals of Applied Statistics*, 144–178. MR2668703. doi: https://doi.org/10.1214/08-AOAS229. 1117

Golightly, A., Bradley, E., Lowe, T., and Gillespie, C. (2019). "Correlated pseudo-marginal schemes for time-discretised stochastic kinetic models." *Computational Statistics & Data Analysis*, 136: 92–107. MR3944674. doi: https://doi.org/10.1016/j.csda.2019.01.006. 1108

Gutmann, M. U. and Corander, J. (2016). "Bayesian optimization for likelihood-free inference of simulator-based statistical models." *The Journal of Machine Learning Research*, 17(1): 4256–4302. MR3555016. 1100, 1110, 1124

Guy, J., Sullivan, M., Conley, A., Regnault, N., Astier, P., Balland, C., Basa, S., Carlberg, R., Fouchez, D., Hardin, D., et al. (2010). "The Supernova Legacy Survey 3-year sample: Type Ia supernovae photometric distances and cosmological constraints." *Astronomy & Astrophysics*, 523: A7. 1116

Haario, H., Saksman, E., and Tamminen, J. (2001). "An adaptive Metropolis algorithm." *Bernoulli*, 7(2): 223–242. MR1828504. doi: https://doi.org/10.2307/3318737. 1100, 1107, 1112, 1114, 1123

Higham, N. (2015). "Modified Cholesky factorization." https://github.com/higham/modified-cholesky. 1102

Higham, N. J. (1988). "Computing a nearest symmetric positive semidefinite matrix." *Linear algebra and its applications*, 103: 103–118. MR0943997. doi: https://doi.org/10.1016/0024-3795(88)90223-6. 1102

Hoffman, M. D. and Gelman, A. (2014). "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research*, 15(1): 1593–1623. MR3214779. 1111

Huterer, D. and Turner, M. S. (2001). "Probing dark energy: Methods and strategies." *Physical Review D*, 64(12): 123527. 1117

Järvenpää, M., Gutmann, M. U., Pleska, A., Vehtari, A., and Marttinen, P. (2019). "Efficient acquisition rules for model-based approximate Bayesian computation." *Bayesian Analysis*, 14(2): 595–622. MR3934099. doi: https://doi.org/10.1214/18-BA1121. 1124

Järvenpää, M., Gutmann, M. U., Vehtari, A., and Marttinen, P. (2020). "Parallel Gaussian process surrogate Bayesian inference with noisy likelihood evaluations." *Bayesian Analysis*. MR4194277. doi: https://doi.org/10.1214/20-BA1200. 1111

Jennings, E. and Madigan, M. (2017). "astroABC: an approximate bayesian computation sequential Monte Carlo sampler for cosmological parameter estimation." *Astronomy and computing*, 19: 16–22. 1116, 1117

Karabatsos, G. and Leisen, F. (2018). "An approximate likelihood perspective on ABC methods." *Statistics Surveys*, 12: 66–104. MR3812816. doi: https://doi.org/10.1214/18-SS120. 1101

Kessler, R., Bernstein, J. P., Cinabro, D., Dilday, B., Frieman, J. A., Jha, S., Kuhlmann, S., Miknaitis, G., Sako, M., Taylor, M., et al. (2009). "SNANA: A public software package for supernova analysis." *Publications of the Astronomical Society of the Pacific*, 121(883): 1028. 1116

Kokko, J., Remes, U., Thomas, O., Pesonen, H., and Corander, J. (2019). "PYLFIRE: Python implementation of likelihood-free inference by ratio estimation." *Wellcome Open Research*, 4(197): 197. 1101

Linder, E. V. (2003). "Exploring the expansion history of the universe." *Physical Review Letters*, 90(9): 091301. 1117

Lintusaari, J., Vuollekoski, H., Kangasrääsiö, A., Skytén, K., Järvenpää, M., Gutmann, M., Vehtari, A., Corander, J., and Kaski, S. (2018). "ELFI: Engine for Likelihood-

Free Inference." *Journal of Machine Learning Research*, 19(16). MR3862423. 1100, 1110, 1124

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). "Markov chain Monte Carlo without likelihoods." *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328. 1107

Nott, D. J., Drovandi, C., and Kohn, R. (2019). "Bayesian inference using synthetic likelihood: asymptotics and adjustments." *arXiv preprint* arXiv:1902.04827. 1118

Ong, V. M.-H., Nott, D. J., Tran, M.-N., Sisson, S. A., and Drovandi, C. C. (2018). "Likelihood-free inference in high dimensions with synthetic likelihood." *Computational Statistics & Data Analysis*, 128: 271–291. MR3850637. doi: https://doi.org/10.1016/j.csda.2018.07.008. 1101

Papamakarios, G., Sterratt, D. C., and Murray, I. (2019). "Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows." In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, 837–848. 1107

Picchini, U. and Everitt, R. G. (2019). "Stratified sampling and bootstrapping for approximate Bayesian computation." *arXiv preprint* arXiv:1905.07976. 1107

Picchini, U. and Forman, J. L. (2019). "Bayesian inference for stochastic differential equation mixed effects models of a tumour xenography study." *Journal of the Royal Statistical Society: Series C*, 68(4): 887–913. MR4002376. 1101

Picchini, U., Simola, U., and Corander, J. (2022). "Supplementary Material for "Sequentially guided MCMC proposals for synthetic likelihoods and correlated synthetic likelihoods"." *Bayesian Analysis*. doi: https://doi.org/10.1214/22-BA1305SUPP. 1101

Prangle, D. (2017). "gk: An R Package for the g-and-k and generalised g-and-h Distributions." *arXiv preprint* arXiv:1706.06889. 1111

Prangle, D. et al. (2017). "Adapting the ABC distance function." *Bayesian Analysis*, 12(1): 289–309. MR3620131. doi: https://doi.org/10.1214/16-BA1002. 1110

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2018). "Bayesian synthetic likelihood." *Journal of Computational and Graphical Statistics*, 27(1): 1–11. MR3788296. doi: https://doi.org/10.1080/10618600.2017.1302882. 1100, 1101, 1102, 1103, 1110

Rasmussen, C. E. and Williams, C. (2006). *Gaussian processes in machine learning*. The MIT Press. MR2514435. 1110

Rayner, G. D. and MacGillivray, H. L. (2002). "Numerical maximum likelihood estimation for the g-and-k and generalized g-and-h distributions." *Statistics and Computing*, 12(1): 57–75. MR1877580. doi: https://doi.org/10.1023/A:1013120305780. 1111

Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer Science & Business Media. MR2080278. doi: https://doi.org/10.1007/978-1-4757-4145-2. 1104

Schön, T. B., Svensson, A., Murray, L., and Lindsten, F. (2018). "Probabilistic learning of nonlinear dynamical systems using sequential Monte Carlo." *Mechanical Systems and Signal Processing*, 104: 866–883. 1108

Simola, U., Cisewski-Kehe, J., Gutmann, M. U., Corander, J., et al. (2020). "Adaptive approximate Bayesian computation tolerance selection." *Bayesian Analysis*. MR4255336. doi: https://doi.org/10.1214/20-ba1211. 1116, 1117, 1118

Sisson, S. A. and Fan, Y. (2011). *Handbook of Markov chain Monte Carlo*, chapter Likelihood-free MCMC. Chapman & Hall/CRC, New York. MR2858454. 1101

Sisson, S. A., Fan, Y., and Beaumont, M. (2018). *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC. MR3889281. 1099

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. (2021). "Likelihood-free inference by ratio estimation." *Bayesian Analysis*. doi: https://doi.org/10.1214/20-BA1238. 1101

Tran, M.-N., Kohn, R., Quiroz, M., and Villani, M. (2016). "The block pseudo-marginal sampler." *arXiv preprint* arXiv:1603.02485. 1108, 1109

Tripathi, A., Sangwan, A., and Jassal, H. (2017). "Dark energy equation of state parameter and its evolution at low redshift." *Journal of Cosmology and Astroparticle Physics*, 2017(06): 012. MR3673419. doi: https://doi.org/10.1088/1475-7516/2017/06/012. 1117

Usmani, A., Ghosh, P., Mukhopadhyay, U., Ray, P., and Ray, S. (2008). "The dark energy equation of state." *Monthly Notices of the Royal Astronomical Society: Letters*, 386(1): L92–L95. 1117

Vihola, M. (2012). "Robust adaptive Metropolis algorithm with coerced acceptance rate." *Statistics and Computing*, 22(5): 997–1008. MR2950080. doi: https://doi.org/10.1007/s11222-011-9269-5. 1107

Warton, D. I. (2008). "Penalized normal likelihood and ridge regularization of correlation and covariance matrices." *Journal of the American Statistical Association*, 103(481): 340–349. MR2394637. doi: https://doi.org/10.1198/016214508000000021. 1118

Wetterich, C. (2004). "Phenomenological parameterization of quintessence." *Physics Letters B*, 594(1-2): 17–22. MR2000282. doi: https://doi.org/10.1103/PhysRevLett.90.231302. 1117

Wiqvist, S., Golightly, A., McLean, A. T., and Picchini, U. (2021). "Efficient inference for stochastic differential equation mixed-effects models using correlated particle pseudo-marginal algorithms." *Computational Statistics & Data Analysis*, 157: 107151. MR4192029. doi: https://doi.org/10.1016/j.csda.2020.107151. 1108

Wood, S. N. (2010). "Statistical inference for noisy nonlinear ecological dynamic systems." *Nature*, 466(7310): 1102. 1100, 1101, 1102, 1103

Acknowledgments

We would like to thank Christopher Drovandi for useful feedback on an earlier draft of this paper. We also thank the anonymous reviewers for helping improve the paper, and especially the reviewer who suggested storing the sample mean of simulated summaries $\bar{s}^*$ into $\mathcal{D}$.