# Neural Empirical Bayes: Source Distribution Estimation and its Applications to Simulation-Based Inference

**Maxime Vandegar**[1]
SLAC National Accelerator Laboratory

**Michael Kagan**[2]
SLAC National Accelerator Laboratory

**Antoine Wehenkel**[3]
University of Liège

**Gilles Louppe**[4]
University of Liège

## Abstract

We revisit empirical Bayes in the absence of a tractable likelihood function, as is typical in scientific domains relying on computer simulations. We investigate how the empirical Bayesian can make use of neural density estimators first to use all noise-corrupted observations to estimate a prior or source distribution over uncorrupted samples, and then to perform single-observation posterior inference using the fitted source distribution. We propose an approach based on the direct maximization of the log-marginal likelihood of the observations, examining both biased and de-biased estimators, and comparing to variational approaches. We find that, up to symmetries, a neural empirical Bayes approach recovers ground truth source distributions. With the learned source distribution in hand, we show the applicability to likelihood-free inference and examine the quality of the resulting posterior estimates. Finally, we demonstrate the applicability of Neural Empirical Bayes on an inverse problem from collider physics.

## 1 Introduction

The estimation of a *source* distribution over latent random variables $\mathbf{x}$ which give rise to a set of observations $\mathbf{y}$, after undergoing a potentially non-linear corruption process, is an inverse problem frequently of interest to the scientific and engineering communities. The source distribution, $p(\mathbf{x})$, may represent the distribution of plausible measurements, or intermediate random variables in a hierarchical model, prior to corruption by a measurement or detection apparatus. The source distribution is of scientific interest as it allows comparison with theoretical predictions and for posterior inference for subsequent observations. Notably, in many scientific domains, the relationship between the source and observed distributions is encoded in a simulator. The simulator provides an approximation of the corruption process and generates samples from the likelihood $p(\mathbf{y}|\mathbf{x})$. However, as is typical with computer simulations, the likelihood function is implicit and rarely known in a tractable closed form.

Formally, we state the problem of likelihood-free source estimation as follows. Given a first dataset $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$ of $N$ noise-corrupted observations $\mathbf{y}_i$ and a second dataset $\{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^M$ of matching pairs of source data and observations, with $\mathbf{x}_j \sim \pi(\mathbf{x})$ drawn from an arbitrary proposal distribution $\pi(\mathbf{x})$ and $\mathbf{y}_j \sim p(\mathbf{y}|\mathbf{x}_j)$, our aim is to learn the source distribution $p(\mathbf{x})$, not necessarily equal to $\pi(\mathbf{x})$, that has generated the observations $\mathbf{Y}$. For the class of problems we consider, we may assume that the dataset of $(\mathbf{x}_j, \mathbf{y}_j)$ pairs is generated beforehand using a simulator $s(\cdot;\mathbf{x}) : \mathcal{E} \to \mathcal{Y}$ of the stochastic corruption process.

The source distribution estimation problem is closely related to likelihood-free inference (LFI, Cranmer et al., 2020), though there are notable differences in problem statements. First, in Bayesian LFI, the objective is the computation of a posterior given a known prior and an implicit likelihood function. In our problem statement, the primary objective is rather to identify an unknown prior or source distribution that, once identified, then enables likelihood-free inference. Second, we only assume access to a pre-generated dataset of pairs of simulated source data and observations. In many settings, simulators are highly complex, with long run times to generate data. As such, sequential methods based on active calls to the simulator, as often found in the LFI literature, would be impractical.

In this work, we follow an empirical Bayes (EB, Robbins, 1956; Dempster et al., 1977) approach to address

1: maxime.vandegar@slac.stanford.edu
2: makagan@slac.stanford.edu
3: antoine.wehenkel@uliege.be
4: g.louppe@uliege.be

this challenge, using modern neural density estimators to approximate both the intractable likelihood and the unknown source distribution. Our method, which we call Neural Empirical Bayes (NEB), proceeds in two steps. First, using simulated pairs $\{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^M$, we use neural density estimation to learn an approximate likelihood. Second, by modeling the source distribution with a parameterized generative model, the log-marginal likelihood of the observations is approximated with Monte Carlo integration, and the parameters of the source distribution learned through gradient-based optimization. While our estimator of the log-marginal likelihood is biased, it is consistent and the use of deep generative models allows for fast and parallelizable Monte Carlo integration to mitigate its bias. Nonetheless, we also examine de-biased and variational estimators for comparison. Finally, once a source distribution and likelihood function are learned, we demonstrate that posterior inference for new observations may be performed with suitable sampling-based methods.

We first review EB and describe our NEB approach in Section 2, followed by an examination of log-marginal likelihood estimators in Section 3. Related work is discussed in Section 4. In Section 5, we present benchmark problems that explore the efficacy of NEB and provide comparison baselines, as well as a demonstration on a real-world application to collider physics. Further discussion and a summary are in Section 6. In addition, we provide a summary of the notations in Appendix A.

## 2 Empirical Bayes

Methods for EB (Robbins, 1956; Dempster et al., 1977) are usually divided into two estimation strategies (Efron, 2014): either modeling on the $\mathbf{x}$-space, called $g$-modeling; or on the $\mathbf{y}$-space, called $f$-modeling.

Here, we revisit $g$-modeling to learn a source distribution that regenerates the observations $\mathbf{Y}$. Specifically, we parameterize the source distribution as $q_\theta(\mathbf{x})$ which, when passed through the likelihood $p(\mathbf{y}|\mathbf{x})$, results in a distribution $q_\theta(\mathbf{y})$ over noisy observations. The log-marginal likelihood of the observations $\mathbf{Y}$ is expressed as

$$\begin{aligned}\log q_\theta(\mathbf{Y}) &= \sum_{i=1}^N \log q_\theta(\mathbf{y}_i) \\ &= \sum_{i=1}^N \log \int p(\mathbf{y}_i|\mathbf{x}) q_\theta(\mathbf{x}) d\mathbf{x}, \end{aligned} \tag{1}$$

and its direct maximization with respect to the parameters $\theta$ leads to a solution for the source distribution.

The maximization of the log-marginal likelihood is equivalent to the minimization of the Kullback–Leibler divergence $\mathrm{KL}(p(\mathbf{y})||q_\theta(\mathbf{y})) = \mathbb{E}_{p(\mathbf{y})}[-\log q_\theta(\mathbf{y})] + \not{C} \approx -\frac{1}{N}\sum_{i=1}^N \log q_\theta(\mathbf{y}_i)$. Therefore, as $\mathbf{Y}$ increases, an optimal solution will correspond to a source distribution that exactly reproduces the observation distribution when passed through the corruption process. We note however that the maximization of Eq. 1 is an ill-posed problem: distinct source distributions may result in the same distribution over observations when folded through the corruption process.

In the likelihood-free setting, the likelihood function $p(\mathbf{y}|\mathbf{x})$ is only implicitly defined by the simulator $s(\cdot;\mathbf{x})$, which prevents the direct estimation of Eq. 1. However, a dataset $\{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^M$ can be generated beforehand by drawing uncorrupted samples $\mathbf{x}_j$ from a proposal distribution $\pi(\mathbf{x})$ and running the simulator to generate corresponding noise-corrupted observations $\mathbf{y}_j$. Similarly to Diggle and Gratton (1984) and D'Agostini (1995) who built likelihood function estimators with kernels or histograms, we use the generated dataset to train a surrogate $q_\phi(\mathbf{y}|\mathbf{x})$ of the likelihood function, but we make use of modern neural density estimators such as normalizing flows (Tabak et al., 2010; Rezende and Mohamed, 2015). After the upfront simulation cost of generating the training data, no additional call to the simulator is needed.

We optimize the parameters $\phi$ by maximizing the total log-likelihood $\sum_{m=1}^M \log q_\phi(\mathbf{y}_m|\mathbf{x}_m)$ with mini-batch stochastic gradient ascent. Again, for large $M$, this is equivalent to minimizing $\mathbb{E}_{\pi(\mathbf{x})}\mathrm{KL}(p(\mathbf{y}|\mathbf{x})||q_\phi(\mathbf{y}|\mathbf{x}))$ and given enough capacity the surrogate likelihood is guaranteed to be a good approximation of $p(\mathbf{y}|\mathbf{x})$ in the support of the proposal distribution $\pi(\mathbf{x})$. As a consequence, the support of $\pi(\mathbf{x})$ should be chosen to cover the full range of plausible source data values.

## 3 Log-marginal likelihood estimation

In Section 3.1 (resp. 3.2) we build a biased (resp. unbiased) estimator of the log-marginal likelihood $\log q_\theta(\mathbf{y})$ that does not require the density of the source distribution $q_\theta(\mathbf{x})$, but can provide samples differentiable with respect to $\theta$. We use a generative model $\mathbf{G}_\theta(\cdot) : \mathcal{E} \to \mathcal{X}$ that defines a differentiable mapping from a base distribution $p(\boldsymbol{\epsilon})$ to $q_\theta(\mathbf{x})$. Then, in Section 3.3, we show how to use variational estimators of $\log q_\theta(\mathbf{y})$ for NEB.

### 3.1 Biased estimator

Given a likelihood function $p(\mathbf{y}|\mathbf{x})$ or its surrogate $q_\phi(\mathbf{y}|\mathbf{x})$ we define an estimator $\mathcal{L}_K(\theta)$ of the log-marginal likelihood. This estimator can be plugged in Eq. 1 to optimize the source distribution parame-

ters $\theta$ by stochastic minibatch gradient ascent. Based on Monte Carlo integration, the estimator is defined as:

$$\begin{aligned}\log q_\theta(\mathbf{y}) &= \log \mathbb{E}_{q_\theta(\mathbf{x})}\left[p(\mathbf{y}|\mathbf{x})\right] \\ &= \log \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}))\right] \\ &\approx \log \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k)) \\ &= \text{logSumExp}\left[\log p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] - \text{C} \\ &=: \mathcal{L}_K(\theta), \end{aligned} \tag{2}$$

where $\boldsymbol{\epsilon}_k \sim p(\boldsymbol{\epsilon})$, $C$ is a constant independent of $\theta$, and the log-sum-exp trick is used for numerical stability. While a large number $K$ of samples may be needed for good Monte Carlo approximation, this difficulty is alleviated by the ease of generating large samples of source data with the neural sampler $\mathbf{G}_\theta$.

We study and prove properties of the estimator $\mathcal{L}_K(\theta)$ in Appendix B. Using the Jensen's inequality, we first show that $\mathcal{L}_K(\theta)$ is biased. We demonstrate however that both its bias and its variance decrease at a rate of $\mathcal{O}(\frac{1}{K})$. Then, similarly to Burda et al. (2016), we show that $\mathcal{L}_K(\theta)$ is monotonically non-decreasing in expectation with respect to $K$, i.e.

$$\mathbb{E}\left[\mathcal{L}_{K+1}(\theta)\right] \geq \mathbb{E}\left[\mathcal{L}_K(\theta)\right]. \tag{3}$$

As $K \to \infty$, we finally show that the estimator is however consistent:

$$\lim_{K\to\infty} \mathcal{L}_K(\theta) = \log q_\theta(\mathbf{y}). \tag{4}$$

### 3.2 Unbiased estimator

Using the Russian roulette estimator (Kahn, 1955), we de-bias the log-marginal likelihood estimator $\mathcal{L}_K(\theta)$ as

$$\hat{\mathcal{L}}_K(\theta) := \mathcal{L}_K(\theta) + \eta(\theta), \tag{5}$$

where $\eta(\theta)$ is a random variable defined as

$$\eta(\theta) = \sum_{j=0}^{J} \frac{\mathcal{L}_{K+j+1}(\theta) - \mathcal{L}_{K+j}(\theta)}{P(\mathcal{J} \geq j)}, \tag{6}$$

with $J \sim P(J)$. Similarly to Luo et al. (2020) in their study of Importance Weighted Auto-Encoders (IWAEs, Burda et al., 2016), we prove in Appendix B.5 that $\hat{\mathcal{L}}_K$ is an unbiased estimator as long as $P(J)$ is a discrete distribution such that $P(\mathcal{J} \geq j) > 0, \forall j > 0$. Ideally, the distribution $P(J)$ should be chosen such that it adds only a small computational overhead, while providing a finite-variance estimator. In our experiments, we reduce the computational overhead by re-using the same Monte Carlo terms used for $\mathcal{L}_{K+j}$ to compute $\mathcal{L}_{K+j+1}$.

### 3.3 Variational empirical Bayes

For EB in high-dimension, Wang et al. (2019) proposed to introduce a variational posterior distribution $q_\psi(\mathbf{x}|\mathbf{y})$ and to jointly learn the parameters $\theta$ of the source distribution and the parameters $\psi$ of the posterior by maximizing the evidence lower bound (ELBO):

$$\begin{aligned}\log q_\theta(\mathbf{y}) &\geq \log q_\theta(\mathbf{y}) - \text{KL}(q_\psi(\mathbf{x}|\mathbf{y})||p(\mathbf{x}|\mathbf{y})) \\ &= \mathbb{E}_{q_\psi(\mathbf{x}|\mathbf{y})}\left[\log q_\phi(\mathbf{y}|\mathbf{x})\right] \\ &\qquad - \text{KL}(q_\psi(\mathbf{x}|\mathbf{y})||q_\theta(\mathbf{x})) \\ &=: \mathcal{L}^{\text{ELBO}}. \end{aligned} \tag{7}$$

When $\mathcal{L}^{\text{ELBO}}$ is optimized with stochastic gradient descent, an unbiased estimator can be obtained with Monte Carlo integration – usually only one Monte Carlo sample is used which yields a tractable objective. While being tractable, the ELBO is a lower bound (and biased estimator) of the log-marginal likelihood. A common approach (Rezende and Mohamed, 2015) to tighten the bound is to model $q_\psi(\mathbf{x}|\mathbf{y})$ from a large distribution family so that it can closely match the posterior distribution, i.e. efficiently minimize $\text{KL}(q_\psi(\mathbf{x}|\mathbf{y})||p(\mathbf{x}|\mathbf{y}))$. Close to our work, IWAEs trade off computational complexity to obtain a tighter log-likelihood lower bound derived from importance sampling. Specifically, IWAEs are trained to maximize

$$\mathcal{L}^{\text{IW}}_K(\theta, \psi) = \log \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{x}_k) w(\mathbf{x}_k), \tag{8}$$

where $w(\mathbf{x}_k) = \frac{q_\theta(\mathbf{x}_k)}{q_\psi(\mathbf{x}_k|\mathbf{y})}$ and $\mathbf{x}_k \sim q_\psi(\mathbf{x}|\mathbf{y})$. IWAEs are a generalization of the ELBO based on importance weighting (setting $K = 1$ retrieves the ELBO objective). Nowozin (2018) showed that the bias and variance of this estimator vanish for $K \to \infty$ at the same rate $\mathcal{O}(\frac{1}{K})$ as $\mathcal{L}_K$.

By design, $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^{\text{IW}}$ require the evaluation of the density of new data points under the source model, whereas $\mathcal{L}_\text{K}$ and $\hat{\mathcal{L}}_K$ only require the efficient sampling from the source model. Which method to use should therefore depend on the downstream usage of the source distribution. While the evaluation of densities required by $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^{\text{IW}}$ limits the range of models that can be used and makes the introduction of inductive bias more difficult, $\mathcal{L}_\text{K}$ and $\hat{\mathcal{L}}_K$ can be used with any generative model.

## 4 Related work

**Empirical Bayes** In the most common forms of $g$-modeling, the likelihood function and the prior distribution are chosen such that the marginal likelihood

can be computed and maximized iteratively or analytically. More recent approaches model the prior distribution analytically but assume both the $\mathbf{x}-$space and $\mathbf{y}-$space are finite and discrete (Narasimhan and Efron, 2016; Efron, 2016). Then, given a known likelihood function encoded in tensor form, the distribution parameters are optimized by maximum marginal likelihood estimation. Similarly to this latter approach, we do not require a likelihood function in closed-form, but we build a continuous surrogate that allows its direct evaluation rather than discretizing it.

While Wang et al. (2019) only theoretically proposed using Eq. 7 in EB, we show experimentally in the next section the applicability of this method. Concurrent work (Dockhorn et al., 2020) also used this approach to solve a density deconvolution task on Gaussian noise processes. Our work differs as we show the applicability of these methods on much more complicated black-box simulators, including a real inverse problem from collider physics. Black-box simulators imply that a neural network surrogate replaces the likelihood function, and thus, learning $\theta$ and $\psi$ requires to backpropagate through the surrogate.

Finally, in the context of likelihood-free inference, Louppe et al. (2019) use adversarial training for learning a prior distribution such that, when corrupted by a non-differentiable black-box model, reproduces the empirical distribution of the observations. Again, this can be seen as $g$-modeling EB where a prior distribution is optimized based on observations.

**Unfolding** Approximating a source distribution $p(\mathbf{x})$ given corrupted observations is often referred to as unfolding in the particle physics literature (for reviews see Cowan, 2002; Blobel, 2011; Adye, 2011). A common approach (Lucy, 1974; D'Agostini, 1995) is to discretize the problem and replace the integral in Eq. 1 with a sum, resulting in a discrete linear inverse problem. The surrogate model $q_\phi(\mathbf{y}|\mathbf{x})$ of the likelihood function is encoded in tensor form while $q_\theta(\mathbf{x})$ is modeled with a histogram. These approaches are typically limited to low dimensions. In order to scale to higher dimensions, preliminary work by Cranmer (2018) explored the idea of modelling $q_\phi(\mathbf{y}|\mathbf{x})$ and $q_\theta(\mathbf{x})$ with normalizing flows to approximate the integral in Eq. 1 with Monte Carlo integration. Aiming to the same objective, Andreassen et al. (2019b) replaced the sum in discrete space with a full-space integral using the likelihood ratio. As opposed to our work, this method is based on re-weighting and does not allow differential sampling of new data points.

**Likelihood-free inference** The use of a surrogate model of the likelihood function that enables inference as if the likelihood was known is not new. Since Diggle and Gratton (1984), kernels and histograms have been vastly used for 1D density estimation. More recently, several Bayesian likelihood-free inference algorithms (Papamakarios et al., 2019; Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019; Hermans et al., 2020; Durkan et al., 2020) have been developed to carry out inference when the likelihood function is implicit and intractable. These methods all operate by learning parts of the Bayes' rule, such as the likelihood function, the likelihood-to-evidence ratio, or the posterior itself, and all require the explicit specification of a prior distribution. By contrast, the primary objective of our work is to learn a prior distribution from a set of noise-corrupted observations which, once it is identified, then enables any of those Bayesian LFI algorithms for posterior inference. We refer the reader to Cranmer et al. (2020) for a broader review of likelihood-free inference.

| Simulator | K | $\mathcal{L}_K$ | $\hat{\mathcal{L}}_K$ |
|---|---|---|---|
| SLCP | 10 | $0.82_{\pm 0.01}$ | $0.65_{\pm 0.04}$ |
| | 128 | $0.57_{\pm 0.01}$ | $0.59_{\pm 0.02}$ |
| | 256 | $0.55_{\pm 0.02}$ | $0.54_{\pm 0.00}$ |
| | 1024 | $0.53_{\pm 0.01}$ | $0.52_{\pm 0.01}$ |
| Two-moons | 10 | $0.69_{\pm 0.02}$ | $0.56_{\pm 0.02}$ |
| | 128 | $0.53_{\pm 0.01}$ | $0.57_{\pm 0.06}$ |
| | 256 | $0.52_{\pm 0.01}$ | $0.52_{\pm 0.02}$ |
| | 1024 | $0.52_{\pm 0.01}$ | $0.53_{\pm 0.01}$ |
| IK | 10 | $0.80_{\pm 0.13}$ | $0.67_{\pm 0.08}$ |
| | 128 | $0.65_{\pm 0.04}$ | $0.67_{\pm 0.12}$ |
| | 256 | $0.66_{\pm 0.02}$ | $0.71_{\pm 0.09}$ |
| | 1024 | $0.66_{\pm 0.03}$ | $0.62_{\pm 0.03}$ |

Table 1: ROC AUC between the observed distribution $p(\mathbf{y})$ and the regenerated distribution $\int p(\mathbf{y}|\mathbf{x})q_\theta(\mathbf{x})d\mathbf{x}$. The closer to 0.5, the better the estimation in $\mathbf{y}$-space. *NEB successfully identifies source distributions that result in distributions over noise-corrupted observations that are almost indistinguishable from the ground truth. When $K$ is low, de-biasing leads to substantial improvements.*

## 5 Experiments

We present three studies of NEB. In Section 5.1, we analyze the intrinsic quality of the recovered source distribution for the estimators discussed in Section 3. In Section 5.2, we explore posterior inference with the learned source distribution. Finally, in Section 5.3 we show the applicability of NEB on an inverse problem from collider physics. All experiments are repeated 5 times with $q_\phi(\mathbf{y}|\mathbf{x})$ and $q_\theta(\mathbf{x})$ relearned in each experiment. Means and standard deviations are reported.

| Simulator | $\mathbf{x}$-space | | | $\mathbf{y}$-space | | | $\mathbf{x}$-space (symmetric prior) | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{L}^{\text{ELBO}}$ | $\mathcal{L}^{\text{IW}}_{128}$ | $\mathcal{L}_{1024}$ | $\mathcal{L}^{\text{ELBO}}$ | $\mathcal{L}^{\text{IW}}_{128}$ | $\mathcal{L}_{1024}$ | $\mathcal{L}^{\text{ELBO}}$ | $\mathcal{L}^{\text{IW}}_{128}$ | $\mathcal{L}_{1024}$ |
| SLCP | $1.00_{\pm 0.00}$ | $0.82_{\pm 0.09}$ | $0.75_{\pm 0.03}$ | $0.92_{\pm 0.04}$ | $0.50_{\pm 0.00}$ | $0.53_{\pm 0.01}$ | $0.99_{\pm 0.01}$ | $0.59_{\pm 0.05}$ | $0.81_{\pm 0.02}$ |
| Two-Moons | $0.75_{\pm 0.00}$ | $0.75_{\pm 0.00}$ | $0.55_{\pm 0.02}$ | $0.50_{\pm 0.01}$ | $0.50_{\pm 0.00}$ | $0.52_{\pm 0.01}$ | $0.51_{\pm 0.01}$ | $0.50_{\pm 0.01}$ | $0.51_{\pm 0.02}$ |
| IK | $1.00_{\pm 0.00}$ | $0.95_{\pm 0.05}$ | $0.74_{\pm 0.03}$ | $0.51_{\pm 0.01}$ | $0.50_{\pm 0.01}$ | $0.62_{\pm 0.03}$ | $0.97_{\pm 0.01}$ | $0.72_{\pm 0.02}$ | $0.66_{\pm 0.04}$ |

Table 2: 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).

### 5.1 Source estimation

We evaluate NEB on three benchmark problems: (a) a toy model with a simple likelihood but complex posterior (SLCP) introduced by Papamakarios et al. (2019), (b) the two-moons model of Greenberg et al. (2019), and (c) an inverse kinematics problem (IK) proposed by Ardizzone et al. (2018). See Appendix C for complementary experimental details. We use datasets of $M = 15000$ samples to train surrogate models $q_\phi(\mathbf{y}|\mathbf{x})$ for each simulator. All density models are parameterized with normalizing flows made of four coupling layers (Dinh et al., 2014, 2017). Further architecture and optimization details can be found in Appendix D. The source distributions $q_\theta(\mathbf{x})$ are optimized on $N = 10000$ observations $\mathbf{y}$. The ground truth source distributions $p(\mathbf{x})$ are $\mathcal{U}(-3, 3)^5$ for SLCP, $\mathcal{U}(-1, 1)^2$ for two-moons and $\mathcal{N}(\mathbf{0}, \mathbf{Diag}(\frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}))$ for IK.

**Biased vs. unbiased estimator** We first compare the biased and unbiased estimators $\mathcal{L}_K$ and $\hat{\mathcal{L}}_K$. Table 1 reports the ROC AUC scores of a classifier trained to distinguish between noise-corrupted observations from the ground truth $p(\mathbf{y})$ and noise-corrupted observations from the marginal $\int p(\mathbf{y}|\mathbf{x})q_\theta(\mathbf{x})d\mathbf{x}$ obtained by passing source data from the $q_\theta(\mathbf{x})$ into the exact simulator $p(\mathbf{y}|\mathbf{x})$. For both estimators, the table shows that we successfully identify a source distribution $q_\theta(\mathbf{x})$ resulting in a distribution over noise-corrupted observations which is almost indistinguishable from the ground truth $p(\mathbf{y})$. When $K$ is low, de-biasing the estimator leads to significant improvements. When $K$ increases, the bias of $\mathcal{L}_K$ drops quickly, and de-biasing, which introduces variance, does not significantly improve the results. Therefore, we recommend using the de-biased estimator when $K$ is constrained to be low, e.g., when the GPU memory is limited. In the following, we set $K = 1024$ and only consider the biased estimator $\mathcal{L}_K(\theta)$.

**Monte Carlo vs. variational methods** We evaluate the quality of $\mathcal{L}_K$, $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^{\text{IW}}_K$. For a fair comparison, we use an Unconstrained Monotonic Neural Network autoregressive flow (UMNN-MAF, Wehenkel and Louppe, 2019) to parameterize the prior for all losses. The recognition network $q_\psi(\mathbf{x}|\mathbf{y})$ for the variational approaches is modeled with the same architecture as the prior, but is conditioned on $\mathbf{y}$. We use $K = 128$ for $\mathcal{L}^{\text{IW}}_K$ due to GPU memory constraints. We show in Appendix E that simpler implicit generative models can be used with the Monte Carlo estimators $\mathcal{L}_K$ and $\hat{\mathcal{L}}_K$, effectively reducing inference time and allowing to use higher values of $K$.

Table 2 shows the ROC AUC of a classifier trained to discriminate between samples from the ground truth source distribution $p(\mathbf{x})$ and samples from the source distribution $q_\theta(\mathbf{x})$ identified by each of the different methods. A ROC AUC score between 0.5 and 0.7 is often considered poor discriminative performance, therefore indicating good source estimation. The estimator $\mathcal{L}_{1024}$ leads to the most accurate source distributions on these three tasks. In particular, the source distribution found for the two-moons problem is almost perfect. At the same time, the results for SLCP and IK are only barely acceptable, although largely better than for the variational methods ($\mathcal{L}^{\text{IW}}_K$ and $\mathcal{L}^{\text{ELBO}}$). Figures 1 and 2 illustrate for $\mathcal{L}_{1024}$ how the exact and learned sources distributions are visually similar.

Table 2 also reports the discrepancy between the corrupted data from the identified source distributions and the ground truth distribution of noise-corrupted observations. While $\mathcal{L}^{\text{ELBO}}$ does not give good results for SLCP, tightening the evidence lower-bound with $\mathcal{L}^{\text{IW}}_{128}$ yields good results on all problems. While $\mathcal{L}_{1024}$ has similar performance to $\mathcal{L}^{\text{IW}}_{128}$ on SLCP and two-moons, it is performing worse for IK, due to the difficulty of approximating $\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$ from Monte Carlo integration since the likelihood function for this problem is almost a Dirac function (see Appendix C for more details).

After observing the different estimators' reconstruction quality, the superiority of $\mathcal{L}_{1024}$ on source estimation over variational methods may be surprising at first glance. However, the variational methods require learning both a source distribution and a recognition network that are consistent with the likelihood function and the observations. This means that a wrong recognition network may prevent learning the correct source distribution as they must be consistent with each other. In the three experiments analyzed here,

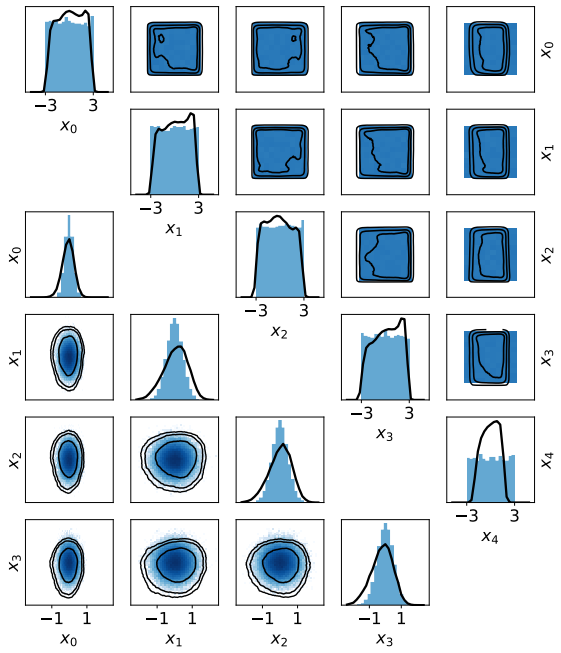


Figure 1: Source estimation results for $\mathcal{L}_{1024}$ on SLCP (top) and IK (bottom). The source distribution $p(\mathbf{x})$ are shown in blue against the estimated source distribution $q_\theta(\mathbf{x})$ in black (the 68-95-99.7% contours are shown). *The identified source distributions are similar to the unseen source distributions.*

the prior distribution is a simple unimodal continuous distribution, whereas the posteriors are discontinuous and multimodal. In these cases, learning only the source distribution is simpler than learning both a source and posterior distributions.

**Symmetric source distribution** As mentioned before, multiple optimal solutions may co-exist when the inverse problem is ill-posed. On close inspection, figures 1 and 2 show that NEB successfully recovers the domain of the source data but fails to exactly reproduce the ground truth source distribution. Indeed, for all problems considered here, the passage of $\mathbf{x}$ through the corruption process results in a loss of information in $\mathbf{y}$, which may lead to multiple solutions. We observe this in Figure 2, where we plot the quantities $|\mathbf{x}_1 + \mathbf{x}_2|$ and $-\mathbf{x}_1 + \mathbf{x}_2$ that are sufficient statistics of $\mathbf{x}$ for estimating $\mathbf{y}$. We see that the distribution over these intermediate variables is nearly equal for the ground truth distribution and the identified source distribution. This indicates that, up to symmetries, NEB recovers the ground truth source distribution.

A reasonable way to encourage learning a good source distribution is to enforce *a priori* known properties such as its domain, symmetries, or smoothness. This type of useful inductive biases can be embedded in the

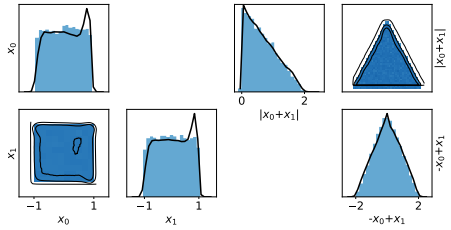


Figure 2: Source estimation results for $\mathcal{L}_{1024}$ on the two-moons problem. The source distribution $p(\mathbf{x})$ is shown in blue against the estimated source distribution $q_\theta(\mathbf{x})$ in black (the 68-95-99.7% contours are shown). *As shown on the right, up to the symmetries of the problem, the identified source distribution matches the unseen source distribution.*

neural network to constrain the solution space. As such, we modify the UMMN-MAF networks $q_\theta(\mathbf{x})$ so that the generated distributions are one-to-one symmetric, i.e. $q_\theta([x_1, ..., x_d]) = q_\theta([\pm x_1, ..., \pm x_d])$ (these modifications are detailed in Appendix H). Table 2 shows that the symmetric distributions are more similar to the unseen distributions $p(\mathbf{x})$ in all but one case. For example, all methods learn to approximately identify the exact source distribution on the two-moons despite the simulator's destructive process. Results for $\mathcal{L}_{1024}$ on SLCP are worse because the regularization pushes the learned distribution to a solution that still reproduces the observed distribution with high accuracy (the ROC AUC between the observed distribution and the regenerated one drops to $0.51_{\pm 0.01}$), but that moves away from the unseen source distribution. Further inductive bias should therefore be introduced; for example, the learned distribution can be bounded using specific activation functions in the last layer of $\mathbf{G}_\theta(\cdot)$. We note here that the non-variational methods are generally better suited for inductive bias as they do not require to evaluate the density $q_\theta(\mathbf{x})$ but only to sample from it and thus could be parameterized by any form of generative model.

### 5.2 Likelihood-free posterior inference

In the context of Bayesian posterior inference, the source distribution we retrieve with NEB can used as a prior distribution. Therefore, the learned prior $q_\theta(\mathbf{x})$ together with the surrogate likelihood $q_\phi(\mathbf{y}|\mathbf{x})$ unlock the subsequent likelihood-free estimation of the posterior $p(\mathbf{x}|\mathbf{y})$ – for which the fidelity will depend on both the prior's and the likelihood surrogate's correctness. Notably, posterior inference may be done efficiently with rejection sampling or importance sampling. In the following experiments, we perform rejection sampling as follows: given $u \sim \mathcal{U}(0, 1)$, we accept samples $\mathbf{x} \sim q_\theta(\mathbf{x})$ such that $u < \frac{q_\phi(\mathbf{y}|\mathbf{x})}{M}$ where

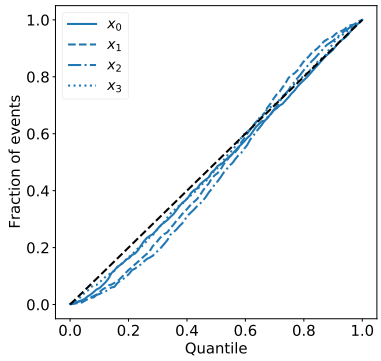



Figure 3: Posterior inference for IK. The plot shows the per-parameter calibration curves obtained with rejection sampling using the learned likelihood $q_\psi(\mathbf{y}|\mathbf{x})$ and the identified source distribution $q_\theta(\mathbf{x})$ with $\mathcal{L}_{1024}$. *The curves indicate a reasonably well calibrated posterior distribution.*

$M > q_\phi(\mathbf{y}|\mathbf{x}), \forall \mathbf{x}$ is determined empirically. As both $q_\theta(\mathbf{x})$ and $q_\phi(\mathbf{y}|\mathbf{x})$ are modeled with neural networks that allow efficient parallel computation, the sampling procedure is particularly fast. On the other hand, variational approaches ($\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^{\text{IW}}_{128}$) directly learn a posterior $q_\psi(\mathbf{x}|\mathbf{y})$ as a function of the single observation $\mathbf{y}$, which enables immediate per-event posterior inference. For $\mathcal{L}^{\text{IW}}_{128}$, Cremer et al. (2017) suggest using importance sampling.

| 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 3: Calibration test from 1,000 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.*

We assess the goodness of the posterior distributions with a calibration test. Inspired by Bellagente et al. (2020), for multiple observations $\mathbf{y}_i$, we approximate the 1D posterior distributions and report the fraction of events as a function of the quantile to which the generating source data $\mathbf{x}_i$ fall. Figure 3 reports calibration curves associated with $\mathcal{L}_{1024}$ for the IK problem. It indicates a well-calibrated posterior distribution. Table 3 reports the Kolmogorov–Smirnov test between the generated curves and the expected results; we expect that $x\%$ of events belong to the $x\%$ quantile. The table reports the mean over all dimensions of this calibration metric. The posterior distribution from $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^{\text{IW}}_{128}$ are not consistently well calibrated in opposition to the posterior distribution of $\mathcal{L}_{1024}$.

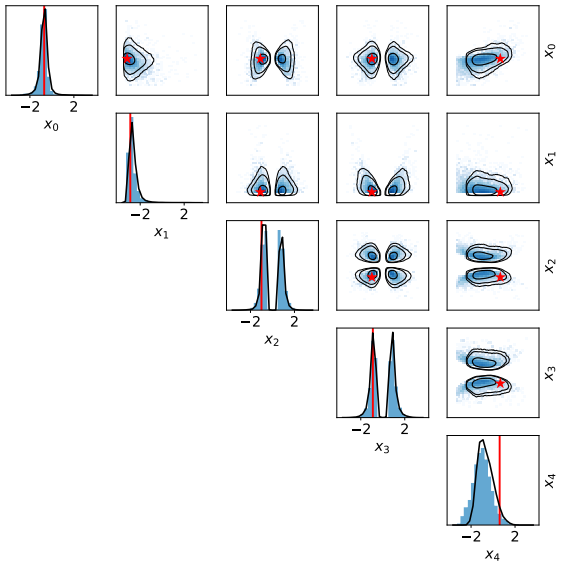
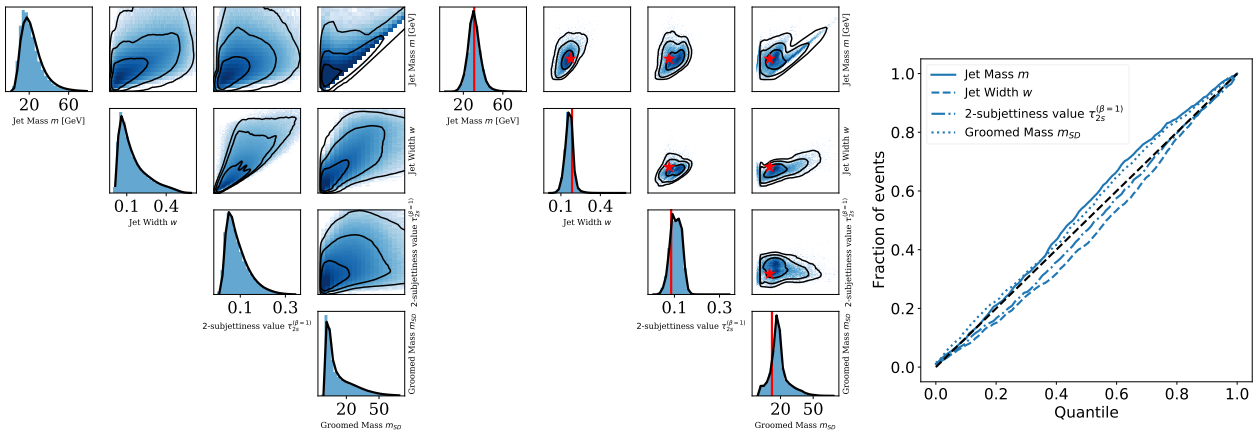



Figure 4: Posterior distribution obtained from MCMC with the exact source distribution and the exact likelihood function on SLCP in blue against the posterior distribution obtained with $q_\phi(\mathbf{y}|\mathbf{x})$ and $q_\theta(\mathbf{x})$ learned from $\mathcal{L}_{1024}$ in black (the 68-95-99.7% contours are shown). Generating source sample $\mathbf{x}$ are indicated in red. *The approximated posterior distribution closely matches the ground truth.*

Finally, we show in Figure 4 an example of posterior distribution obtained with rejection sampling using $q_\theta(\mathbf{x})$ and $q_\phi(\mathbf{y}|\mathbf{x})$ as learned with $\mathcal{L}_{1024}$, against the ground truth posterior obtained with Markov Chain Monte Carlo using the exact likelihood and source data distribution. We emphasize that NEB can recover nearly the exact posterior with no access to the likelihood function or to the prior distribution. To the best of our knowledge, this is the first work to show posterior inference is possible in this extreme setting.

### 5.3 Detector correction in collider physics

At colliders like the LHC, the distribution of particles produced from an interaction and incident on detectors can be predicted from theoretical models. Thus measurements of such distributions can be used to directly test theoretical predictions. However, while detectors measure the energy and momentum of parti-




(a) (b) (c)

Figure 5: Neural Empirical Bayes for detector correction in collider physics. (a) The source distribution $p(\mathbf{x})$ is shown in blue against the estimated source distribution $q_\theta(\mathbf{x})$ in black. (b) Posterior distribution obtained with rejection sampling, with generating source sample $\mathbf{x}$ indicated in red. (c) Calibration curves for each jet property obtained with rejection sampling on 10000 observations. In (a) and (b), contours represent the 68-95-99.7% levels.

cles, they also induce noise due to the stochastic nature of particle-material interactions and of the signal acquisition process. Thus a key challenge in comparing measurements to theoretical predictions is to correct noisy detector observations to obtain experimentally observed incident particle source distributions. This is frequently done by binning 1D or 2D distributions and solving a discrete linear inverse problem. Instead we apply NEB for estimating the multi-dimensional source distribution. We use the publicly available simulated dataset (Andreassen et al., 2019a) of paired source and corrupted measurements of properties of jets, or collimated streams of particles produce by high energy quarks and gluons. Simulation details are found in Appendix G.

Surrogate training was performed using one source simulator as a proposal distribution. The same surrogate architecture and hyperparameters as in the toy experiments were used (see Appendix D for details). We assess NEB in a dataset with the source distribution produced by a different simulator of the same physical process. Both datasets for surrogate training and source distribution learning contain approximately 1.6 million events. This is an example setting where sequential inference methods cannot be used as only a fixed dataset is available and not the simulator.

**Source estimation** We focus on the $\mathcal{L}_{1024}$ estimator for source distribution learning although we also report results with the other estimators. Optimization is done with Adam using default parameters and an initial learning rate of $10^{-4}$. We train for 10 epochs with minibatches of size 256. The density estimator for the source distribution comprised 6 coupling layers, with 3-layer MLPs with 32 units per layer and ReLU activations used for the scaling and translation functions. Parameters were determined with a hyperparameter grid search using a held out validation set from the dataset on which the surrogate is trained. The learned source distribution is observed to closely match the true simulated source distribution, as seen in Figure 5a. Table 4 reports the ROC AUC between the learned and ground truth distributions, indicating only small discrepancies between them.

| | $\mathcal{L}^{\text{ELBO}}$ | $\mathcal{L}^{\text{IW}}_{128}$ | $\mathcal{L}_{1024}$ |
|---|---|---|---|
| $\mathbf{x}$-space | $0.99_{\pm 0.02}$ | $0.63_{\pm 0.06}$ | $0.57_{\pm 0.05}$ |
| $\mathbf{y}$-space | $0.87_{\pm 0.08}$ | $0.51_{\pm 0.01}$ | $0.50_{\pm 0.01}$ |

Table 4: Source estimation in collider physics. ROC AUC between $q_\theta(\mathbf{x})$ and the unseen source distribution $p(\mathbf{x})$ ($\mathbf{x}$-space), and between the observed distribution $p(\mathbf{y})$ and the regenerated distribution $\int q_\phi(\mathbf{y}|\mathbf{x})q_\theta(\mathbf{x})d\mathbf{x}$.

**Likelihood-free posterior inference** Figure 5b shows the learned posterior distribution against the generating source data. Plots are scaled to the prior-space. The model learns nicely a region of plausible values for the generating source data. To assess the quality of the posterior inference on more data, Figure 5c shows the fraction of events as a function of the quantile to which the generating source data belongs under the learned posterior distribution. Results indicate reasonably well calibrated posterior distributions.

## 6 Summary and discussion

In this work, we revisit $g$-modeling empirical Bayes with neural networks to estimate source distributions from non-linearly corrupted observations. We propose both a biased and de-biased estimator of the log-marginal likelihood, and examine variational methods for this challenge. We show that we can successfully recover source distributions from corrupted observations. We find that inductive bias is highly beneficial for solving ill-posed inverse problems and can be embedded in the structure of the neural networks used to model the source distribution. Although the explored approaches are general, we specifically study the likelihood-free setting, and we successfully perform posterior inference without direct access to either a likelihood function or a prior distribution.

**Future work** In this work we have mainly examined low-dimensional settings. We believe that further analysis of these methods for high-dimensional data such as images and time series could be of strong interest from both a theoretical and practical point of view. In particular, assessing the computational challenges of each method and the importance of inductive bias in this challenging setting are promising directions towards improvements in solving high-dimensional inverse problems.

### Acknowledgments

We thank Johann Brehmer and Kyle Cranmer for their helpful feedback on the manuscript. Antoine Wehenkel is a research fellow of the F.R.S.-FNRS (Belgium) and acknowledges its financial support. Gilles Louppe is recipient of the ULiège - NRB Chair on Big data and is thankful for the support of NRB. Michael Kagan and Maxime Vandegar are supported by the US Department of Energy (DOE) under grant DE-AC02-76SF00515, and Michael Kagan is also supported by the SLAC Panofsky Fellowship.

### References

ATLAS Pythia 8 tunes to 7 TeV datas. Technical Report ATL-PHYS-PUB-2014-021, CERN, Geneva, Nov 2014.

Tim Adye. Unfolding algorithms and tests using RooUnfold. In *PHYSTAT 2011*, pages 313–318, Geneva, 2011. CERN. doi: 10.5170/CERN-2011-006.313.

Anders Andreassen, Patrick Komiske, Eric Metodiev, Benjamin Nachman, and Jesse Thaler. Pythia/Herwig + Delphes Jet Datasets for OmniFold Unfolding, November 2019a.

Anders Johan Andreassen, Patrick T. Komiske, Eric M. Metodiev, Benjamin Nachman, and Jesse Thaler. Omnifold: A method to simultaneously unfold all observables. 2019b.

Jordanka A Angelova. On moments of sample mean and variance. *Int. J. Pure Appl. Math*, 79(1):67–85, 2012.

Lynton Ardizzone, Jakob Kruse, Sebastian J. Wirkert, Daniel Rahner, Eric W. Pellegrini, Ralf S. Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. *CoRR*, abs/1808.04730, 2018.

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *CoRR*, abs/1907.02392, 2019.

ATLAS Collaboration. The ATLAS Experiment at the CERN Large Hadron Collider. *Journal of Instrumentation*, 3(08):S08003–S08003, aug 2008.

Manuel Bahr, S. Gieseke, M. A. Gigg, David Grellscheid, K. Hamilton, O. Latunde-Dada, Simon Platzer, P. Richardson, Mike H. Seymour, A. Sherstnev, and Bryan Webber. Herwig++ 2.1 release note. 1999.

Marco Bellagente, Anja Butter, Gregor Kasieczka, Tilman Plehn, Armand Rousselot, Ramon Winterhalder, Lynton Ardizzone, and Ullrich Köthe. Invertible networks or partons to detector and back again, 2020.

Volker Blobel. Unfolding Methods in Particle Physics. pages 240–251. 12 p, Jan 2011. doi: 10.5170/CERN-2011-006.240.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2016.

Manuel Bähr, Stefan Gieseke, Martyn A. Gigg, David Grellscheid, Keith Hamilton, Oluseyi Latunde-Dada, Simon Plätzer, Peter Richardson, Michael H. Seymour, Alexander Sherstnev, and et al. Herwig++ physics and manual. 58(4):639–707, Nov 2008.

Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, pages 9916–9926, 2019.

CMS Collaboration. The CMS experiment at the CERN LHC. *Journal of Instrumentation*, 3(08): S08004–S08004, aug 2008.

G. Cowan. A survey of unfolding methods for particle physics. *Conf. Proc. C*, 0203181:248–257, 2002.

Kyle Cranmer. Neural unfolding. 2018. URL https://github.com/cranmer/neural_unfolding.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 2020.

Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders, 2017.

G. D'Agostini. A multidimensional unfolding method based on bayes' theorem. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 362(2):487 – 498, 1995. ISSN 0168-9002.

J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi. Delphes 3: a modular framework for fast simulation of a generic collider experiment. *Journal of High Energy Physics*, 2014(2), Feb 2014.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Peter J. Diggle and Richard J. Gratton. Monte carlo methods of inference for implicit statistical models. 46(2):193–227, 1984.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *ArXiv*, abs/1605.08803, 2017.

Tim Dockhorn, James A. Ritchie, Yaoliang Yu, and Iain Murray. Density deconvolution with normalizing flows, 2020.

Conor Durkan, Iain Murray, and George Papamakarios. On contrastive learning for likelihood-free inference. In *International Conference on Machine Learning*, 2020.

Bradley Efron. Two modeling strategies for empirical bayes estimation. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 29(2): 285, 2014.

Bradley Efron. Empirical bayes deconvolution estimates. *Biometrika*, 103:1–20, 03 2016.

Lyndon Evans and Philip Bryant. LHC machine. *Journal of Instrumentation*, 3(08):S08001–S08001, aug 2008.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. volume 37 of *Proceedings of Machine Learning Research*, pages 881–889, Lille, France, 07–09 Jul 2015. Proceedings of Machine Learning Research.

David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic posterior transformation for likelihood-free inference. In *Proceedings of Machine Learning Research*, 2019.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free mcmc with amortized approximate ratio estimators. In *International Conference on Machine Learning*, 2020.

Herman Kahn. Use of different monte carlo sampling techniques, 1955.

Gilles Louppe, Joeri Hermans, and Kyle Cranmer. Adversarial variational optimization of non-differentiable simulators. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1438–1447, 2019.

L. B. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79:745, June 1974. doi: 10.1086/111605.

Jan-Matthis Lueckmann, Pedro J. Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H. Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, 2017.

Yucen Luo, Alex Beatson, Mohammad Norouzi, Jun Zhu, David Duvenaud, Ryan P. Adams, and Ricky T. Q. Chen. Sumo: Unbiased estimation of log marginal probability for latent variable models, 2020.

Balasubramanian Narasimhan and Bradley Efron. A g-modeling program for deconvolution and empirical bayes estimation. 2016.

Sebastian Nowozin. Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference. In *International Conference on Learning Representations*, 2018.

George Papamakarios and Iain Murray. Fast $\epsilon$-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*. 2016.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. volume 89 of *Proceedings of Machine Learning Research*, pages 837–848. PMLR, 16–18 Apr 2019.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

Herbert Robbins. An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 157–163, 1956.

Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An introduction to pythia 8.2. *Computer Physics Communications*, 191:159 – 177, 2015.

Esteban G Tabak, Eric Vanden-Eijnden, et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1): 217–233, 2010.

Yixin Wang, Andrew C. Miller, and David M. Blei. Comment: Variational autoencoders as empirical bayes. *Statist. Sci.*, 34(2):229–233, 05 2019.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. In *Advances in Neural Information Processing Systems*, pages 1545–1555, 2019.

# A Summary of the notations used in the paper

All notations used in the paper are summarized in Table 5.

| Notation | Definition | Example |
|---|---|---|
| $s(\cdot;\mathbf{x})$ | Stochastic process of a simulator that maps source data $\mathbf{x}$ to observations $\mathbf{y}$ | A particle detector in physics |
| $p(\mathbf{y}\|\mathbf{x})$ | Likelihood function implicitly defined by $s(\cdot;\mathbf{x})$ | - |
| $q_\phi(\mathbf{y}\|\mathbf{x})$ | Surrogate model of $p(\mathbf{y}\|\mathbf{x})$ | A parametrized density estimator |
| $p(\mathbf{y})$ | Observed distribution | Some particle energy distribution observed at the detector level |
| $q_\theta(\mathbf{y})$ | Estimator of $p(\mathbf{y})$ | The evidence lower bound |
| $p(\mathbf{x})$ | Unseen source distribution that has generated $p(\mathbf{y})$ | The exact particle energy distribution |
| $q_\theta(\mathbf{x})$ | Surrogate model of $p(\mathbf{x})$ | A neural network |
| $q_\phi(\mathbf{x}\|\mathbf{y})$ | Variational posterior distribution | A parametrized density estimator |
| $\pi(\mathbf{x})$ | Proposal distribution used to generate a dataset in order to train $q_\phi(\mathbf{y}\|\mathbf{x})$ | Any probability distribution |

Table 5: 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:

$$\mathbb{E}\left[\mathcal{L}_K\right] = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\log p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] \tag{9a}$$

$$= \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\log p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] \tag{9b}$$

$$\leq \frac{1}{K}\sum_{k=1}^{K}\log\mathbb{E}\left[p(\mathbf{y}|\mathbf{G}_\theta(\boldsymbol{\epsilon}_k))\right] \tag{9c}$$

$$= \log q_\theta(\mathbf{y}) \tag{9d}$$

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 $\gamma_i$ with the central moments $\mu_i$ 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:

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

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

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

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

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

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:

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

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

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

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

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

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

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

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., 2014, 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 $\alpha$ 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 6.

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

Table 6: 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 7.

$\mathcal{L}_K$ 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 $\mathcal{L}^{\text{IW}}_K$ 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 | $3 \times 75$ (MADE) |
| Integrand network | $3 \times 75$ (MLP) |
| N°flows | 6 |
| Embedding Size | 10 |
| Batch size | 128 |
| Optimizer | Adam |
| Weight decay | 0.0 |
| Learning rate | $10^{-4}$ |

Table 7: Hyperparameters used to train and model $q_\theta(\mathbf{x})$ and $q_\psi(\mathbf{x}|\mathbf{y})$.

## E 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 $\theta$ 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 8: 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).

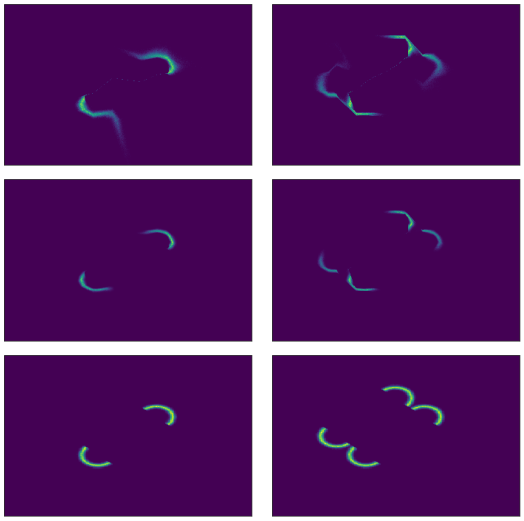
Table 8 reports the discrepancy between the corrupted data from the identified source distributions and the ground truth distribution of noise-corrupted observations ($\mathbf{y}$-space). It shows that simple architectures allow to learn a source distribution that can closely reproduce the observed distribution. 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.

## F 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. While in general this is not restrictive, as in science many events are observed at the same time, Figure 6 shows interestingly 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.

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.



(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.

## G 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.

## H 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. □