A Summary of the notations used in the paper

All notations used in the paper are summarized in Table 4.

Notation	Definition
$p(\mathbf{y}\|\mathbf{x})$	Likelihood function implicitly defined by the simulator
$q_\phi(\mathbf{y}\|\mathbf{x})$	Surrogate model of $p(\mathbf{y}\|\mathbf{x})$
$p(\mathbf{y})$	Observed distribution
$q_\theta(\mathbf{y})$	Estimator of $p(\mathbf{y})$
$p(\mathbf{x})$	Unseen source distribution that has generated $p(\mathbf{y})$
$q_\theta(\mathbf{x})$	Surrogate model of $p(\mathbf{x})$
$q_\phi(\mathbf{x}\|\mathbf{y})$	Variational posterior distribution
$\pi(\mathbf{x})$	Proposal distribution used to generate a dataset in order to train $q_\phi(\mathbf{y}\|\mathbf{x})$

Table 4: Summary of the notations used in the paper.

B Properties of the log-marginal estimators $\mathcal{L}_K$ and $\hat{\mathcal{L}}_K$

B.1 Bias of $\mathcal{L}_K(\theta)$

The bias of $\mathcal{L}_K$ is derived from the Jensen's inequality:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}_K\right] =& \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\log p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] && \text{(9a)}\\
=& \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\log p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] && \text{(9b)}\\
\leq& \frac{1}{K}\sum_{k=1}^{K}\log\mathbb{E}\left[p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] && \text{(9c)}\\
=& \log q_\theta(\mathbf{y}) && \text{(9d)}
\end{aligned}$$

where, since the logarithm is strictly concave, the equality in Eq. 9c holds iif the random variable $p(\mathbf{y}|\mathbf{x}_k),\ \mathbf{x}_k = \mathbf{G}_\theta(\boldsymbol{\epsilon}_k)$ is degenerate, that is $\exists!\mathbf{c} : p(\mathbf{x}_k) = \delta_\mathbf{c}(\mathbf{x}_k)$, which is not the case in general.

B.2 Convergence rate of $\mathcal{L}_K(\theta)$

Closely following Nowozin (2018), we show that the bias of the estimator $\mathcal{L}_K(\theta)$ decreases at a rate $\mathcal{O}(\frac{1}{K})$, in particular:

$$\mathbb{E}\left[\mathcal{L}_K(\theta)\right] = \log p(\mathbf{y}) - \frac{1}{K}\frac{\mu_2}{2\mu^2} + \mathcal{O}(\frac{1}{K}),$$

which implies

$$\mathbb{E}\left[\mathcal{L}_K(\theta)\right] = \log p(\mathbf{y}) + \mathcal{O}(\frac{1}{K}).$$

Proof. Let $w := p(\mathbf{y}|\mathbf{x}), \mathbf{x} \sim q_\theta(\mathbf{x})$ and $Y_K := \frac{1}{K}\sum_{i=1}^{K} w_i$. We have $\gamma := \mathbb{E}\left[Y_K\right] = \mathbb{E}\left[w\right] =: \mu$ because the expectation is a linear operator. Let us expand $\log Y_K$ around $\mathbb{E}\left[w\right]$ with a Taylor series:

$$\log Y_K = \log\mathbb{E}\left[w\right] - \sum_{j=1}^{\infty}\frac{(-1)^j}{j\mathbb{E}\left[w\right]^j}(Y_K - \mathbb{E}\left[w\right])^j.$$

Taking the expectation with respect to the samples $\mathbf{x}_i$ leads to:

$$\mathbb{E}\left[\log Y_K\right] = \log \mathbb{E}\left[w\right] - \sum_{j=1}^{\infty} \frac{(-1)^j}{j\mathbb{E}\left[w\right]^j} \mathbb{E}\left[(Y_K - \mathbb{E}\left[w\right])^j\right].$$

We can relate the moments $\gamma_i := \mathbb{E}\left[(Y_K - \mathbb{E}\left[Y_K\right])^i\right]$ of the sample mean Y_K to the moments $\mu_i := \mathbb{E}\left[(w - \mathbb{E}\left[w\right])^i\right]$ of the samples w using the Theorem 1 of (Angelova, 2012):

$$\gamma_2 = \frac{\mu_2}{K}$$
$$\gamma_3 = \frac{\mu_3}{K^2}.$$

Expanding the Taylor series to order 3 leads to:

$$\mathbb{E}\left[\log Y_K\right] = \log \mathbb{E}\left[w\right] - \frac{1}{2\mu^2}\frac{\mu_2}{K} + \frac{1}{3\mu^3}\left(\frac{\mu_3}{K^2}\right) + o(\frac{1}{K}),$$

which implies

$$\mathbb{E}\left[\mathcal{L}_K(\theta)\right] = \log p(\mathbf{y}) - \frac{1}{K}\frac{\mu_2}{2\mu^2} + \mathcal{O}(\frac{1}{K}).$$

□

Again, we directly copy Nowozin (2018) to show the convergence rate of the variance of $\mathcal{L}_K(\theta)$ to 0 in $\mathcal{O}(\frac{1}{K})$.

Proof. Using the definition of the variance and the Taylor series of the logarithm, we have:

$$\begin{aligned}
\mathbb{V}\left[\log Y_K\right] &= \mathbb{E}\left[(\log Y_K - \mathbb{E}\left[\log Y_K\right])^2\right] \\
&= \mathbb{E}\left[\left(\log \mu - \sum_{i=1}^{\infty} \frac{(-1)^i}{i\mu^i}(Y_K - \mu)^i - \log \mu + \sum_{i=1}^{\infty} \frac{(-1)^i}{i\mu^i}\mathbb{E}\left[(Y_K - \mu)^i\right]\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{\infty} \frac{(-1)^i}{i\mu^i}(\mathbb{E}\left[(Y_K - \mu)^i\right] - (Y_K - \mu)^i\right)^2\right].
\end{aligned}$$

If we expand the last expression to the third order and substitute the samples moments γ_i with the central moments μ_i we eventually obtain:

$$\mathbb{V}\left[\log Y_K\right] = \frac{1}{K}\frac{\mu_2}{\mu^2} - \frac{1}{K^2}(\frac{\mu_3}{\mu^3} - \frac{5\mu_2^2}{2\mu^4}) + o(\frac{1}{K^2}).$$

□

B.3 $\mathcal{L}_K$ non-decreasing with K

Closely following Burda et al. (2016), we show the estimator is non-decreasing with K,

$$\mathbb{E}\left[\mathcal{L}_{K+1}(\theta)\right] \geq \mathbb{E}\left[\mathcal{L}_K(\theta)\right].$$

Proof. Let $I = \{i_1, ..., i_K\} \subset \{1, ..., K+1\}$ with $|I| = K$ be a uniformly distributed subset of K distinct indices from $\{1, ..., K+1\}$. We notice that $\mathbb{E}_I\left[\frac{\sum_{k=1}^{K} a_{i_k}}{K}\right] = \frac{\sum_{k=1}^{K+1} a_k}{K+1}$ for any sequence of numbers $a_1, ..., a_{K+1}$.

Using this observation and Jensen's inequality leads to

$$\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+1}}\left[\mathcal{L}_{K+1}(\theta)\right] = \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+1}}\left[\log \frac{1}{K+1}\sum_{k=1}^{K+1} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] \tag{10a}$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+1}}\left[\log \mathbb{E}_I\left[\frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_{i_k}))\right]\right] \tag{10b}$$

$$\geq \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+1}}\left[\mathbb{E}_I\left[\log \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_{i_k}))\right]\right] \tag{10c}$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K}}\left[\log \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_{k}))\right] \tag{10d}$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K}}\left[\mathcal{L}_K\right] \tag{10e}$$

□

B.4 $\mathcal{L}_K$ consistency

We show the consistency of the estimator $\mathcal{L}_K$, that is:

$$\lim_{K\to\infty} \mathcal{L}_K(\theta) = \log q_\theta(\mathbf{y}). \tag{11}$$

Proof. Using the strong law of large numbers:

$$\lim_{K\to\infty} \mathcal{L}_K(\theta) = \lim_{K\to\infty} \log \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k)) \tag{12a}$$

$$= \log \lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k)) \tag{12b}$$

$$= \log \mathbb{E}_{p(\boldsymbol{\epsilon})} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon})) \tag{12c}$$

$$= \log \mathbb{E}_{q_\theta(\mathbf{x})} p(\mathbf{y}|\mathbf{x}) \tag{12d}$$

$$= \log q_\theta(\mathbf{y}). \tag{12e}$$

In Eq. 12a, we rewrite the definition of the estimator and then, in Eq. 12b we interchange the limit and logarithm operators by continuity of the logarithm. In Eq. 12c, we use the strong law of large numbers and then, in Eq. 12d we use the LOTUS theorem to rewrite the expectation with respect to the distribution $q_\theta(\mathbf{x})$ implicitly defined by the generative model $\mathbf{G}_\theta(\cdot)$. Finally, Eq. 12e is obtained by marginalization. □

B.5 Unbiased estimator $\hat{\mathcal{L}}_K$

We want to show this estimator is unbiased,

$$\mathbb{E}_{J\sim P(J),\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+J}\sim p(\boldsymbol{\epsilon})}\left[\hat{\mathcal{L}}_K\right] = \log q_\theta(\mathbf{y}),$$

where

$$\hat{\mathcal{L}}_K = \mathcal{L}_K + \eta$$

with $\eta = \sum_{j=0}^{J} \frac{\mathcal{L}_{K+j+1}(\theta) - \mathcal{L}_{K+j}(\theta)}{P(\mathcal{J}\geq j)}$.

Proof. Following closely Luo et al. (2020), we proceed as follows. First we observe that:

$$\mathbb{E}_{J\sim P(J),\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+J}\sim p(\boldsymbol{\epsilon})}\left[\hat{\mathcal{L}}_K\right] = \mathbb{E}_{J\sim P(J),\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+J}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_K + \eta\right] \tag{13a}$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_K\right] + \mathbb{E}_{J\sim P(J),\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+J}\sim p(\boldsymbol{\epsilon})}\left[\eta\right], \tag{13b}$$

where we have:

$$\mathbb{E}_{J\sim P(J),\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+J}\sim p(\boldsymbol{\epsilon})}\left[\eta\right] = \mathbb{E}_{J\sim P(J),\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+J}\sim p(\boldsymbol{\epsilon})}\left[\sum_{j=0}^{J}\frac{\mathcal{L}_{K+j+1}(\theta)-\mathcal{L}_{K+j}(\theta)}{P(\mathcal{J}\geq j)}\right] \tag{13c}$$

$$= \mathbb{E}_{J\sim P(J)}\left[\sum_{j=0}^{J}\frac{\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+j+1}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_{K+j+1}(\theta)\right]-\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+j}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_{K+j}(\theta)\right]}{P(\mathcal{J}\geq j)}\right] \tag{13d}$$

$$= \sum_{j=0}^{\infty}\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+j+1}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_{K+j+1}(\theta)\right]-\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K+j}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_{K+j}(\theta)\right] \tag{13e}$$

$$= \lim_{j\to\infty}\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{j}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_{j}(\theta)\right]-\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_{K}(\theta)\right] \tag{13f}$$

$$= \log q_\theta(\mathbf{y}) - \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_{K}\sim p(\boldsymbol{\epsilon})}\left[\mathcal{L}_{K}(\theta)\right], \tag{13g}$$

where Eq. 13e is a property of the Russian roulette estimator (see Lemma 3 of (Chen et al., 2019)) that holds if (i) $P(\mathcal{J}\geq k)>0, \forall k>0$ and (ii) the series converge absolutely. The first condition is ensured by the choice of $P(J)$ and the second condition is also ensured thanks to the non-decreasing and consistency properties of the biased estimator. □

C Benchmark problems

Beyond doing inference on a real simulator from collider physics, we show the applicability of the methods on three benchmark simulators inspired from the literature that are described below.

C.1 Simple likelihood and complex posterior (SLCP)

Given parameters $\mathbf{x}\in\mathbb{R}^5$, the SLCP simulator (Papamakarios et al., 2019) generates $\mathbf{y}\in\mathbb{R}^8$ according to:

$$\boldsymbol{\mu} = [x_1, x_2]^\top \tag{14a}$$

$$s_1 = x_3^2 \tag{14b}$$

$$s_2 = x_4^2 \tag{14c}$$

$$\rho = \tanh(x_5) \tag{14d}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2 \end{bmatrix} \tag{14e}$$

$$\mathbf{y}_j \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad j=1,\ldots,4 \tag{14f}$$

$$\mathbf{y} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_4^\top]^\top. \tag{14g}$$

The source data $p(\mathbf{x})$ is uniform between $[-3,3]$ for each x_i.

C.2 Two-moons

Given parameters $\mathbf{x}\in\mathbb{R}^2$, the the two-moons simulator (Ardizzone et al., 2019) generates $\mathbf{y}\in\mathbb{R}^2$ according to:

$$a \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2}) \tag{15a}$$

$$r \sim \mathcal{N}(0.1, 0.01^2) \tag{15b}$$

$$\mathbf{p} = [r\cos(a) + 0.25, r\sin(a)]^\top \tag{15c}$$

$$\mathbf{y} = \mathbf{p} + [-\frac{|x_1+x_2|}{\sqrt{2}}, \frac{-x_1+x_2}{\sqrt{2}}]^\top. \tag{15d}$$

The source data $p(\mathbf{x})$ is uniform between $[-1,1]$ for each x_i.

C.3 Inverse Kinematics

Ardizzone et al. (2018) introduced a problem where $\mathbf{x} \in \mathbb{R}^4$ but that can still be easily visualized in 2-D. They model an articulated arm that can move vertically along a rail and that can rotate at three joints. Given parameters $\mathbf{x}$, the arm's end point $\mathbf{y} \in \mathbb{R}^2$ is defined as:

$$y_1 = x_1 + l_1 \sin(x_2) + l_2 \sin(x_2 + x_3) + l_3 \sin(x_2 + x_3 + x_4) \tag{16a}$$
$$y_2 = l_1 \cos(x_2) + l_2 \cos(x_2 + x_3) + l_3 \cos(x_2 + x_3 + x_4) \tag{16b}$$

with arm lengths $l_1 = l_2 = 0.5, l_3 = 1.0$.

As the forward model defined in Eq. 16 is deterministic and that we are interested in stochastic simulators, we add noise at each rotating joint. Noise is sampled from a normal distribution $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.00017$ rad $\equiv 0.01°$.

The source data $p(\mathbf{x})$ follows a gaussian $\mathcal{N}(0, \sigma_i^2)$ for each $\mathbf{x}_i$ with $\sigma_1 = 0.25$ rad $\equiv 14.33°$ and $\sigma_2 = \sigma_3 = \sigma_4 = 0.5$ rad $\equiv 28.65°$.

D Benchmark problems - hyperparameters

The surrogate models $q_\phi(\mathbf{y}|\mathbf{x})$ are modeled with coupling layers (Dinh et al., 2015, 2017) where the scaling and translation networks are modeled with MLPs with ReLu activations. In Dinh et al. (2017), the scaling function is squashed by a hyperbolic tangent function multiplied by a trainable parameter. We rather use soft clamping of scale coefficients as introduced in Ardizzone et al. (2019):

$$s_{clamp} = \frac{2\alpha}{\pi} \arctan(\frac{s}{\alpha}) \tag{17}$$

which gives $s_{clamp} \approx s$ for $s \ll |\alpha|$ and $s_{clamp} \approx \pm\alpha$ for $|s| \gg \alpha$. We performed a grid search over the surrogate model hyperparameters and found $\alpha = 1.9$ to be a good value for most architectures, as in Ardizzone et al. (2019). Therefore, we fixed α to 1.9 in all models.

The surrogate models are trained for 300 epochs over the whole dataset of pairs of source and corrupted data. Conditioning is done by concatenating the conditioning variables $\mathbf{x}$ on the inputs of the scaling and translation networks. More details are given in Table 5.

Architecture	
Network architecture	Coupling layers
Scaling network	3×50 (MLP)
Translation network	3×50 (MLP)
N°flows	4
Batch size	128
Optimizer	Adam
Weight decay	5×10^{-5}
Learning rate	10^{-4}

Table 5: Hyperparameters used to train and model $q_\phi(\mathbf{y}|\mathbf{x})$

The source data distributions $q_\theta(\mathbf{x})$ are modeled with UMNN-MAFs (Wehenkel and Louppe, 2019). The forward evaluation of these models defines a bijective and differentiable mapping from a distribution to another one which allows to compute the jacobian of the transformation in $\mathcal{O}(d)$ where d is the dimension of the distributions. However, inverting the model requires to solve a root finding algorithm which is not trivially differentiable. For $\mathcal{L}_K$ and $\hat{\mathcal{L}}_K$, the forward model defines a differentiable mapping from noise $\mathbf{z}$ to $\mathbf{x}$. This design allows to sample new data points in a differentiable way and to evaluate their densities.

For $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}_K^{\text{IW}}$, the forward model defines a differentiable mapping from $\mathbf{x}$ to $\mathbf{z}$ which allows to evaluate in a differentiable way the density of any data point $\mathbf{x}$, as required by the two losses. $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}_K^{\text{IW}}$ also require

to introduce a recognition network $q_\psi(\mathbf{x}|\mathbf{y})$ which should allow to differentially sample new data points and evaluate their densities. Therefore, the same architecture as $q_\theta(\mathbf{x})$ is used. The core architecture of all models is the same and detailed in Table 6.

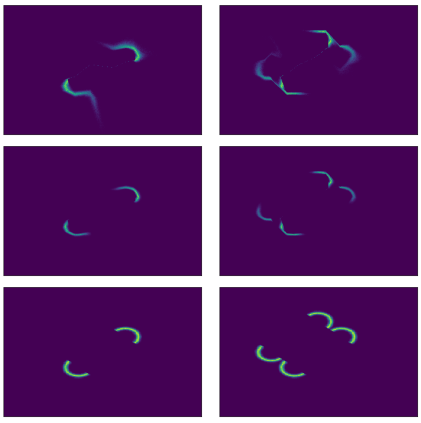
$\mathcal{L}_K$ and $\hat{\mathcal{L}}_K$ are trained over 100 epochs over the whole observed dataset. For $\mathcal{L}_K$, 10% of the data were held out to stop training if the loss did not improve for 10 epochs. The $\hat{\mathcal{L}}_K$ loss was extremely noisy and therefore, no early stopping was performed. Nonetheless, other strategies could have been used such as stopping training when the discrepancy between the observed distribution $p(\mathbf{y})$ and the regenerated one $\int q_\phi(\mathbf{y}|\mathbf{x})q_\theta(\mathbf{x})$ did not improve. $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}_K^{\text{IW}}$ need more epochs to converge, likely due to the training of two networks simultaneously. When using these losses, training was done over 300 epochs over the whole observed dataset with 10% of the data held out to stop training if the losses did not improve for 10 epochs.

Architecture	
Network architecture	UMNN-MAF
N°integ. steps	20
Embedding network	3×75 (MADE)
Integrand network	3×75 (MLP)
N°flows	6
Embedding Size	10
Batch size	128
Optimizer	Adam
Weight decay	0.0
Learning rate	10^{-4}

Table 6: Hyperparameters used to train and model $q_\theta(\mathbf{x})$ and $q_\psi(\mathbf{x}|\mathbf{y})$.

E N=1 Empirical Bayes

Throughout the paper, the prior has been learned from the data given a large number of observations as it is often the case in the Empirical Bayes literature. Interestingly, Figure 6 shows that even with a single (or two) observation(s), the method is able to learn the set of source data that may have generated the observation(s). When the number of observations is low, we observed that UMMN-MAFs tend to degenerate and concentrate all their masses to single points. Therefore, for this experiment, we used coupling layers that act as regularizers and do not collapse. We aim at studying the regularization introduced by bijective neural networks and how this may affect the learning of source data in the Neural Empirical Bayes framework in future work.



(a) N=1. (b) N=2.

Figure 6: Empirical Bayes with only $N = 1$ or $N = 2$ observations. (**Top row**) Learned (prior) distributions over source data. (**Middle row**) Learned distributions weighted by the likelihood approximated with the surrogate model. (**Bottom row**) Set of source data that may have generated the observation(s). Even with few observations, the method learns good posterior distributions.

In this experiment, we used the $\mathcal{L}_K$ loss with $K = 1024$. The distribution $q_\theta(\mathbf{x})$ was modeled by 3 coupling layers where the scaling and translation networks are MLPs of 3 layers of 16 hidden units with ReLu activation.

F Empirical Bayes with simple models

While the need to evaluate the density of new data points under the source model with $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^{\text{IW}}$ heavily restricts the model architectures that can be used to model $q_\theta(\mathbf{x})$, $\mathcal{L}_{\text{K}}$ and $\hat{\mathcal{L}}_K$ allow to use any generative model mapping some noise $\mathbf{z} \in \mathbb{R}^n$ to $\mathbf{x} \in \mathbb{R}^d$.

Normalizing flows have been consistently used in this paper to model $q_\theta(\mathbf{x})$. Wile these models may in themselves act as a good inductive bias for continuous and smooth source distributions, we show here that simple MLPs can also learn good source distributions. This experiment is particularly useful as it shows that $\mathcal{L}^{\text{K}}$ and $\hat{\mathcal{L}}_K$ allow to use a broader class of model architectures than $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^{\text{IW}}$. This opens interesting research directions where useful inductive bias can be embedded in the source model. For example CNNs and RNNS can be used for image and time series analysis.

In this experiment, we model $q_\theta(\mathbf{x})$ with a 3−layer MLPs with 100 units per layer and ReLU activations. We optimize θ with the same hyperparameters described in Appendix D. For a fixed GPU memory, the usage of simpler and lighter models allows to use higher values of K. In this experiment, we use $K = 2^{10}$ and $K = 2^{12}$.

Simulator	$\mathbf{y}$-space		$\mathbf{x}$-space	
	$\mathcal{L}_{1024}$	$\mathcal{L}_{4096}$	$\mathcal{L}_{1024}$	$\mathcal{L}_{4096}$
SLCP	$0.55_{\pm 0.01}$	$0.52_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.92_{\pm 0.01}$
Two-moons	$0.53_{\pm 0.02}$	$0.52_{\pm 0.01}$	$0.68_{\pm 0.04}$	$0.62_{\pm 0.05}$
IK	$0.66_{\pm 0.03}$	$0.58_{\pm 0.02}$	$0.92_{\pm 0.01}$	$0.90_{\pm 0.02}$

Table 7: Source estimation for the benchmark problems. ROC AUC between $q_\theta(\mathbf{x})$ and $p(\mathbf{x})$ ($\mathbf{x}$-space), and between the observed distribution $p(\mathbf{y})$ and the regenerated distribution $\int p(\mathbf{y}|\mathbf{x})q_\theta(\mathbf{x})d\mathbf{x}$ ($\mathbf{y}$-space).

Table 7 reports the discrepancy between the corrupted data from the identified source distributions and the ground truth distribution of noise-corrupted observations ($\mathbf{y}$-space). It shows that simple architectures allow to learn a source distribution that can closely reproduce the observed distribution. The ROC AUC between the source distribution $q_\theta(\mathbf{x})$ and the ground truth distribution $p(\mathbf{x})$ shows that the source distribution learned on the two-moons problem is close to the ground truth. For the other problems, useful inductive bias should be introduced to constrain the solution space.

G Symmetric UMNN-MAF

UMNN-MAF are autoregressive architectures such that:

$$\mathbf{x} = \mathbf{G}(\mathbf{z}) = [g^1(z_1), ..., g^d(\mathbf{z}_{1:d})], \tag{18}$$

where each $g^i(\cdot)$ is a bijective scalar function such that:

$$g^i(\mathbf{z}_{1:i}) = \int_0^{z_i} f^i(t, \mathbf{h}^i(\mathbf{z}_{1:i-1}))dt + \beta^i(\mathbf{h}^i(\mathbf{z}_{1:i-1})), \tag{19}$$

where $\mathbf{h}^i(\cdot) : \mathbb{R}^{i-1} \to \mathbb{R}^q$ is a q-dimensional neural embedding of the variables $\mathbf{z}_{1:i-1}$, $f^i(\cdot) \in \mathbb{R}^+$ and $\beta^i(\cdot)$ is a scalar function.

In order to make the distribution $q_\theta(\mathbf{x})$ one-to-one symmetric, i.e. $q_\theta([x_1, ..., x_d]) = q_\theta([\pm x_1, ..., \pm x_d])$, it is sufficient that (i) the distribution $p(\mathbf{z})$ is one-to-one symmetric, (ii) $\beta^i(\cdot)$ is set to 0 and, (iii) the integrand function is such that $f^i(t, \mathbf{h}^i(x_1, ..., x_{i-1})) = f^i(\pm t, \mathbf{h}^i(\pm x_1, ..., \pm x_{i-1}))$. The condition (iii) is enforced by taking the absolute value of the input variables in the first layer of the integrand and embedding networks.

Then, $q_\theta([x_1, ..., x_d]) = q_\theta([\pm x_1, ..., \pm x_d]$.

Proof. First note that if conditions (ii) and (iii) are met:

$$x^i = g^i(\pm z_1, ..., \pm z_{i-1}, z_i) \Leftrightarrow g^i(\pm z_1, ..., \pm z_{i-1}, -z_i) = -x^i \tag{20}$$

and

$$|\det J_{g^i(\pm z_1,...,\pm z_{i-1},z_i)}| = |\det J_{g^i(\pm z_1,...,\pm z_{i-1},-z_i)}| = f^i(\pm z_i, \mathbf{h}^i(\pm z_1, ..., \pm z_{i-1})), \tag{21}$$

where $J_{g^i(\pm z_1,...,\pm z_i)}$ is the Jacobian of $g^i(\cdot)$ with respect to z_i.

It follows that:

$$\begin{aligned}
q_\theta([x_1, ..., x_d]) &= p(z_1, ..., z_d)|\det J_{\mathbf{G}(\mathbf{z})}|^{-1} && \text{(22a)}\\
&= p(z_1, ..., z_d)\prod_{i=1}^{d} f^i(z_i, \mathbf{h}^i(z_1, ..., z_{i-1}))^{-1} && \text{(22b)}\\
&= p(\pm z_1, ..., \pm z_d)\prod_{i=1}^{d} f^i(\pm z_i, \mathbf{h}^i(\pm z_1, ..., \pm z_{i-1}))^{-1} && \text{(22c)}\\
&= p(\pm z_1, ..., \pm z_d)|\det J_{\mathbf{G}(\pm z_1,...,\pm z_d)}|^{-1} && \text{(22d)}\\
&= q_\theta([\pm x_1, ..., \pm x_d]\,. && \text{(22e)}
\end{aligned}$$

Eq. 22a is a direct application of the change of variable theorem while Eq. 22b is obtained by definition. Conditions (i) and (iii) allow us to write Eq. 22b as Eq. 22c. The equalities in Eq. 21 yields Eq. 22d. and finally, the last equation is obtained from Eq. 22d and Eq. 20. □

H Simulation-Based Calibration

In order to perform simulation-based calibration, we repeatedly i) sample $\mathbf{x}^*$ from $p(\mathbf{x})$, ii) generate $\mathbf{y}^*$ by running the simulator conditioned on $\mathbf{x}^*$, iii) perform rejection sampling in order to empirically approximate $p(\mathbf{x}|\mathbf{y}^*)$, and iv) store for each dimension i in which quantile of $p(x_i|y_i^*)$, x_i^* fall.

For each dimension, it is expected that $x\%$ of the parameters belong to the $x\%$ quantile of $p(x_i|y_i)$ and this can be assessed qualitatively by plotting the fraction of events per quantile as in figures 3 and 5c. To perform a quantitative assessment, one can for example, compute the maximum absolute difference between the fraction of events that fall within a quantile and the value of that quantile, or in order words, report the Kolmogorov–Smirnov (KS) test between the empirical cumulative distribution function (blue lines on Figure 3 or Figure 5c) and the expected cumulative distribution function (black line on Figure 3 or Figure 5c). We report those quantities in Table 8 for the different estimators.

The strength of this approach is that it allows to get insights about the learned posterior distribution without access to a ground truth. For example, if the model tends to assign more than $x\%$ of the parameters to the $x\%$ quantile, the model is overconfident. On the other hand, if it tends to assign less than $x\%$ of the parameters to the $x\%$ quantile, the model is underconfident.

In terms of weaknesses, the described approach only independently evaluate the 1D marginal distributions rather than the full-space distribution. In higher dimension, quantiles can be extended to contours but these might not be easily computable. Moreover, the approach is a necessary, but not sufficient, condition for well-calibrated posterior distribution. For example, by design $p(\mathbf{x})$ would pass the calibration test.

Simulator	Estimator	KS
SLCP	$\mathcal{L}^{\text{ELBO}}$	$0.37_{\pm 0.01}$
	$\mathcal{L}^{\text{IW}}_{128}$	$0.15_{\pm 0.02}$
	$\mathcal{L}_{1024}$	$\mathbf{0.14_{\pm 0.01}}$
Two-moons	$\mathcal{L}^{\text{ELBO}}$	$0.40_{\pm 0.01}$
	$\mathcal{L}^{\text{IW}}_{128}$	$0.45_{\pm 0.01}$
	$\mathcal{L}_{1024}$	$\mathbf{0.10_{\pm 0.01}}$
IK	$\mathcal{L}^{\text{ELBO}}$	$0.65_{\pm 0.04}$
	$\mathcal{L}^{\text{IW}}_{128}$	$0.47_{\pm 0.05}$
	$\mathcal{L}_{1024}$	$\mathbf{0.09_{\pm 0.02}}$

Table 8: Calibration test from 1000 posterior estimates obtained with rejection sampling for $\mathcal{L}_{1024}$, importance sampling for $\mathcal{L}^{\text{IW}}_{128}$ and directly from the recognition network $q_\psi(\mathbf{x}|\mathbf{y})$ for $\mathcal{L}^{\text{ELBO}}$. *As opposed to* $\mathcal{L}_{1024}$*, the posterior distributions for* $\mathcal{L}^{IW}_{128}$ *and* $\mathcal{L}^{ELBO}$ *are not consistently correctly calibrated.*

I Collider Physics Simulation

The simulated physics dataset, made publically available by Andreassen et al. (2019b), targets conditions similar to those produced by the proton-proton collisions at $\sqrt{s} = 14$ TeV at the Large Hadron Collider (Evans and Bryant, 2008). For surrogate training, source distributions of jets from collisions producing Z bosons recoiling off of jets are modeled with the the Monte Carlo simulator Pythia 8.243 (Sjöstrand et al., 2015) with Tune 26 (ATL, 2014). For learning the source distribution with NEB, an alternative simulation of the source distribution of jets from collisions producing Z bosons recoiling off of jets is performed with Herwig 7.1.5 (Bähr et al., 2008; Bahr et al., 1999) with default tune. The Delphes simulator (de Favereau et al., 2014) is used to model the impact of detector effects on particle measurements using a parameterized detector smearing that models the smearing effects in the ATLAS (ATLAS Collaboration, 2008) or CMS (CMS Collaboration, 2008) experiments.