Supplementary Material for Bayesian Experimental Design for Implicit Models by Mutual Information Neural Estimation

A. Additional Information for the Linear Model

A.1. Reference MI Calculation

In order to compute a reference mutual information (MI) value at $\mathbf{d}^*$, we rely on a nested Monte-Carlo sample average of the MI and an approximation to the likelihood $p(\mathbf{y} \mid \mathbf{d}^*, \boldsymbol{\theta})$.

In the setting where we wish to make D independent measurements at $\mathbf{d}^* = [d_1^*, \dots, d_D^*]^\top$, the likelihood factorises as $p(\mathbf{y} \mid \mathbf{d}^*, \boldsymbol{\theta}) = \prod_{j=1}^D p(y_j \mid d_j^*, \boldsymbol{\theta})$. Using this and by means of a sample-average of the marginal $p(\mathbf{y} \mid \mathbf{d}^*)$, we can approximate the MI (as shown in the main text) as follows,

$$I(\mathbf{d}^*) \approx \frac{1}{N} \sum_{i=1}^{N} \left[\log \frac{\prod_{j=1}^D p(y_j^{(i)} \mid d_j^*, \boldsymbol{\theta}^{(i)})}{\frac{1}{M} \sum_{s=1}^{M} \prod_{j=1}^{D} p(y_j^{(i)} \mid d_j^*, \boldsymbol{\theta}^{(s)})} \right], \tag{17}$$

where $y_j^{(i)} \sim p(y_j \mid d_j^*, \boldsymbol{\theta}^{(i)})$, $\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta})$ and $\boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta})$.

In order to be able to compute the MI approximation in (17) for the linear model, we need to built an approximation to the density $p(y_j \mid d_j^*, \boldsymbol{\theta})$. The sampling path for the linear model is given by $y_j = \theta_0 + \theta_1 d_j^* + \epsilon + \nu$, where $\epsilon \sim \mathcal{N}(0, 1)$ and $\nu \sim \Gamma(2, 2)$ are sources of noise. The distribution p_{noise} of $\epsilon + \nu$ is given by the convolution of the densities of ϵ and ν. It could be computed via numerical integration. Here we compute it by a Kernel Density Estimate (KDE) based on $50{,}000$ samples of ϵ and ν. By rearranging the sampling path to $y_j - (\theta_0 + \theta_1 d) = \epsilon + \nu$ we then obtain that $p(y_j \mid d_j, \boldsymbol{\theta}) = p_{\text{noise}}(y_j - (\theta_0 + \theta_1 d))$, allowing us to estimate the MI using (17). We here use $N = 5{,}000$ and $M = 500$.

A.2. Hyper-Parameter Optimisation

We wish to find the optimal neural network size for the 10-dimensional linear model. To do so, we train several neural networks with one hidden layer of sizes $H \in \{50, 100, 150, 200, 250\}$ for 50,000 epochs using learning rates $l_\psi = 10^{-4}$ and $l_d = 10^{-2}$. By generating samples from the prior distribution and the data-generating distribution at the optimal design, we can build a validation set that we use to obtain an estimate of the MI lower bound. Repeating this several times for every trained neural network yields mean MI lower bound estimates, as well as standard deviations, as shown in Figure 7. Given that we wish to obtain the maximum MI lower bound, we see that a neural network of size $H = 150$ is most appropriate in our setting.

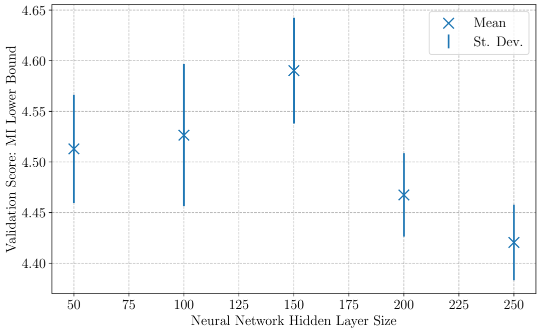

Figure 7. Mean and standard deviation of validation scores (MI lower bound) for the 10-dimensional noisy linear model, using one-layer neural networks with different number of hidden units.

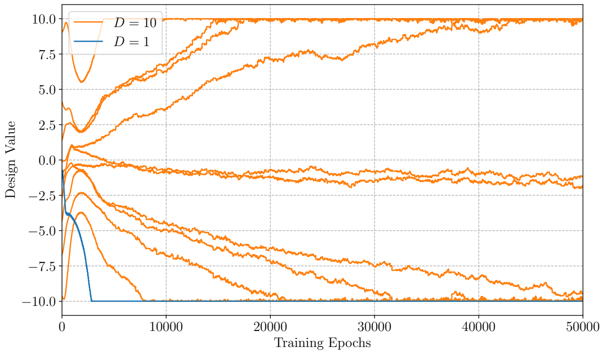

Figure 8. Convergence of the individual design dimensions for the one-dimensional (blue curve) and 10-dimensional (orange curves) linear model. Note how for the 10-dimensional linear model the design dimensions end up in three different clusters.

A.3. Convergence of Designs

In Figure 8 we show the convergence of the individual design dimensions for the one-dimensional and 10-

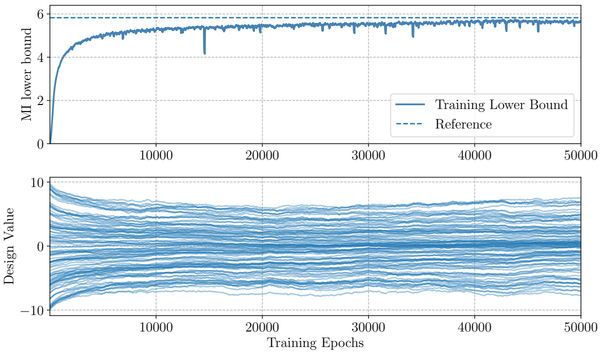

Figure 9. Convergence of the mutual information lower bound (top) and individual design dimensions (bottom) for the 100-dimensional linear model. The dashed line shows a reference mutual information value at the final, optimal design.

dimensional linear model. The one-dimensional design converges quickly to the optimal design, while the different dimensions of the 10-dimensional design converge more slowly and end up in three clusters. Two clusters of optimal designs are at the boundaries where the signal-to-noise ratio is highest, allowing us to estimate the slope of the linear model well. The other cluster is near zero, reducing the effect of the slope and allowing us to estimate the offset better.

A.4. 100-Dimensional Linear Model

In order to test the scalability of our method, we here apply MINEBED to a 100-dimensional version of the linear model. Because of the higher dimensionality of both the data vector $\mathbf{y}$ and the design vector $\mathbf{d}$, we require a neural network with more parameters to obtain a tight bound. We found that a deep neural network, i.e. more layers with less hidden units, seemed to work better than a wide neural network, i.e. less layers with more hidden units. Hence, we opted to use a 5-layered network with 50 hidden units for each layer. We train the network with 10,000 samples and use the Adam optimiser, with initial learning rates of $l_\psi = 10^{-4}$ and $l_\mathbf{d} = 10^{-2}$.

The convergence of the mutual information lower bound and the dimensions of the design vector are shown in the top and bottom of Figure 9, respectively. Also shown is a reference mutual information value computed as explained in Section A.1. The mutual information lower bound converges smoothly to a value that is higher than for the 1-dimensional and 10-dimensional version of the linear model (see the main text). This is intuitive, as more data allows us to gain more information about the model parameters. The final lower bound is relatively tight, as it is close to the reference MI value. As can be seen from the bottom of Figure 9, the different dimensions of the design vector converge to the region $d_j \in [-8, 8]$, with more designs centred around zero. Interestingly, this is a different strategy as for 1 and 10 design dimensions, see Figure 8, where most optimal designs were found to be at the domain boundaries $d_j = -10$ and $d_j = 10$.

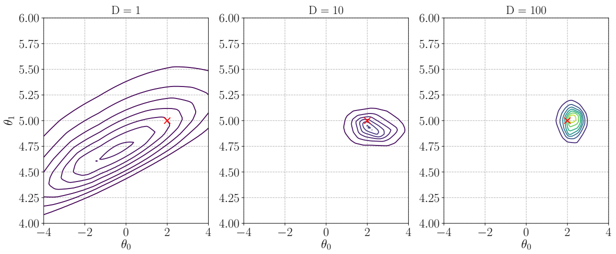

Figure 10. Comparison of posterior densities obtained for the 1-dimensional (left), 10-dimensional (middle) and 100-dimensional (right) noisy linear model. The red cross shows the true model parameters.

In order to ascertain whether or not this experimental design strategy is sensible, we compute reference MI values for different strategies: 1) designs only clustered at the boundaries (MI $= 4.61$), 2) designs clustered at the boundaries and at zero (MI $= 4.83$), akin to the optimal design for 10 dims, 3) equally spaced designs (MI $= 4.91$) and 4) random designs (average of 4.92, standard deviation of 0.08 and maximum value of 5.11 for 100 repeats). These values are smaller than the value we obtained (MI $= 5.78$), indicating that we may have found a (locally) optimal design. This further implies that extrapolating the design strategy from the 10-dimensional to the 100-dimensional linear model is sub-optimal. If the experimenter knows that they can make that many measurements, centring them around zero with some dispersion allows for better parameter estimation.

In Figure 10 we show a comparison of the posterior distributions obtained for the 1-dimensional (left), 10-dimensional (middle) and 100-dimensional (right) noisy linear model. The posterior densities were computed using the trained neural network, as explained in the main text. We find that the posterior distribution becomes narrower, with modes that are closer to the true model parameters, as we increase the design dimensions. This is again intuitive, as more measurements allow us to estimate the model parameters better. Overall, these results show that we can effectively find optimal designs and compute corresponding posterior densities even for 100-dimensional experimental design problems.

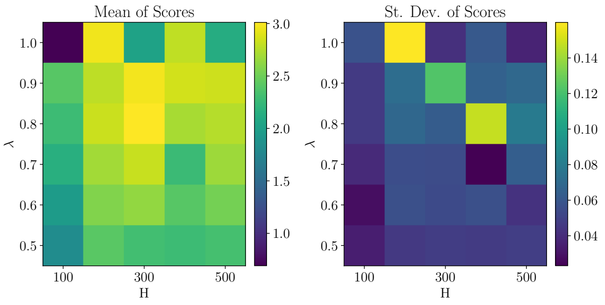

Figure 11. Mean validation scores, including standard deviations, of different neural networks for the 10-dimensional PK model. Tested were neural networks with one hidden layer of different sizes $H \in \{100, 200, 300, 400, 500\}$ and a multiplicative learning rate scheduler of multiplier $\lambda \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

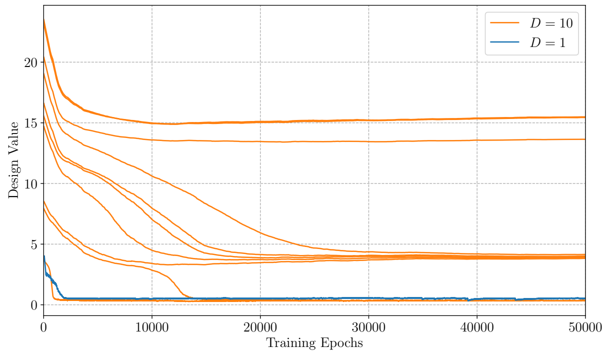

Figure 12. Convergence of the individual design dimensions for the one-dimensional (blue curve) and 10-dimensional (orange curves) PK model. Note how the optimal design clusters for the 10-dimensional PK Model are spread over early, middle and late measurement times.

B. Additional Information for the PK Model

B.1. Reference MI Calculation

Just like for the linear model, we use the nested Monte-Carlo approximation of MI given in (17) to compute a reference MI at $\mathbf{d}^* = [t_1^*, \ldots, t_D^*]^\top$ for the pharmacokinetic (PK) model. Even though computing the MI is intractable, we can write down an equation for the data-generating distribution $p(y_j \mid t_j, \boldsymbol{\theta})$, where $\boldsymbol{\theta} = [k_a, k_e, V]^\top$, as both noise sources are Gaussian,

$$p(y_j \mid t_j, \boldsymbol{\theta}) = \mathcal{N}(y_j; f(t_j, \boldsymbol{\theta}), f(t_j, \boldsymbol{\theta})^2 0.01^2 + 0.1^2), \tag{18}$$

where the function $f(t_j, \boldsymbol{\theta})$ is given by

$$f(t_j, \boldsymbol{\theta}) = \frac{D_V}{V} \frac{k_a}{k_a - k_e} \left[e^{-k_e t_j} - e^{-k_a t_j} \right]. \tag{19}$$

Using this expression and a sample average of $p(\mathbf{y} \mid \mathbf{d}^*)$ we can then compute a numerical approximation of the MI given in (17). We here use $N = 5{,}000$ and $M = 500$.

B.2. Hyper-Parameter Optimisation

We here wish to select the best neural network architecture for the task of finding $D = 10$ optimal designs for the PK model. We test one-layer neural networks with a ReLU activation function after the input layer and hidden units $H \in \{100, 200, 300, 400, 500\}$. Furthermore, we also compare different multipliers $\lambda \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ in a multiplicative learning rate schedule; essentially, this means that the initial learning rates of $l_\psi = 10^{-3}$ and $l_d = 10^{-2}$ are multiplied by λ every 5,000 epochs up until a maximum of 50,000 training epochs. Using prior samples we then generate a validation set of size 50,000 at $\mathbf{d}^*$ and use this to compute the MI lower bound, i.e. the validation score, given the trained neural network with a certain hyper-parameter combination. Doing this for a range of H and λ yields the comparison of validation scores shown in Figure 11. In this figure, we show the mean and standard deviation of validation scores computed with several validation sets, as we can generate synthetic data at will. The overall highest mean validation score is achieved by a neural network with $H = 300$ and $\lambda = 0.8$.

B.3. Convergence of Designs

In Figure 12 we show the convergence of the design vector for the one-dimensional (blue curve) and 10-dimensional (orange curve) PK model. The one-dimensional design converges after around 2,000 training epochs, while the elements of the 10-dimensional design vector converge after roughly 30,000 training epochs. The one-dimensional design ends up at a low design time. Looking at the sampling path of the PK model and doing a Taylor expansion for the exponents shows that this optimal design effectively removes the effect of the elimination rate k_e and estimates the ratio k_a/V. For the 10-dimensional PK model, we have optimal designs at early, middle and late measurement times. Late measurements allow us to reduce the effect of k_a and middle measurements are needed to identify the remaining parameter. These optimal design clusters are intuitive and match the ones found by Ryan et al. (2014).

B.4. Full Posterior Plots

We show the joint and marginal posterior distributions of the model parameters $\boldsymbol{\theta} = [k_a, k_e, V]^\top$ for the one-dimensional ($D = 1$) and 10-dimensional ($D = 10$) PK model in Figures 13 and 14, respectively. As opposed to the one-

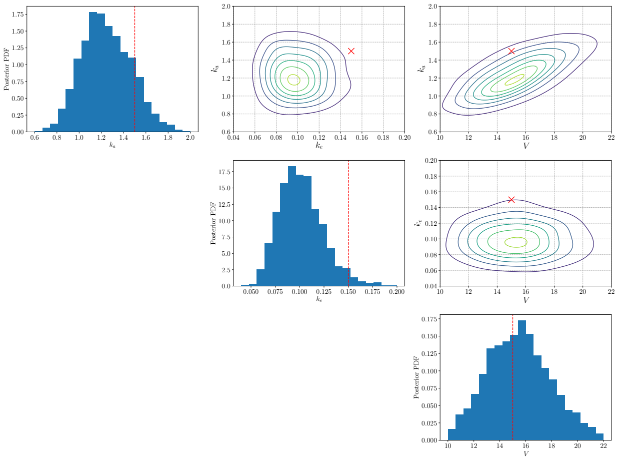

Figure 13. Joint and marginal posterior distributions of the model parameters for the one-dimensional PK model, computed using posterior samples. Shown as red-dotted lines and red crosses are the true model parameters.

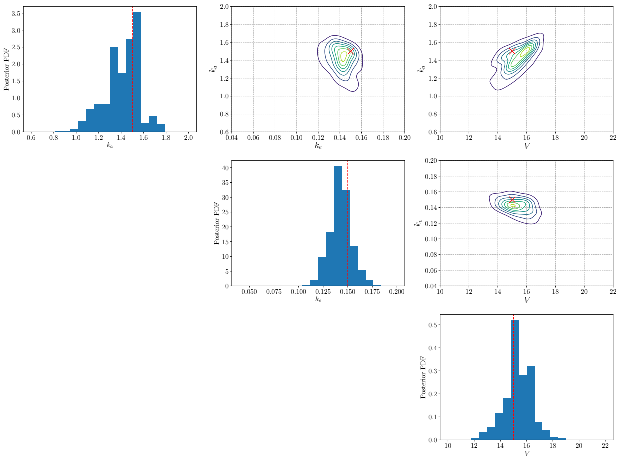

Figure 14. Joint and marginal posterior distributions of the model parameters for the 10-dimensional PK model, computed using posterior samples. Shown as red-dotted lines and red crosses are the true model parameters.

dimensional PK model, the marginal posterior distributions for the 10-dimensional PK model are narrower and closer to the true model parameter values of $\boldsymbol{\theta}_{\text{true}} = [1.5, 0.15, 15]^\top$. Similarly, the mode of the joint posteriors for $D = 10$ are closer to the $\boldsymbol{\theta}_{\text{true}}$ than for $D = 1$. Interestingly, k_a and V are correlated for both $D = 1$ and $D = 10$, allowing us to measure the ratio of k_a/V relatively well.

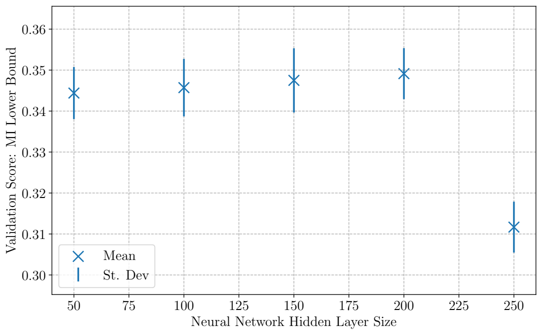

Figure 15. Mean and standard deviation of validation scores (MI lower bound) for the Gas Leak model with $D = 5$, using one-layer neural networks with different number of hidden units.

C. Additional Information for the Gas Leak Model

C.1. Hyperparameter Optimisation

We here aim to find the optimal neural network size for the gas leak model when the number of measurements is $D = 5$. To do so, we train several neural networks with one hidden layer of sizes $H \in \{50, 100, 150, 200, 250\}$ for 50,000 epochs using learning rates $l_\psi = 10^{-3}$ and $l_d = 10^{-2}$. By generating samples from the prior and the likelihood at the optimal design, we can build a validation set that we use to obtain an estimate of the MI lower bound. Repeating this several times for every trained neural network yields mean MI lower bound estimates, as well as standard deviations, as shown in Figure 15. The highest validation score is achieved by a neural network with $H = 200$.

C.2. Additional Plots

To further visualise the design optimisation procedure when gradients of the sampling path are unavailable, we show the GP posterior mean of $\widehat{I}(\mathbf{d}, \boldsymbol{\psi})$ for the gas leak model when we perform $D = 1$ measurement in Figure 16. The BO evaluations occur in locations where the MI lower bound is high, quickly converging to the optimum in the north east corner of the grid.

In Figure 17 we show the posterior density of the gas leak source location $\boldsymbol{\theta}$ when we know that the wind direction is $W_d = 45°$, for both $D = 1$ (left) and $D = 5$ (right). As opposed to the situation where we marginalised out the wind direction, see the main text, the posterior for $D = 5$ is unimodal.

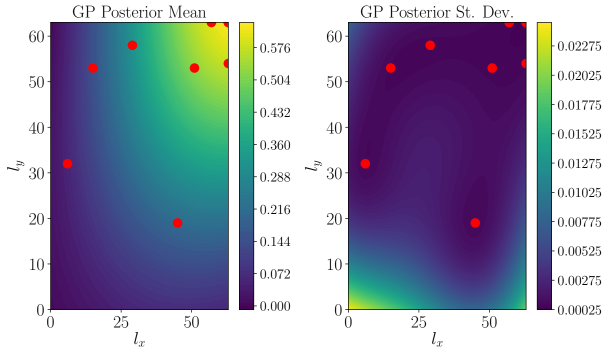

Figure 16. GP posterior mean (left) and standard deviation of the MI lower bound surface for the gas leak model with $D = 1$. Shown as red circles are the BO evaluations of the MI lower bound.

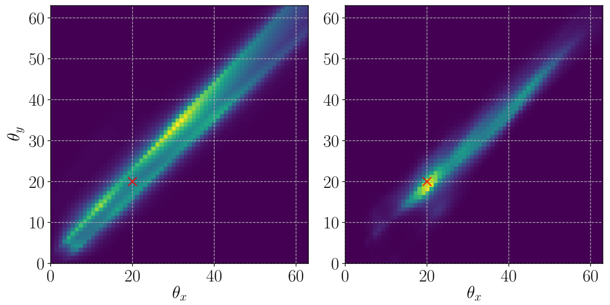

Figure 17. Posterior density for the source locations of the gas leak with one measurement (left) and five measurements (right), assuming we know that the wind points in the direction of $W_d = 45°$. Shown as the red cross is the true gas leak location.

D. Comparison with Bayesian D-Optimality

Different utility functions in Bayesian experimental design tend to be geared towards different purposes. This usually makes a meaningful, direct comparison between utility functions difficult. Mutual information, which we have considered in this work, is used in order to optimally estimate model parameters. Another popular utility function that is used for parameter estimation is the Bayesian D-Optimality (BD-Opt),

$$U(\mathbf{d}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{d})}\left[\frac{1}{\det(\text{cov}(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{d}))}\right], \qquad (20)$$

which is a measure of how precise, on average, the resulting posterior is (Ryan et al., 2016). We here provide a short comparison of optimal designs obtained via mutual information and BD-Opt.

We consider an oscillatory toy model that describes noisy measurements of a stationary waveform $\sin(\omega t)$. The design variable is the measurement time t and our experimental aim is to estimate the frequency ω in an optimal manner. The data-generating distribution is given by

$$p(y \mid \omega, t) = \mathcal{N}(y; \sin(\omega t), \sigma^2_{\text{noise}}), \qquad (21)$$

where $\sigma_{\text{noise}} = 0.1$ is the standard deviation of some measurement noise not depending on t; we here set the true model parameter to $\omega_{\text{true}} = 0.5$. Furthermore, we choose a uniform prior distribution $p(\omega) = \mathcal{U}(\omega; 0, \pi)$ over the model parameter ω. We note that reference posterior densities can be obtained by using the likelihood in (21) and Bayes' rule. This kind of model has also been considered by Kleinegesse et al. (2020) to illustrate their sequential Bayesian experimental design method.

We estimate and optimise the mutual information utility with our MINEBED framework, using a two-layered neural network with 100 hidden units each. The neural network is trained with the Adam optimiser and 10,000 samples as the training set. The initial learning rates are $l_\psi = 5 \times 10^{-3}$ and $l_\mathbf{d} = 10^{-3}$, both multiplied by a factor of 0.9 every $1{,}000$ epochs.

In order to estimate the BD-Opt utility in (20) we require samples from the posterior distribution (which we assume is intractable for now). We here utilise LFIRE (Thomas et al., 2016) to estimate the posterior density for a set of prior parameter samples. Using the posterior density and prior sample pairs, we then use categorical sampling to generate posterior samples (see e.g. Kleinegesse & Gutmann, 2019). These samples can then be used to approximate the determinant of the covariance matrix needed in (20), for a given marginal sample $y \mid t$. We can then approximate (20) with a Monte-Carlo sample average, using $1{,}000$ samples from $p(y|t)$. We optimise the approximated BD-Opt utility by means of Bayesian Optimisation (Shahriari et al., 2016) with a Gaussian Process surrogate model.

Using our MINEBED framework we converge to an optimal design of $t^* = 2.19$, whereas the optimum of the BD-Opt utility is $t^* = 1.66$. We note that the time to converge to the optimum was significantly lower for MINEBED than for the BD-Opt utility with Bayesian Optimisation. While the optimal designs t^* are quite close, the values are still subtlety different, with BD-Opt favouring smaller measurement times. This is because, by definition, BD-Opt penalises posterior multi-modality that leads to larger variance because it only takes into account the covariance matrix. Mutual information on the other hand, is sensitive to multi-modality and therefore does take into account multiple explanations for observations.

This is further reflected in Figure 18, where we show the posterior densities for data obtained using MINEBED (top) and BD-Opt (bottom). The real-world observations at t^*

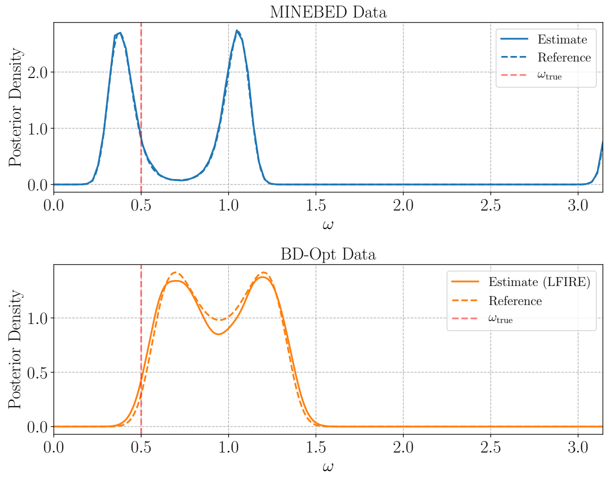

Figure 18. Posterior densities for the oscillatory toy model using MINEBED data (top) and BD-Opt data (bottom). The dashed curves represent reference computations and the vertical dashed, red lines represent the true model parameter value.

were generated using (21) and $\omega_{\text{true}} = 0.5$. The posterior density for MINEBED data was computed via the trained neural network (see the main text), while for BD-Opt data this was done by Gaussian kernel density estimation of the posterior samples. Reference posteriors are shown as dashed lines. Ultimately, the posterior obtained with BD-Opt data has modes that are closer together than for MINEBED data, as we would expect because it favours posteriors with small variances.

References

Kleinegesse, S. and Gutmann, M. U. Efficient Bayesian experimental design for implicit models. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 476–485. PMLR, Apr 2019.

Kleinegesse, S., Drovandi, C., and Gutmann, M. U. Sequential Bayesian Experimental Design for Implicit Models via Mutual Information. *arXiv e-prints [accepted at Bayesian Analysis]*, art. arXiv:2003.09379, 2020.

Ryan, E., Drovandi, C. C., Thompson, H., and Pettitt, A. N. Towards Bayesian experimental design for nonlinear models that require a large number of sampling times. *Computational Statistics and Data Analysis*, 70:45–60, February 2014.

Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, Jan 2016.

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. Likelihood-free inference by ratio estimation. *ArXiv e-prints: 1611.10242*, art. arXiv:1611.10242, Nov 2016.