Likelihood-Free Inference with Deep Gaussian Processes

Alexander Aushev [*]
Aalto University
alexander.aushev@aalto.fi

Henri Pesonen [†]
University of Oslo
henri.pesonen@medisin.uio.no

Markus Heinonen [*]
Aalto University
markus.o.heinonen@aalto.fi

Jukka Corander [†]
University of Oslo
jukka.corander@medisin.uio.no

Samuel Kaski [*‡]
Aalto University and University of Manchester
samuel.kaski@aalto.fi

Abstract

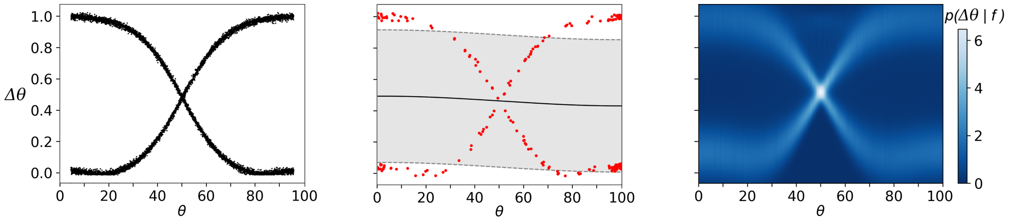
In recent years, surrogate models have been successfully used in likelihood-free inference to decrease the number of simulator evaluations. The most data-efficient solution for this task has been achieved by Bayesian Optimization with Gaussian Processes (GPs). While this combination works well for unimodal target distributions, it appears restrictive in more irregular cases. On the other hand, neural network approaches are extremely adaptable given sufficient data, which are rarely available when working with computationally expensive simulators. In this extended abstract, we address a trade-off between data-efficiency and flexibility by proposing a Deep Gaussian Process (DGP) surrogate model that can handle more irregularly behaved target distributions with few simulator evaluations. Our experiments show how DGPs can outperform GPs on objective functions with multimodal distributions, maintaining a comparable performance in unimodal cases. At the same time, DGPs in general require much fewer data to achieve the same performance as Mixture Density Networks and Masked Autoregressive Flows. This confirms that DGPs as surrogate models for Bayesian Optimization provide a good tradeoff between data-efficiency and flexibility for likelihood-free inference with computationally intensive simulators.

1 Introduction

In likelihood-free inference (LFI) we aim to infer the generative parameters θ of an observed dataset $\mathbf{x}_{\text{obs}}$, whose likelihood $p(\mathbf{x}_{\text{obs}}|\theta)$ is intractable which prevents conventional statistical parameter estimation [4]. Instead, we assume we can simulate new data $\mathbf{x}_\theta \sim p(\mathbf{x}|\theta)$ from arbitrary parameter values, and we relate the probability of a parameter to how similar its simulated dataset is to the observed one [10], measured via a discrepancy function. This simulator-based LFI has been proposed under the names of approximate Bayesian computation (ABC) [2], indirect inference [7] and synthetic likelihood [20] in domains ranging from genetics to economics and ecology [7].

[*] Helsinki Institute for Information Technology, Department of Computer Science, Aalto University, Finland
[†] Department of Biostatistics, University of Oslo, Norway
[‡] Department of Computer Science, University of Manchester, UK

NeurIPS 2020 Workshop on Deep Learning and Inverse Problems, virtual.



(a) True $\Delta\theta$ (b) Vanilla GP fit (c) DGP fit

Figure 1: **(a)** An example of a multimodal target distribution: the discrepancy $\Delta\theta$ is bimodal for each value of the parameter θ. **(b)** Vanilla GP as a surrogate distribution is unable to fit the target (red: observed data; line and shading: GP prediction with uncertainty). **(c)** Deep GP surrogate is able to model the bimodal target distribution accurately.

One approach to LFI is to find a data-efficient surrogate to the discrepancy function, which can be used to derive a proxy for the unknown likelihood. When the simulator call-time is long, the number of simulator queries has to be limited for computational reasons. Previous research by Gutmann and Corander [9] addressed this issue by using Gaussian Processes (GPs) as the discrepancy surrogates and applying Bayesian Optimization (BO) as an efficient search strategy. This approach drastically reduced the number of simulations required for accurate inference.

However, inferring simulator-based statistical models often requires approximating distributions which may exhibit too complex behavior to be adequately represented by GPs, especially in the high-dimensional case. In particular, multimodal distributions represent a serious problem for the current LFI methods (Figure 1). Sequential neural density estimation methods, based on Masked Autoregressive Flows (MAFs) and Mixture Density Networks (MDNs) [17, 16], use powerful deep network models to address this issue. However, to our knowledge, no current method is flexible enough to handle multimodal target distributions with only few hundreds of samples. Our research hypothesis is that by using highly flexible Deep Gaussian Processes (DGPs) as surrogates in BO, we can simultaneously model both uni- and multimodal target distributions with non-stationarity and heteroscedasticity.

2 Likelihood-free inference with deep Gaussian processes

After observing $\mathbf{X}_{\text{obs}}$, traditionally the likelihood-free inference of the parameter θ is based on the metric distance $d(\cdot,\cdot)$ between the summarized observed and synthetic datasets

$$\Delta_\theta = d\big(\mathbf{s}(\mathbf{X}_{\text{obs}}), \mathbf{s}(\mathbf{X}_\theta)\big) \in \mathbb{R}_+, \tag{1}$$

where $\mathbf{s}(\cdot)$ is a vector of summarising functions with lower dimension than the datasets. This is used in a kernel function $\phi(\cdot)$, such as RBF or uniform, to approximate the likelihood

$$p(\mathbf{s}_{\text{obs}}|\theta) \approx \mathrm{E}_{p(\mathbf{X}_\theta|\theta)}[\phi(\Delta_\theta)]. \tag{2}$$

Bayesian Optimization. The task of finding θ that minimizes the discrepancy Δ_θ is in general a non-convex search problem. To minimize the number of generated datasets $\mathbf{X}_\theta$ we turn to BO with DGP surrogates that are capable of handling multimodal and non-stationary discrepancy distributions. Here, we describe a single-layer LV-GP architecture of the LV-DGP surrogate [18], which consists of a LV layer followed by a GP prior (Equation 3) and with the Gaussian likelihoods (Equation 4)

$$f([\theta, w]) \sim \mathcal{GP}(m([\theta, w]), k([\theta, w], [\theta', w'])) \tag{3}$$

$$p(\Delta\theta|f, w) = \mathcal{N}(\Delta\theta|f([\theta, w]), \sigma^2), \tag{4}$$

The acquisition function $A^t(\theta)$ for the DGP surrogate should also allow exploration of objectives with multimodal uncertainties. Since we do not make any assumptions about the form of the discrepancy marginals, we propose a simple quantile-based modification of the lower confidence bound selection criterion (LCBSC) [3] for selecting a new parameter point $\theta^{t+1} = \operatorname{argmin}_\theta \{A^t(\theta)\}$:

$$\mathrm{A}^t(\theta) = \mu_q(\theta) - \sqrt{\eta_t^2 \cdot \nu_q(\theta)} \tag{5}$$

$$\mu_q(\theta) = \mathrm{E}[\{f(\theta^i) : f(\theta^i) \leq Q(\epsilon_q)\}_{i=1}^N] \tag{6}$$

$$\nu_q(\theta) = \mathrm{var}[\{f(\theta^i) : f(\theta^i) \leq Q(\epsilon_q)\}_{i=1}^N], \tag{7}$$

where η_t^2 is a user-defined tuning parameter, N is the number of samples, μ_q and ν_q are the mean and the variance of DGP posterior samples below a quantile threshold ϵ_q, and $Q(\cdot)$ is the quantile function.

Likelihood approximation. Finally, we replace the expectation in the likelihood approximation from Equation 2 with an estimated probabilistic model of $\Delta\theta$. The quantile threshold conditioning on the posterior mean and the variance allows to focus on the representation of lower discrepancy regions in multimodal distributions

$$p(\mathbf{s}_{\mathrm{obs}}|\theta) \propto F\left(\frac{\epsilon - \mu_q(\theta)}{\sqrt{\nu_q(\theta) + \sigma^2}}\right), \tag{8}$$

where $F(\cdot)$ is the cumulative distribution function of Gaussian with the mean 0 and variance 1. In summary, we introduced a way how DGP surrogates can handle irregularly behaved marginal distributions in the context of BO for the LFI problem, by proposing a quantile-based likelihood approximation and an acquisition rule.

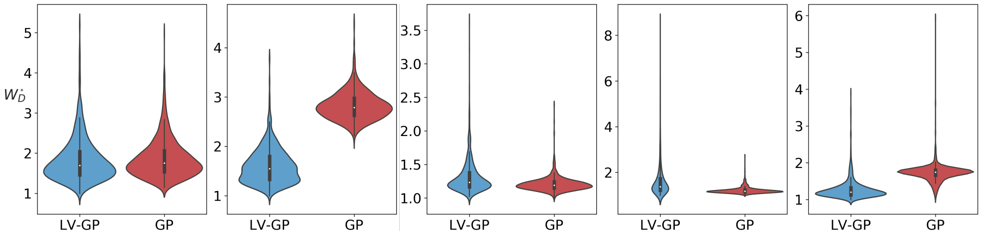
3 Experiments

We study the merits of DGP surrogates in BO, compared to vanilla GPs, masked autoregressive flows (MAFs) [17] and mixture density networks (MDNs) [16], first in illustrative demonstrations and then in two case studies. One-dimensional toy examples (TE) represent three types of objective functions: non-stationary (TE1), multimodal (TE2) and heteroscedastic (TE3). Birth-death model (BDM) describes tuberculosis transmission in San Francisco bay area, as formulated in [12]. Navigation World (NW) model [1] describes an inverse-reinforcement learning problem, where the goal is to approximate the multidimensional distribution over the parameters of the Q-learning agent's [14] reward function operating on the NW map. We report empirical scaled Wassertein distance between surrogate posteriors and the ground truth posterior across 1000 runs with different random seeds. The ground truth is estimated numerically by rejection sampling ABC with 10^8 simulations, retaining 0.1% samples with the lowest discrepancy. More details on the simulators and experimental setups can be found in the supplementary material.

3.1 Results

In all studies, LV-GP architecture of DGPs is either better or on the same level than vanilla GP in approximating the posterior, as shown in Figure 2. The clearest advantage is shown on the TE2 and NW cases, where DGPs handle multimodality, while vanilla GP cannot. The results of TE1 indicate that even though the DGP model can provide a better approximation of the whole function, it is sufficient to accurately represent the function at global minima. In TE3 the difference between posteriors is negligible due to a significant and complex noise component of the example, with DGPs having a higher variance. Such performance of both models on TE3 was expected, and demonstrates that DGPs, as a more flexible model, have greater flexibility than traditional GPs. The BDM performance of DGP is comparable to GPs, but offers no clear advantage over GPs because of higher variance. In summary, DGPs unlike GPs can work with both multimodal and unimodal uncertainties, making them especially suitable for cases when no prior information about the form of the uncertainty is available.

The results in Table 1 show that both DGPs and GPs produced better approximations of the posterior than MDNs and MAFs (even with the increased simulation budget). Only in the BDM case, MDNs outperformed DGP (but not GP), and in TE2 MAF outperformed GPs (but not DGPs). MAFs and MDNs are trying to model the likelihood directly in contrast to DGPs that model the discrepancy. The



(a) TE1 (b) TE2 (c) TE3 (d) BDM (e) NW

Figure 2: Scaled Wasserstein distance between the surrogate models (GP and LV-GP, an instance of DGP) posteriors and the true posterior of θ; the smaller the distance the better is the quality of approximations. The DGP approximations of the true posterior are better on multimodal TE2 (b) and NW (c) examples, maintaining comparable performance on the rest of the cases. The white dot on the violin plot is the median, the black bar is the interquartile range, and lines stretched from the bar show lower/upper adjacent values.

Table 1: LV-DGP models showed the best results in four out of five test cases (columns) across all alternative models (rows). The performance was measured with 95% confidence interval (CI) of the scaled Wasserstein distance between the surrogate model posterior and the true posterior of θ, across 1000 runs. The best results in each column are highlighted in bold. * denotes models that used 1000 total observations instead of 200 for the sample-efficiency comparison.

Model	**TE1**	**TE2**	**TE3**	**BDM**	**NW**
GP	(1.89, 1.95)	(2.65, 2.68)	(1.2, 1.21)	**(1.23, 1.25)**	(1.67, 1.7)
1GP	(1.84, 1.9)	(2.1, 2.22)	(1.19, 1.2)	(1.51, 1.59)	(1.33, 1.35)
3GP	(1.86, 1.92)	(2.03, 2.06)	**(1.18, 1.19)**	(1.49, 1.57)	(1.31, 1.33)
LV-GP	(1.83, 1.89)	**(1.6, 1.64)**	(1.23, 1.26)	(1.51, 1.61)	**(1.24, 1.29)**
LV-2GP	**(1.82, 1.88)**	(1.68, 1.72)	(1.23, 1.25)	(1.47, 1.55)	(1.25, 1.29)
LV-3GP	(1.85, 1.9)	(1.7, 1.74)	(1.22, 1.24)	(1.5, 1.6)	(1.26, 1.29)
MAF	(10.44, 14.47)	(1.99, 2.02)	(62.71, 84.58)	(2.03, 2.16)	(2.37, 2.5)
MAF*	(13.66, 18.45)	(2.02, 2.04)	(59.13, 79.92)	(1.79, 1.88)	(2.29, 2.42)
MDNs	(8.66, 11.7)	(15.63, 18.16)	(14.5, 22.28)	(1.38, 1.4)	(1.8, 1.83)
MDNs*	(12.95, 17.62)	(29.37, 34.6)	(36.35, 51.16)	(1.38, 1.4)	(1.8, 1.84)

former is a more general and harder problem, that requires much more observations with the benefit of not having to retrain the model if the observed data is changed. In summary, both considered alternatives have the necessary flexibility to show good performance on the considered cases, however, they require significantly more data than DGPs, making them unsuitable for modelling irregularly behaved distributions in a small data setting. Therefore, DGPs is the preferable candidate for doing LFI with computationally expensive simulators.

4 Discussion

We introduced a novel method for statistical inference when the likelihood is not available, but drawing samples from a simulator is possible although computationally intensive. The introduced method is an extension of BOLFI [9] where we have used DGP surrogates instead of GP surrogates to model the relationship of the parameters and the stochastic discrepancy between observed data and simulated data. The proposed extension retains the active learning property of BOLFI so that the posterior distribution is sought out with as few samples as possible. The flexibility of the DGPs improved the resulting posterior approximations in cases where flexibility was required and otherwise the observed performance was similar in both cases. Especially good improvements were observed in cases where the distribution of the discrepancy was multimodal, i.e. in cases where GP is known to perform poorly as an estimator.

The improvements from using DGP surrogates come with increased computational cost, which is negligible for computationally heavy simulators. DGPs also had a higher variance in a unimodal higher dimensional example. Even though data-efficiency experiments indicated that the DGP variance improves with more observations, a major contribution to this high variance is likely related to the ability to model multimodality. Comparison methods, that showed this ability as well, had similar variance in the unimodal case. The best neural density alternatives were outperformed by DGPs in a grand majority of cases, providing better approximations with fewer available data. We recommend using DGPs in cases with complicated target distributions, where their more expressive surrogates are needed and work better than vanilla GPs.

Acknowledgments and Disclosure of Funding

AA, MH and SK were supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI; grants 328400, 325572, 319264, 292334). HP and JC were supported by European Research Council grant 742158 (SCARABEE, Scalable inference algorithms for Bayesian evolutionary epidemiology). Computational resources were provided by the Aalto Science-IT Project.

References

[1] D. Abel. simple rl: Reproducible reinforcement learning in Python. In *ICLR Workshop on Reproducibility in Machine Learning*, 2019.

[2] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[3] D. D. Cox and S. John. A statistical method for global optimization. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1241–1246. IEEE, 1992.

[4] P. J. Diggle and R. J. Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):193–227, 1984.

[5] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, pages 3440–3448, 2016.

[6] M. Germain, K. Gregor, I. Murray, and H. Larochelle. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889, 2015.

[7] C. Gourieroux, A. Monfort, and E. Renault. Indirect inference. 8:85–118, 1993.

[8] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

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

[10] F. Hartig, J. M. Calabrese, B. Reineking, T. Wiegand, and A. Huth. Statistical inference for stochastic simulation models – theory and application. *Ecology Letters*, 14(8):816–827, 2011.

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

[12] J. Lintusaari, P. Blomstedt, T. Sivula, M. U. Gutmann, S. Kaski, and J. Corander. Resolving outbreak dynamics using approximate Bayesian computation for stochastic birth-death models. *Wellcome Open Research*, 4, 2019.

[13] J. Lintusaari, H. Vuollekoski, A. Kangasrääsiö, K. Skytén, M. Järvenpää, P. Marttinen, M. U. Gutmann, A. Vehtari, J. Corander, and S. Kaski. Elfi: engine for likelihood-free inference. *The Journal of Machine Learning Research*, 19(1):643–649, 2018.

[14] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[15] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017.

[16] G. Papamakarios, D. C, and I. Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019*, volume 89, pages 837–848. PMLR, 4 2019.

[17] G. Papamakarios, T. Pavlakou, and I. Murray. Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2335–2344, Red Hook, NY, USA, 2017. Curran Associates Inc.

[18] H. Salimbeni, V. Dutordoir, J. Hensman, and M. P. Deisenroth. Deep Gaussian processes with importance-weighted variational inference. *arXiv preprint arXiv:1905.05435*, 2019.

[19] P. M. Small, P. C. Hopewell, S. P. Singh, A. Paz, J. Parsonnet, D. C. Ruston, G. F. Schecter, C. L. Daley, and G. K. Schoolnik. The epidemiology of tuberculosis in San Francisco–a population-based study using conventional and molecular methods. *New England Journal of Medicine*, 330(24):1703–1709, 1994.

[20] S. N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010.

A Details of simulators

Demonstrations: non-stationarity, multimodality and heteroscedasticity. The discrepancy function of the first case TE1 is non-stationary with the ground truth $\theta_{\text{true}} = 50$. The simulator function $g_{\text{TE1}}(\theta)$ generates data from the sum of three Gaussian density functions with different means and variances,

$$g_{\text{TE1}}(\theta) = N(\theta|30, 15) + N(\theta|60, 5) + N(\theta|100, 4) + \epsilon, \tag{9}$$

where $\epsilon \sim N(0, 0.005)$.

The second toy example, TE2, has a multimodal discrepancy function with the ground truth $\theta_{\text{true}} = 20$. The simulator function g_{TE2} randomly 'chooses' one of the two logistic functions, and generates the observation according to

$$g_{\text{TE2}}(\theta') = \begin{cases} \frac{\theta'}{1+\theta'} + \epsilon_1, \text{ if } \epsilon_2 \geq 0 \\ \frac{1}{1+\theta'} + \epsilon_1, \text{ if } \epsilon_2 < 0. \end{cases} \tag{10}$$

where $\theta' = \exp(-0.1(\theta - 50))$, $\epsilon_1 \sim N(0, 0.01)$ and $\epsilon_2 \sim N(0, 1)$. The simulator function creates several modes in the observation space, that later transfer to the discrepancy function.

Finally, the discrepancy function of the third case TE3 is heteroscedastic. The output of the simulator is generated as a sum of samples from two different beta distributions, that are defined through the input parameter θ:

$$g_{\text{TE3}}(\theta) = \mathsf{Beta}(\theta + 1, 5) + \mathsf{Beta}(5, \theta + 1). \tag{11}$$

The ground truth of this case is $\theta_{\text{true}} = 20$. Uniform prior on the interval $(0, 100)$ is used, as in the first two cases.

Birth-Death model. Our goal in inferring the BDM parameters is to approximate the posterior distribution $P(R_1, R_2, \beta, t_1|\mathbf{x}_{\text{obs}})$, where $\mathbf{x}_{\text{obs}}$ was generated with the vector of ground-truth parameters (5.88, 0.09, 192, 6.74). These parameter values were inferred by Lintusaari et al. [12] from the summaries of real data [19]. The weighted Euclidean distance was used as the discrepancy measure. The summaries and the corresponding distance weights are shown in Table 2. For detailed interpretation of simulator parameters and summaries, see [12]. The time cost of a simulation is about 5 seconds. We used the same hierarchical priors as Lintusaari et al:

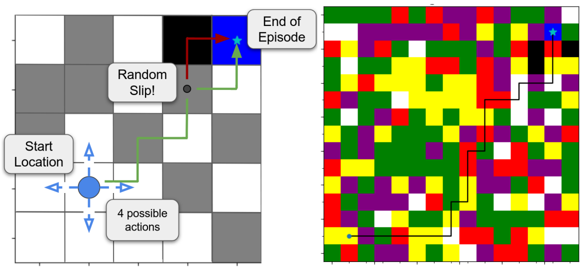
$$\begin{aligned} \theta_{\text{burden}} &\sim N(200, 30) && (12) \\ \theta_{R_1} &\sim \mathsf{Unif}(1.01, 20) && (13) \\ \theta_{R_2}|\theta_{R_1} &\sim \mathsf{Unif}(1.01, (1 - 0.05 \cdot \theta_{R_1})/0.95) && (14) \\ \theta_{t_1} &\sim \mathsf{Unif}(0.01, 30). && (15) \end{aligned}$$

Navigation World. Grid world is a simplified planning environment, and we show how multimodality naturally arises in this kind of setting. Figure 3a shows a simple NW environment. In the NW map, there is an agent that needs to reach the blue goal cell. The map is discrete, and the agent can take four actions that correspond to the directions the agent can go to: up, down, left and right (blue arrows in Figure 3a). Each tile of the map corresponds to a colour that indicates the reward the agent receives from visiting the cell (e.g. +100 for reaching the goal, -500 for the black cell).

The agent always starts at a fixed position. It is first trained on the map, and after a certain number of training episodes we ask it to sample a trajectory. There is no step cost, so the agent is encouraged to explore. The green trajectory on Figure 3a shows one of the optimal paths the agent can learn. However, when we sample the trajectory the environment is stochastic, meaning that the agent can by accident slip into an adjacent cell (red trajectory in Figure 3a). Visiting black cells has a strong fixed negative reward. Since the environment is stochastic and we cannot be sure that the agent will never visit a black tile, this results in a multimodal reward.

Table 2: The summary statistics for the BDM case, their weights in the discrepancy function [12]

Summary	Weight
Number of observations	1
Number of clusters	1
Relative number of singleton clusters	100/0.60
Relative number of clusters of size 2	100/0.4
Size of the largest cluster	2
Mean of the successive differences in size among the four largest clusters	10
Number of months from the first observation to the last in the largest cluster	10
The number of months in which at least one observation was made from the largest cluster	10



(a) NW demonstration

(b) NW experiment environment

Figure 3: (a) In the NW environment the agent (blue circle) starts at a fixed location and can perform four actions: going up, down, left and right. Since the environment is stochastic, the agent may deviate from the optimal green trajectory and end up in a black cell that heavily penalizes the reward. The episode ends once the agent reaches the goal. (b) The NW map that we used in the experiments, with an example observed trajectory shown. Our model needs to infer the reward cell colour parameters, given the summary statistics of the trajectory: in this case 9 turns, 24 steps, and 51 reward.

The experiments were conducted on a more complex map, with tiles of five different colours corresponding to different rewards, shown in Figure 3b. The simulator starts by setting the reward parameters for each colour (five-dimensional vector), and then training the Q-agent for 8,000 episodes in a completely deterministic environment. Once the agent is trained, we sample 5 trajectories and learn their individual summaries: number of turns, number of steps and the reward. The Euclidean distance between the summaries of the sampled and observed trajectories is then used to fit the surrogate model. A trajectory with summaries (9, 24, 51) is illustrated in Figure 3b. Independent uniform priors on the interval (-20, 0) were used for the simulator parameters, whereas true parameter values were (0.0, -1.0, -1.0, -5.0, -10.0). The simulator requires around 40 seconds to sample one observation.

B Experimental setup

In each simulation experiment, we select true parameter values, and use them to produce the observed data set with the simulator. Each experiment is repeated 1,000 times, the runs differing in the choice of random seeds that affect the observations used as initial evidence. We limit the number of total simulator calls to 200 with 100 initial evidence points drawn from the prior before the active learning procedure starts; when targeting computationally heavy simulators this is already plenty. We study how the performance of the surrogate changes with fewer observations, where a half of all observations comes from initial evidence points.

When evaluating goodness of the posterior approximations of θ, we estimate the ground truth posterior numerically by Rejection ABC with 10^8 simulations, and then select 0.1% samples with the lowest discrepancy to represent the posterior distribution. Closeness of the estimated posterior $p_{\text{sur}}(\theta|\mathbf{s}_{\text{obs}})$ to this ground truth reference posterior $p_{\text{ref}}(\theta|\mathbf{s}_{\text{obs}})$ is measured with the empirical Wasserstein distance [5]. We report the scaled (divided by the smallest value) Wasserstein distance W_D^*.

For each simulator we report marginal distributions and corresponding approximations for every parameter, showing how accurately the surrogate posterior matches the true marginal posterior. Additionally, we plot the discrepancy function for one-dimensional cases.

We use a squared-exponential kernel in both GP and DGP. The GP model had kernel lengthscale, variance and added bias component as hyperparameters. Gamma priors were used for all three of them, initialized by the expected value and variance chosen based on initial standardized data. In the DGP model, the lengthscale was set to the square root of the dimension, and the variance to 1. Only the kernel parameters and the likelihood variance (initialized with 0.01) were optimized from their initial values: the final layer using natural gradients (initial step size of 0.01) and the inner layers with the Adam optimizer (initial step size of 0.005) [11]. Scaled conjugate gradient optimization with the maximum number of function evaluations of 50 was used for the GP. All models were implemented in Python with the GPFlow [15] and the GPy [8] packages for DGP and vanilla GP respectively. Engine for Likelihood-Free Inference (ELFI) [13] was used as the platform for the implementations, and the proposed model is available in ELFI for application and further development (elfi.ai).

B.1 Tested models

Additional LV-DGPs architectures. Additional experiments were conducted with multiple architectures of the LV-DGP model, mentioned in the main body of the paper. We used a naming convention where the name of the architecture specifies the exact sequence of layers, e.g. 'LV-3GP' describes the DGP with a LV layer, followed by three GP layers. Importance-weighted variational inference (IWVI) by Salimbeni et al. [18] was used. In all experiments with LV-DGP models, we used 50 inducing points, 5 importance-weighted samples and 20 samples for predictions and gradients. The quantile threshold ϵ_q for the acquisition function and the surrogate likelihood was set to 0.3. Similarly as in the main text, we compare the proposed solution against GPs with the LCBSC acquisition as a baseline (here, denoted simply as 'GP'). We report results for 200 total observations.

Masked autoregressive flows (MAF) [16]. MAF is an implementation of normalizing flow that uses Masked Autoencoder for Distribution Estimation (MADE)[6] as building blocks, where each conditional probability is modelled by a single Gaussian component. In the experiments, we used the architecture with 5 stacked MADEs in the flow and 2 hidden layers, containing 50 hidden units (sequential strategy for assigning degrees to hidden nodes was used) with hyperbolic tangent as an activation function. The model was trained with Adam [11] optimization, using a minibatch size of 100, and a learning rate of $1e^{-4}$. L-2 regularization with coefficient $1e^{-6}$ was added. The training was performed with 300 epochs in 5 batches, with the number of populations equal to the total number of observations divided by the number of batches. We report results for 200 and 1000 total observations.

Mixture density networks (MDN) [17]. MDN is a feedforward neural network that takes the observation $\mathbf{s}_\theta$ as an input and outputs the parameters of a Gaussian mixture over θ. We use an ensemble of 5 MDNs in our experiments with the same architecture: 2 hidden layers with 30 hidden units in each with the hyperbolic tangent activation function. The parameters for optimization and training procedures were the same as for the MAF. We report results for 200 and 1000 total observations.

B.2 Evaluation

Wasserstein distance for assessing the quality of the posterior. Similarity of the estimated surrogate posterior to the ground truth is measured with the empirical Wasserstein distance [5], defined as

$$W_\varepsilon(\mu,\nu) = \int_{\mathcal{X}} u(x)\mathrm{d}\mu(x) + \int_{\mathcal{Y}} v(y)\mathrm{d}\nu(y) - \iota^{\varepsilon}_{U_c}(u,v) \tag{16}$$

where $\mu \in \mathcal{M}_+^1(\mathcal{X})$ and $\nu \in \mathcal{M}_+^1(\mathcal{Y})$ are two measures, defined on metric spaces $\mathcal{X}$ and $\mathcal{Y}$, $(u,v) \in C(\mathcal{X}) \times C(\mathcal{Y})$ of "ground cost" space and $\iota^{\varepsilon}_{U_c}(u,v)$ is an indicator function. In our case, μ

is the posterior of the surrogate model $p_{\text{sur}}(\theta|\mathbf{s}_{\text{obs}})$, and ν the ground truth posterior $p_{\text{ref}}(\theta|\mathbf{s}_{\text{obs}})$. We report the scaled (divided by the smallest value) Wasserstein distance W_D^*. See [5] for further details.