Advanced Machine Learning

Lecture 5

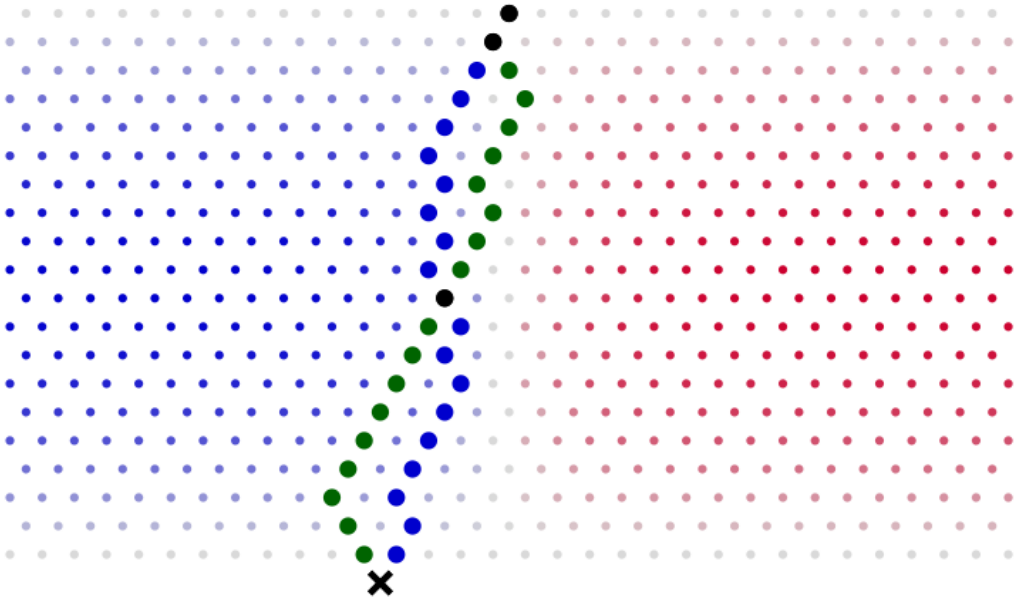
Paper: Gilles Louppe, Joeri Hermans and Kyle Cranmer,
Adversarial Variational Optimization of Non-Differentiable Simulators,
in International Conference on Artificial Intelligence and Statistics 2019.

Prof. Gilles Louppe
g.louppe@uliege.be

LIÈGE université

Likelihood-free inference


GALTON
BOARD
@physicsfun



The probability of ending in bin x corresponds to the total probability of all the paths z from start to x.

$$p(x|\theta) = \int p(x, z|\theta) dz = \binom{n}{x} \theta^x (1-\theta)^{n-x}$$

Inference

Given a set of realizations $\mathbf{d} = \{x_i\}$ at the bins, inference consists in determining the value of θ that best describes these observations.

For example, following the principle of maximum likelihood estimation, we have

$$\hat{\theta} = \arg\max_{\theta} \prod_{x_i \in \mathbf{d}} p(x_i|\theta).$$

In general, when $p(x_i|\theta)$ can be evaluated, this problem can be solved either analytically or using optimization algorithms.

What if we shift or remove some of the pins?

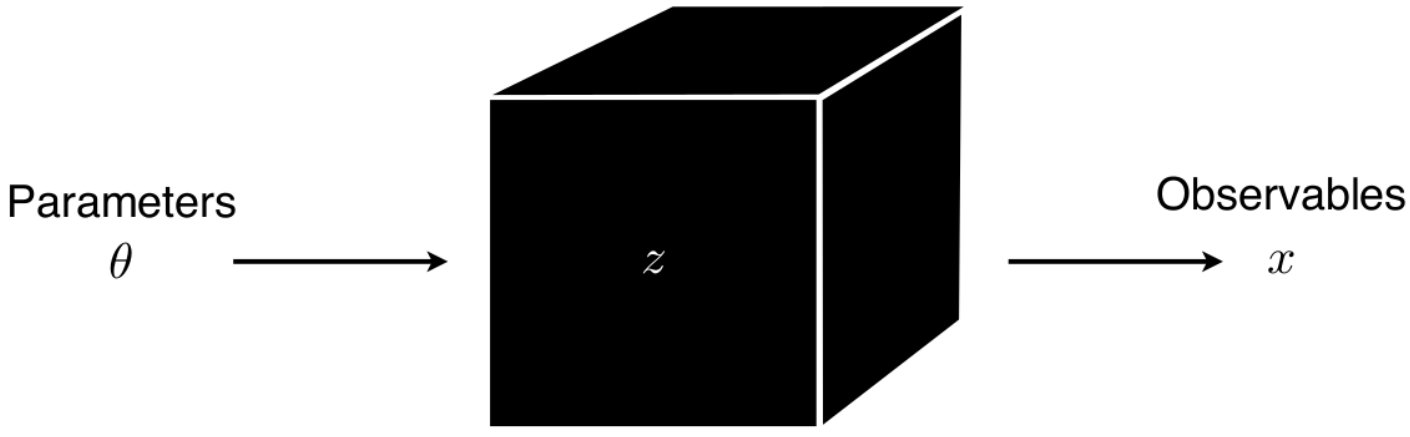
$$p(x|\theta) = \underbrace{\int}_{\text{intractable!}} p(x, z|\theta)dz$$
$$\neq \binom{n}{x} \theta^x (1-\theta)^{n-x}$$

Does this mean inference is no longer possible?

The Galton board is a metaphore of simulation-based science:

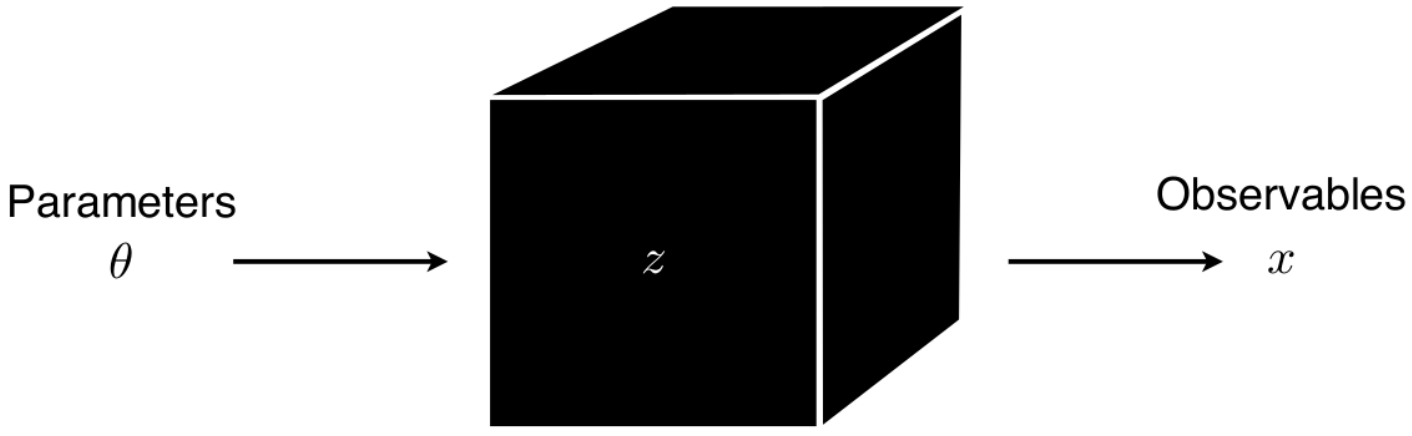
Galton board device	$\rightarrow$	Computer simulation
Parameters θ	$\rightarrow$	Model parameters θ
Buckets x	$\rightarrow$	Observables x
Random paths z	$\rightarrow$	Latent variables z (stochastic execution traces through simulator)

Inference in this context requires likelihood-free algorithms.



Prediction (simulation):

- Well-understood mechanistic model
- Simulator can generate samples



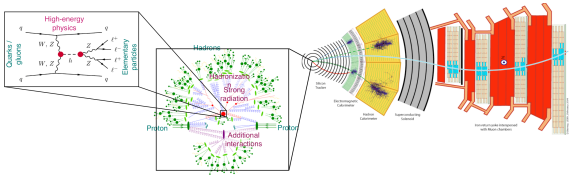
Prediction (simulation):

- Well-understood mechanistic model
- Simulator can generate samples

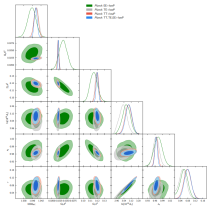
Inference:

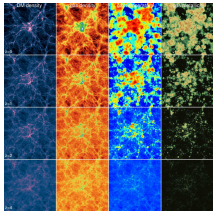
- Likelihood function $p(x|\theta)$ is intractable
- Goal: estimator $\hat{p}(x|\theta)$

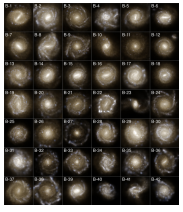
Applications



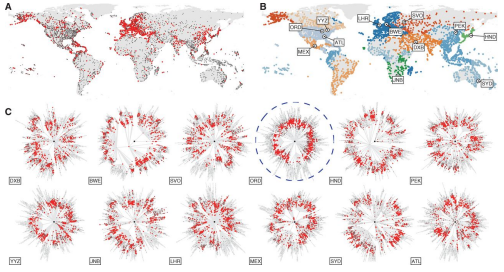
Particle physics



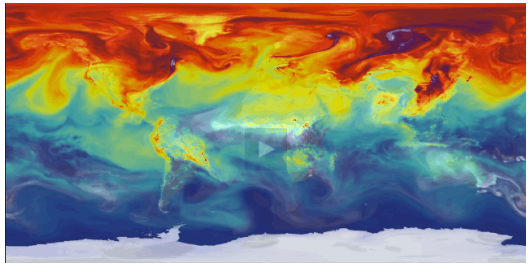




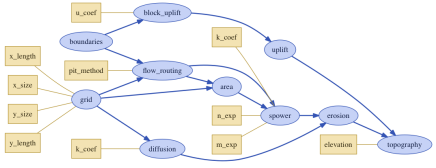
Cosmology

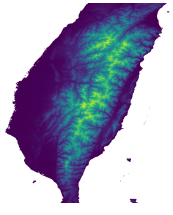


Epidemiology



Climatology

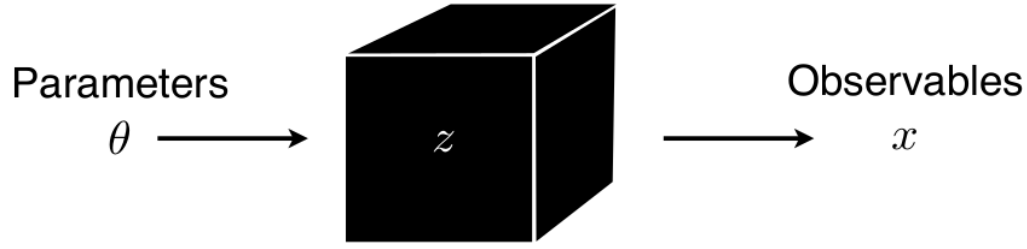




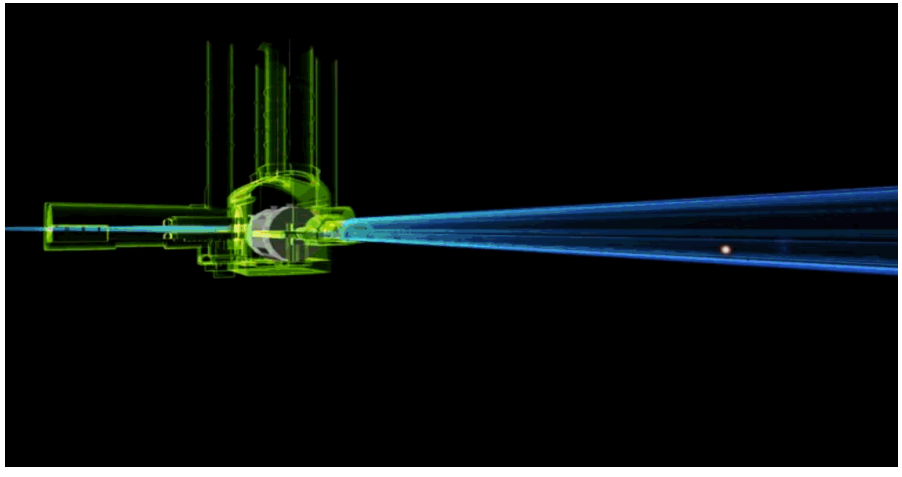
Computational topography

Astronomy

Particle physics



$$\mathcal{L}_{SM} = -\tfrac{1}{2}\partial_\nu g^a_\mu \partial_\nu g^a_\mu - g_s f^{abc}\partial_\mu g^a_\nu g^b_\mu g^c_\nu - \tfrac{1}{2}g_s^2 f^{abc} f^{ade} g^b_\mu g^c_\nu g^d_\mu g^e_\nu - \partial_\nu W^+_\mu \partial_\nu W^-_\mu - M^2 W^+_\mu W^-_\mu - \tfrac{1}{2}\partial_\nu Z^0_\mu \partial_\nu Z^0_\mu - \tfrac{1}{2c_w^2} M^2 Z^0_\mu Z^0_\mu - \tfrac{1}{2}\partial_\mu A_\nu \partial_\mu A_\nu - igc_w(\partial_\nu Z^0_\mu (W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - Z^0_\mu (W^+_\nu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu) + Z^0_\nu (W^+_\mu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu)) - igs_w(\partial_\nu A_\mu (W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - A_\nu (W^+_\mu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu) + A_\mu (W^+_\nu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu)) - \tfrac{1}{2}g^2 W^+_\mu W^-_\mu W^+_\nu W^-_\nu + \tfrac{1}{2}g^2 W^+_\mu W^-_\nu W^+_\mu W^-_\nu + g^2 c_w^2 (Z^0_\mu W^+_\mu Z^0_\nu W^-_\nu - Z^0_\mu Z^0_\mu W^+_\nu W^-_\nu) + g^2 s_w^2 (A_\mu W^+_\mu A_\nu W^-_\nu - A_\mu A_\mu W^+_\nu W^-_\nu) + g^2 s_w c_w (A_\mu Z^0_\nu (W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - 2A_\mu Z^0_\mu W^+_\nu W^-_\nu) - \tfrac{1}{2}\partial_\mu H \partial_\mu H - 2M^2 \alpha_h H^2 - \partial_\mu \phi^+ \partial_\mu \phi^- - \tfrac{1}{2}\partial_\mu \phi^0 \partial_\mu \phi^0 - \beta_h \left(\tfrac{2M^2}{g^2} + \tfrac{2M}{g} H + \tfrac{1}{2}(H^2 + \phi^0\phi^0 + 2\phi^+\phi^-) \right) + \tfrac{2M^4}{g^2}\alpha_h - g\alpha_h M (H^3 + H\phi^0\phi^0 + 2H\phi^+\phi^-) - \tfrac{1}{8}g^2 \alpha_h (H^4 + (\phi^0)^4 + 4(\phi^+\phi^-)^2 + 4(\phi^0)^2\phi^+\phi^- + 4H^2\phi^+\phi^- + 2(\phi^0)^2 H^2) - gMW^+_\mu W^-_\mu H - \tfrac{1}{2}g\tfrac{M}{c_w^2} Z^0_\mu Z^0_\mu H - \tfrac{1}{2}ig\left(W^+_\mu(\phi^0\partial_\mu\phi^- - \phi^-\partial_\mu\phi^0) - W^-_\mu(\phi^0\partial_\mu\phi^+ - \phi^+\partial_\mu\phi^0)\right) + \tfrac{1}{2}g\left(W^+_\mu(H\partial_\mu\phi^- - \phi^-\partial_\mu H) + W^-_\mu(H\partial_\mu\phi^+ - \phi^+\partial_\mu H)\right) + \tfrac{1}{2}g\tfrac{1}{c_w}(Z^0_\mu(H\partial_\nu\phi^0 - \phi^0\partial_\mu H) + M(\tfrac{1}{c_w}Z^0_\mu\partial_\mu\phi^0 + W^+_\mu\partial_\mu\phi^- + W^-_\mu\partial_\mu\phi^+) - ig\tfrac{s_w^2}{c_w}MZ^0_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) + igs_wMA_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) - ig\tfrac{1-2c_w^2}{2c_w}Z^0_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) + igs_wA_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) - \tfrac{1}{4}g^2W^+_\mu W^-_\mu(H^2 + (\phi^0)^2 + 2\phi^+\phi^-) - \tfrac{1}{8}g^2\tfrac{1}{c_w^2}Z^0_\mu Z^0_\mu(H^2 + (\phi^0)^2 + 2(2s_w^2 - 1)^2\phi^+\phi^-) - \tfrac{1}{2}g^2\tfrac{s_w^2}{c_w}Z^0_\mu\phi^0(W^+_\mu\phi^- + W^-_\mu\phi^+) - \tfrac{1}{2}ig^2\tfrac{s_w^2}{c_w}Z^0_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) + \tfrac{1}{2}g^2 s_w A_\mu\phi^0(W^+_\mu\phi^- + W^-_\mu\phi^+) + \tfrac{1}{2}ig^2 s_w A_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) - g^2\tfrac{s_w}{c_w}(2c_w^2 - 1)Z^0_\mu A_\mu\phi^+\phi^- + g^2 s_w^2 A_\mu A_\mu\phi^+\phi^- + \tfrac{1}{2}ig_s\lambda^a_{ij}(\bar{q}^\sigma_i\gamma^\mu q^\sigma_j)g^a_\mu - \bar{e}^\lambda(\gamma\partial + m^\lambda_e)e^\lambda - \bar{\nu}^\lambda(\gamma\partial + m^\lambda_\nu)\nu^\lambda - \bar{u}^\lambda_j(\gamma\partial + m^\lambda_u)u^\lambda_j - \bar{d}^\lambda_j(\gamma\partial + m^\lambda_d)d^\lambda_j + igs_wA_\mu\left(-(\bar{e}^\lambda\gamma^\mu e^\lambda) + \tfrac{2}{3}(\bar{u}^\lambda_j\gamma^\mu u^\lambda_j) - \tfrac{1}{3}(\bar{d}^\lambda_j\gamma^\mu d^\lambda_j)\right) + \tfrac{ig}{4c_w}Z^0_\mu\{(\bar{\nu}^\lambda\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar{e}^\lambda\gamma^\mu(4s_w^2 - 1 - \gamma^5)e^\lambda) + (\bar{d}^\lambda_j\gamma^\mu(\tfrac{4}{3}s_w^2 - 1 - \gamma^5)d^\lambda_j) + (\bar{u}^\lambda_j\gamma^\mu(1 - \tfrac{8}{3}s_w^2 + \gamma^5)u^\lambda_j)\} + \tfrac{ig}{2\sqrt{2}}W^+_\mu\left((\bar{\nu}^\lambda\gamma^\mu(1+\gamma^5)U^{lep}_{\lambda\kappa}e^\kappa) + (\bar{u}^\lambda_j\gamma^\mu(1+\gamma^5)C_{\lambda\kappa}d^\kappa_j)\right) + \tfrac{ig}{2\sqrt{2}}W^-_\mu\left((\bar{e}^\kappa U^{lep\dagger}_{\kappa\lambda}\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar{d}^\kappa_j C^\dagger_{\kappa\lambda}\gamma^\mu(1+\gamma^5)u^\lambda_j)\right) + \tfrac{ig}{2M\sqrt{2}}\phi^+\left(-m^\kappa_e(\bar{\nu}^\lambda U^{lep}_{\lambda\kappa}(1-\gamma^5)e^\kappa) + m^\lambda_\nu(\bar{\nu}^\lambda U^{lep}_{\lambda\kappa}(1+\gamma^5)e^\kappa)\right) + \tfrac{ig}{2M\sqrt{2}}\phi^-\left(m^\lambda_e(\bar{e}^\lambda U^{lep\dagger}_{\lambda\kappa}(1+\gamma^5)\nu^\kappa) - m^\kappa_\nu(\bar{e}^\lambda U^{lep\dagger}_{\lambda\kappa}(1-\gamma^5)\nu^\kappa)\right) - \tfrac{g}{2}\tfrac{m^\lambda_e}{M}H(\bar{e}^\lambda e^\lambda) + \tfrac{ig}{2}\tfrac{m^\lambda_e}{M}\phi^0(\bar{e}^\lambda\gamma^5 e^\lambda) - \tfrac{1}{4}\bar{\nu}_\lambda M^R_{\lambda\kappa}(1-\gamma_5)\bar{\nu}_\kappa - \tfrac{1}{4}\bar{\nu}_\lambda M^R_{\lambda\kappa}(1-\gamma_5)\bar{\nu}_\kappa + \tfrac{ig}{2M\sqrt{2}}\phi^+\left(-m^\kappa_d(\bar{u}^\lambda_j C_{\lambda\kappa}(1-\gamma^5)d^\kappa_j) + m^\lambda_u(\bar{u}^\lambda_j C_{\lambda\kappa}(1+\gamma^5)d^\kappa_j)\right) + \tfrac{ig}{2M\sqrt{2}}\phi^-\left(m^\lambda_d(\bar{d}^\lambda_j C^\dagger_{\lambda\kappa}(1+\gamma^5)u^\kappa_j) - m^\kappa_u(\bar{d}^\lambda_j C^\dagger_{\lambda\kappa}(1-\gamma^5)u^\kappa_j)\right) - \tfrac{g}{2}\tfrac{m^\lambda_u}{M}H(\bar{u}^\lambda_j u^\lambda_j) - \tfrac{g}{2}\tfrac{m^\lambda_d}{M}H(\bar{d}^\lambda_j d^\lambda_j) + \tfrac{ig}{2}\tfrac{m^\lambda_u}{M}\phi^0(\bar{u}^\lambda_j\gamma^5 u^\lambda_j) - \tfrac{ig}{2}\tfrac{m^\lambda_d}{M}\phi^0(\bar{d}^\lambda_j\gamma^5 d^\lambda_j) + \bar{G}^a\partial^2 G^a + g_s f^{abc}\partial_\mu\bar{G}^a G^b g^c_\mu + \bar{X}^+(\partial^2 - M^2)X^+ + \bar{X}^-(\partial^2 - M^2)X^- + \bar{X}^0(\partial^2 - \tfrac{M^2}{c_w^2})X^0 + \bar{Y}\partial^2 Y + igc_w W^+_\mu(\partial_\mu\bar{X}^0 X^- - \partial_\mu\bar{X}^+X^0) + igs_w W^+_\mu(\partial_\mu\bar{Y}X^- - \partial_\mu\bar{X}^+Y) + igc_w W^-_\mu(\partial_\mu\bar{X}^- X^0 - \partial_\mu\bar{X}^0X^+) + igs_w W^-_\mu(\partial_\mu\bar{X}^- Y - \partial_\mu\bar{Y}X^+) + igc_w Z^0_\mu(\partial_\mu\bar{X}^+X^- - \partial_\mu\bar{X}^-X^-) + igs_w A_\mu(\partial_\mu\bar{X}^+X^- - \partial_\mu\bar{X}^-X^-) - \tfrac{1}{2}gM\left(\bar{X}^+X^+H + \bar{X}^-X^-H + \tfrac{1}{c_w^2}\bar{X}^0X^0H\right) + \tfrac{1-2c_w^2}{2c_w}igM\left(\bar{X}^+X^0\phi^+ - \bar{X}^-X^0\phi^-\right) + \tfrac{1}{2c_w}igM\left(\bar{X}^0X^-\phi^+ - \bar{X}^0X^+\phi^-\right) + igMs_w\left(\bar{X}^0X^-\phi^+ - \bar{X}^0X^+\phi^-\right) + \tfrac{1}{2}igM\left(\bar{X}^+X^+\phi^0 - \bar{X}^-X^-\phi^0\right).$$

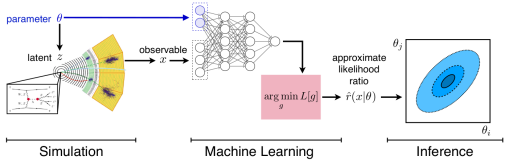


Algorithms

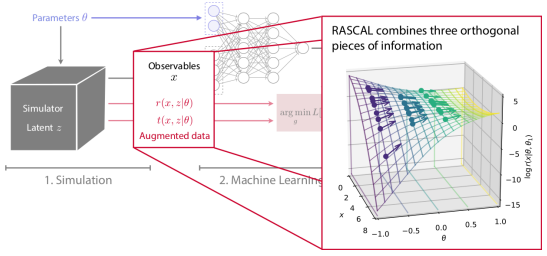
Treat the simulator as a black box

Make use of the inner structure

Learn a proxy for inference

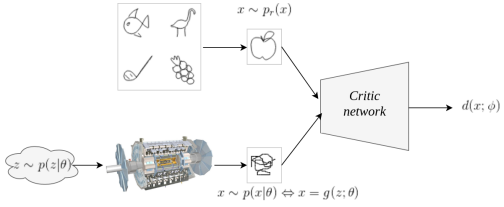


Histograms of observables
Neural density (ratio) estimation

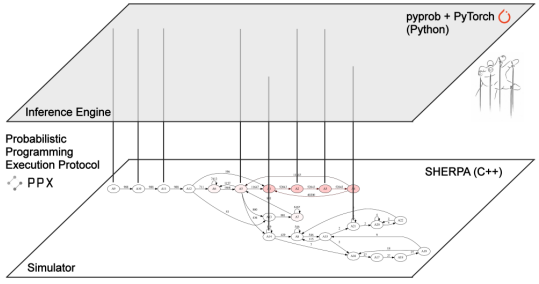


Mining gold from implicit models

Learn to control the simulator



Adversarial variational optimization

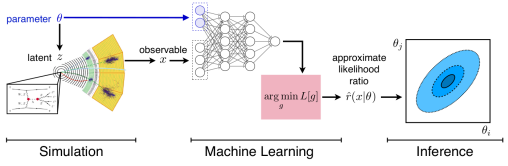


Probabilistic programming

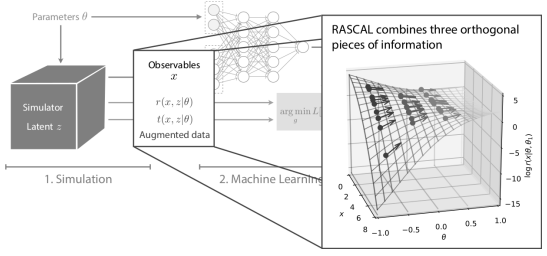
Treat the simulator as a black box

Make use of the inner structure

Learn a proxy for inference

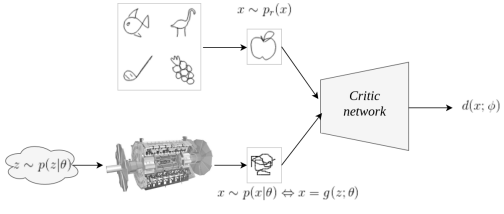


Histograms of observables
Neural density (ratio) estimation

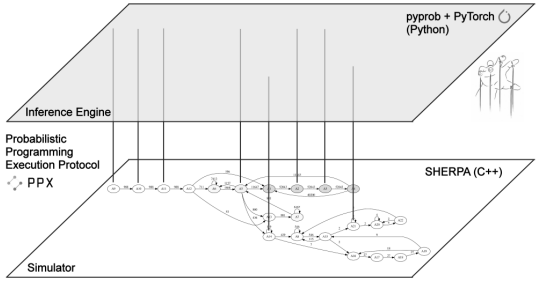


Mining gold from implicit models

Learn to control the simulator



Adversarial variational optimization



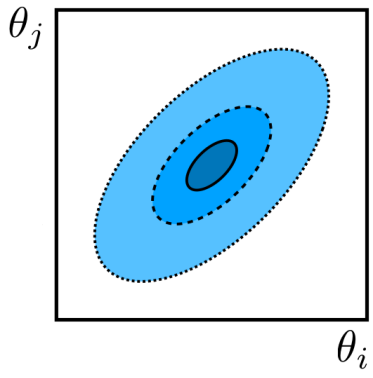
Probabilistic programming

The physicist's way

The Neyman-Pearson lemma states that the likelihood ratio

$$r(x|\theta_0, \theta_1) = \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

is the most powerful test statistic to discriminate between a null hypothesis θ_0 and an alternative θ_1.



IX. *On the Problem of the most Efficient Tests of Statistical Hypotheses.*

By J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. PEARSON, *Department of Applied Statistics, University College, London.*

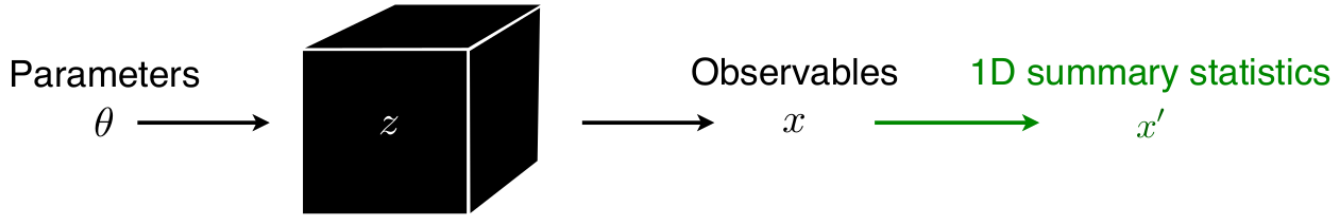
(*Communicated by* K. PEARSON, *F.R.S.*)

(Received August 31, 1932.—Read November 10, 1932.)

CONTENTS.

PAGE.

I. Introductory 289
II. Outline of General Theory 293



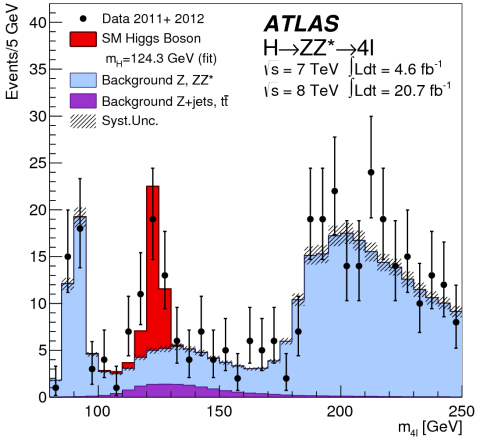
Define a projection function $s : \mathcal{X} \to \mathbb{R}$ mapping observables x to a summary statistics $x' = s(x)$.

Then, approximate the likelihood $p(x|\theta)$ as

$$p(x|\theta) \approx \hat{p}(x|\theta) = p(x'|\theta).$$

From this it comes

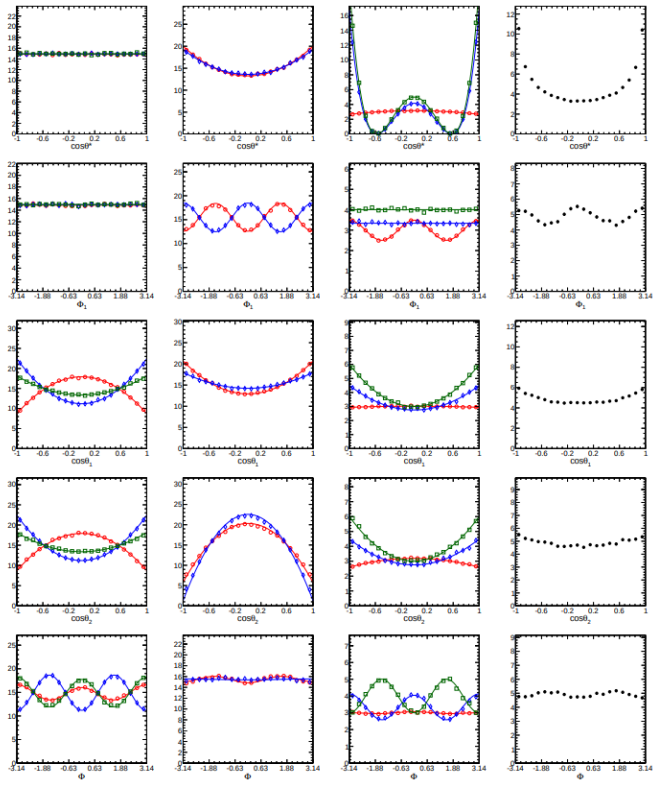
$$\frac{p(x|\theta_0)}{p(x|\theta_1)} \approx \frac{\hat{p}(x|\theta_0)}{\hat{p}(x|\theta_1)} = \hat{r}(x|\theta_0, \theta_1).$$



This methodology has worked great for physicists for the last 20-30 years, but ...

- Choosing the projection s is difficult and problem-dependent.
- Often there is no single good variable: compressing to any x' loses information.
- Ideally: analyse high-dimensional x', including all correlations.

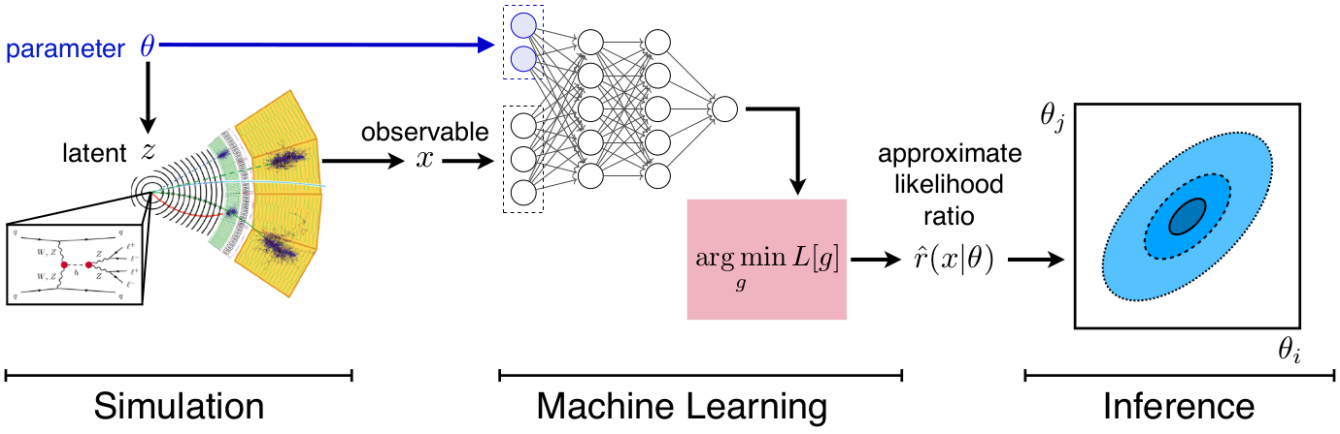
Unfortunately, filling high-dimensional histograms is not tractable.



Who you gonna call? Machine learning!

Refs: Bolognesi et al, 2012 (arXiv:1208.4018)

CARL



Key insights

- The likelihood ratio is sufficient for inference.
- Evaluating the likelihood ratio does not require evaluating the individual likelihoods.
- Supervised learning indirectly estimates likelihood ratios.

Refs: Cranmer et al, 2016 (arXiv:1506.02169)

Supervised learning provides a way to automatically construct s:

- Let us consider a binary classifier $\hat{s}$ (e.g., a neural network) trained to distinguish $x \sim p(x|\theta_0)$ from $x \sim p(x|\theta_1)$.
- $\hat{s}$ is trained by minimizing the cross-entropy loss

$$L_{XE}[\hat{s}] = -\mathbb{E}_{p(x|\theta)\pi(\theta)}[1(\theta = \theta_0) \log \hat{s}(x) + 1(\theta = \theta_1) \log(1 - \hat{s}(x))]$$

The solution $\hat{s}$ found after training approximates the optimal classifier

$$\hat{s}(x) \approx s^*(x) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}.$$

Therefore,

$$r(x|\theta_0, \theta_1) \approx \hat{r}(x|\theta_0, \theta_1) = \frac{1 - \hat{s}(x)}{\hat{s}(x)}$$

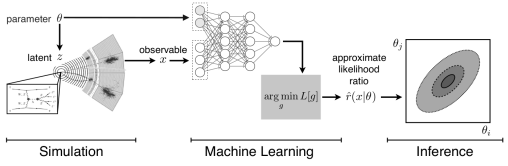
That is, supervised classification is equivalent to likelihood ratio estimation and can therefore be used for MLE inference.

Adversarial Variational Optimization

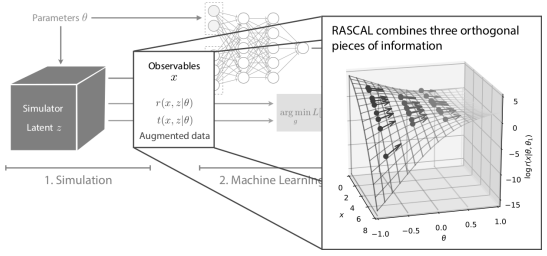
Treat the simulator as a black box

Make use of the inner structure

Learn a proxy for inference

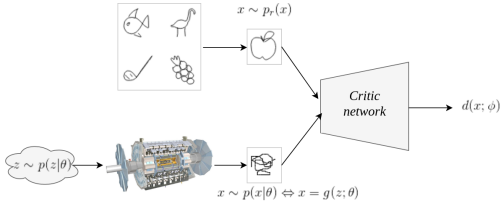


Histograms of observables
Neural density (ratio) estimation

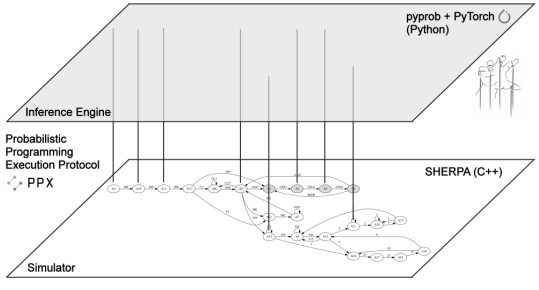


Mining gold from implicit models

Learn to control the simulator



Adversarial variational optimization



Probabilistic programming

Adversarial Variational Optimization of Non-Differentiable Simulators

Gilles Louppe
University of Liège

Joeri Hermans
University of Liège

Kyle Cranmer
New York University

Abstract

Complex computer simulators are increasingly used across fields of science as generative models tying parameters of an underlying theory to experimental observations. Inference in this setup is often difficult, as simulators rarely admit a tractable density or likelihood function. We introduce Adversarial Variational Optimization (AVO), a likelihood-free inference algorithm for fitting a non-differentiable generative model incorporating ideas from generative adversarial networks, variational optimization and empirical Bayes. We adapt the training procedure of generative adversarial networks by replacing the differentiable generative network with a domain-specific simulator. We solve the resulting non-differentiable minimax problem by minimizing variational upper bounds of the two adversarial objectives. Effectively, the procedure results in learning a proposal distribution over simulator parameters, such that the JS divergence between the marginal distribution of the synthetic data and the empirical distribution of observed data is minimized. We evaluate and compare the method with simulators producing both discrete and continuous data.

problem has motivated an active research effort in so-called *likelihood-free inference* algorithms such as Approximate Bayesian Computation (ABC) and density estimation-by-comparison algorithms (Beaumont et al., 2002; Marjoram et al., 2003; Sisson et al., 2007; Sisson and Fan, 2011; Marin et al., 2012; Cranmer et al., 2015).

In parallel, with the introduction of variational autoencoders (Kingma and Welling, 2013) and generative adversarial networks (Goodfellow et al., 2014), there has been a vibrant research program around implicit generative models based on neural networks (Mohamed and Lakshminarayanan, 2016). While some of these models also do not admit a tractable density, they are all differentiable by construction. In addition, generative models based on neural networks are highly parameterized and the model parameters have no obvious interpretation. In contrast, scientific simulators can be thought of as highly regularized generative models as they typically have relatively few parameters and they are endowed with some level of interpretation. In this setting, inference on the model parameters $\boldsymbol{\theta}$ is often of more interest than the latent variables $\mathbf{z}$.

In this work, we introduce Adversarial Variational Optimization (AVO), a likelihood-free inference algorithm for non-differentiable, implicit generative models. We adapt the adversarial training procedure of generative adversarial networks by replacing the implicit generative network with a domain-based scientific simula-

Generative adversarial networks

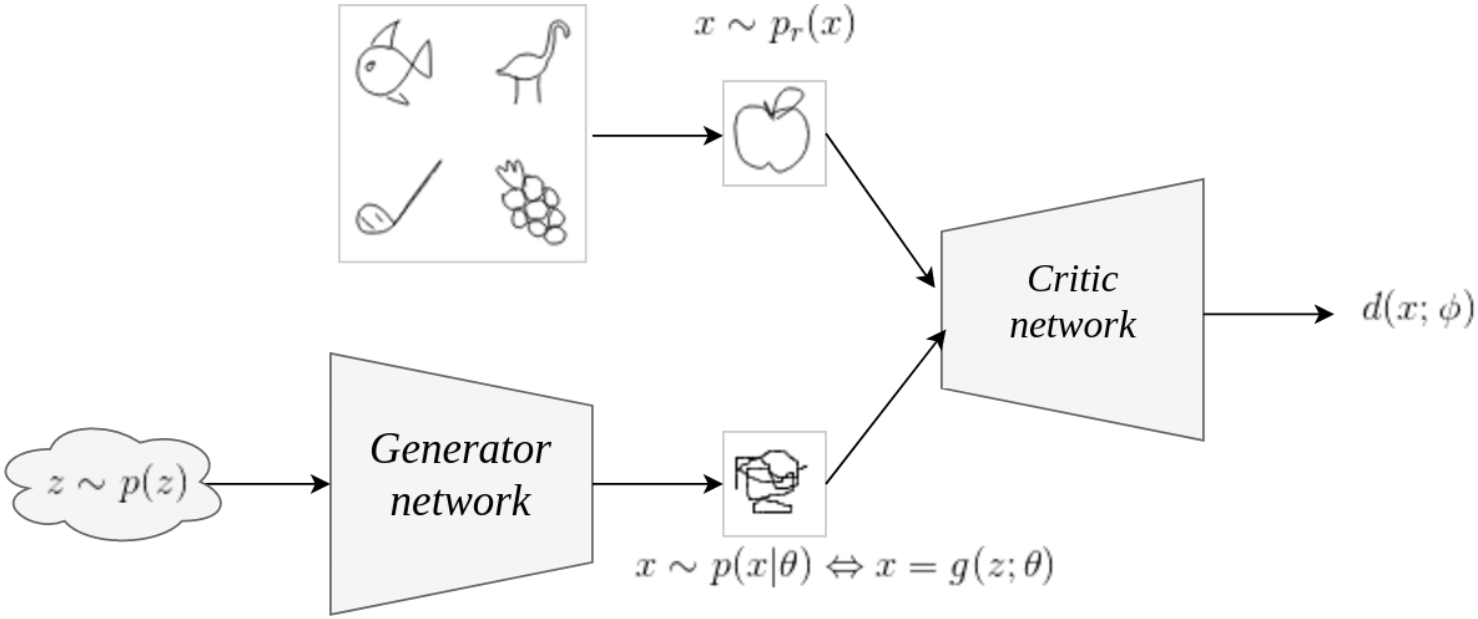
Express the task of learning a generative model as a two-player zero-sum game between two networks:

- The first network is a generator $g(\cdot;\theta) : \mathcal{Z} \to \mathcal{X}$, mapping a latent space equipped with a prior distribution $p(\mathbf{z})$ to the data space, thereby inducing a distribution

$$\mathbf{x} \sim p(\mathbf{x};\theta) \Leftrightarrow \mathbf{z} \sim p(\mathbf{z}), \mathbf{x} = g(\mathbf{z};\theta).$$

- The second network $d(\cdot;\phi) : \mathcal{X} \to [0,1]$ is a classifier trained to distinguish between true samples $\mathbf{x} \sim p_r(\mathbf{x})$ and generated samples $\mathbf{x} \sim p(\mathbf{x};\theta)$.

The central mechanism will be to use supervised learning to guide the learning of the generative model.



$$\arg\min_{\theta}\max_{\phi}\mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\log d(\mathbf{x};\phi)\right]+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log(1-d(g(\mathbf{z};\theta);\phi))\right]$$

In practice, the min-max solution is approximated using alternating stochastic gradient descent on the losses

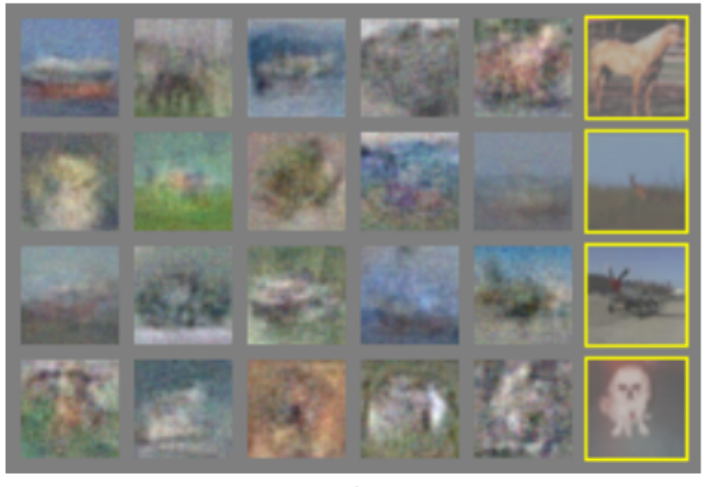
$$\mathcal{L}_d(\phi) = \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[-\log(d(\mathbf{x};\phi))\right] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[-\log(1-d(g(\mathbf{z};\theta);\phi))\right]$$
$$\mathcal{L}_g(\theta) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log(1-d(g(\mathbf{z};\theta);\phi))\right],$$

for which unbiased gradient estimates can be computed with Monte Carlo integration.

- For one step on θ, we can optionally take k steps on ϕ, since we need the classifier to remain near optimal.
- Note that to compute $\nabla_\theta \mathcal{L}_g$, it is necessary to backprop all the way through d before computing the partial derivatives with respect to g's internals.

a)

b)



c)

d)

Goodfellow et al, 2014.

Odena et al
2016

Miyato et al
2017

Zhang et al
2018

Brock et al
2018

Figure 2. Uncurated set of images produced by our style-based generator (config F) with the FFHQ dataset. Here we used a variation of the truncation trick [5, 29] with $\psi = 0.7$ for resolutions $4^2 - 32^2$. Please see the accompanying video for more results.

Karras et al, 2018.

Game analysis

Consider a generator g fixed at θ. Given a set of observations

$$\mathbf{x}_i \sim p_r(\mathbf{x}), i = 1, ..., N,$$

we can generate a two-class dataset

$$\mathbf{d} = \{(\mathbf{x}_1, 1), ..., (\mathbf{x}_N, 1), (g(\mathbf{z}_1; \theta), 0), ..., (g(\mathbf{z}_N; \theta), 0))\}.$$

The best classifier d is obtained by minimizing the cross-entropy

$$\begin{aligned}\mathcal{L}_d(\phi) &= -\frac{1}{2N}\left(\sum_{i=1}^{N}\left[\log d(\mathbf{x}_i; \phi)\right] + \sum_{i=1}^{N}\left[\log(1 - d(g(\mathbf{z}_i; \theta); \phi))\right]\right) \\ &\approx -\frac{1}{2}\left(\mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\log d(\mathbf{x}; \phi)\right] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log(1 - d(g(\mathbf{z}; \theta); \phi))\right]\right)\end{aligned}$$

with respect to ϕ.

Let us define the value function

$$V(\phi, \theta) = \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})} \left[\log d(\mathbf{x}; \phi) \right] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \left[\log(1 - d(g(\mathbf{z}; \theta); \phi)) \right].$$

We have,

- $V(\phi, \theta)$ is high if d is good at recognizing true from generated samples.
- If d is the best classifier given g, and if V is high, then this implies that the generator is bad at reproducing the data distribution.
- Conversely, g will be a good generative model if V is low when d is a perfect opponent.

Therefore, the ultimate goal is

$$\theta^* = \arg \min_{\theta} \max_{\phi} V(\phi, \theta).$$

For a generator g fixed at θ, the classifier d with parameters ϕ^*_θ is optimal if and only if

$$\forall \mathbf{x}, d(\mathbf{x}; \phi^*_\theta) = \frac{p_r(\mathbf{x})}{p(\mathbf{x}; \theta) + p_r(\mathbf{x})}.$$

Therefore,

$$\begin{aligned}
&\min_\theta \max_\phi V(\phi, \theta) = \min_\theta V(\phi^*_\theta, \theta) \\
&= \min_\theta \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})} \left[\log \frac{p_r(\mathbf{x})}{p(\mathbf{x}; \theta) + p_r(\mathbf{x})} \right] + \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}; \theta)} \left[\log \frac{p(\mathbf{x}; \theta)}{p(\mathbf{x}; \theta) + p_r(\mathbf{x})} \right] \\
&= \min_\theta \mathrm{KL} \left(p_r(\mathbf{x}) || \frac{p_r(\mathbf{x}) + p(\mathbf{x}; \theta)}{2} \right) \\
&\qquad + \mathrm{KL} \left(p(\mathbf{x}; \theta) || \frac{p_r(\mathbf{x}) + p(\mathbf{x}; \theta)}{2} \right) - \log 4 \\
&= \min_\theta 2 \, \mathrm{JSD}(p_r(\mathbf{x}) || p(\mathbf{x}; \theta)) - \log 4
\end{aligned}$$

where JSD is the Jensen-Shannon divergence.

In summary, solving the min-max problem

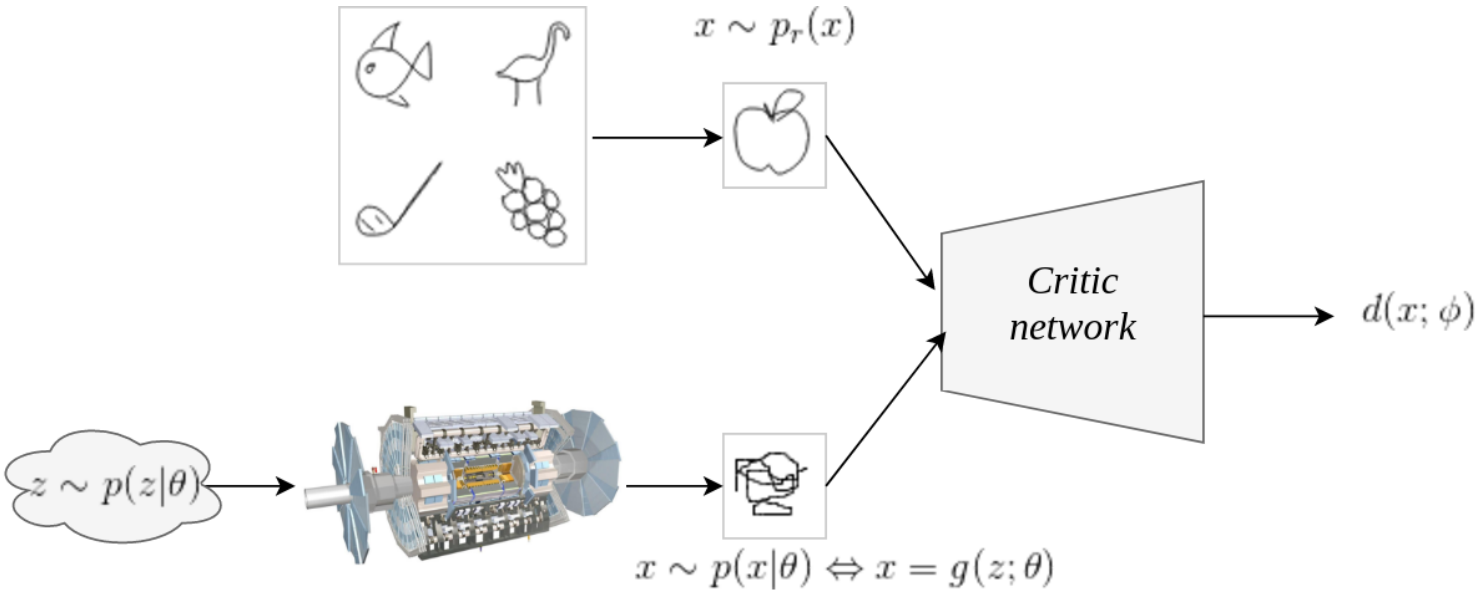
$$\theta^* = \arg\min_{\theta} \max_{\phi} V(\phi, \theta)$$

is equivalent to

$$\theta^* = \arg\min_{\theta} \mathrm{JSD}(p_r(\mathbf{x})||p(\mathbf{x};\theta)).$$

Since $\mathrm{JSD}(p_r(\mathbf{x})||p(\mathbf{x};\theta))$ is minimum if and only if $p_r(\mathbf{x}) = p(\mathbf{x};\theta)$, this proves that the min-max solution corresponds to a generative model that perfectly reproduces the true data distribution.

AVO



Replace g with an actual scientific simulator!

Refs: Louppe et al, 2017 (arXiv:1707.07113)

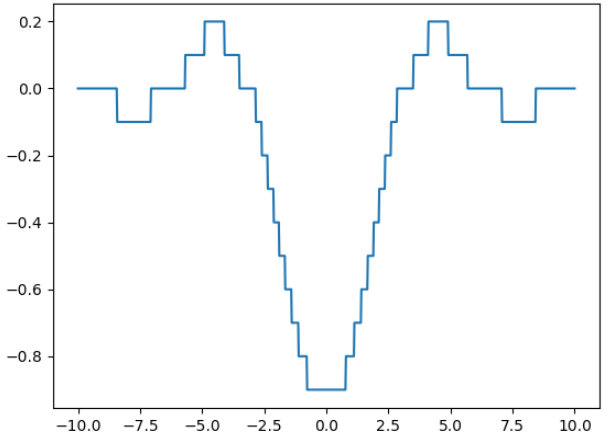
Key insights

- Replace the generative network with a non-differentiable forward simulator $g(\mathbf{z}; \theta)$.
- Let the neural network critic figure out how to adjust the simulator parameters.
- Bypass the non-differentiability with variational optimization.

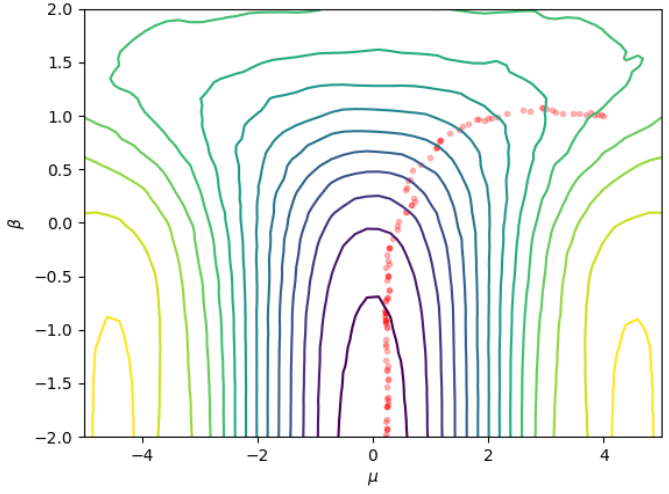
Variational optimization

$$\min_\theta f(\theta) \leq \mathbb{E}_{\theta \sim q(\theta|\psi)}[f(\theta)] = U(\psi)$$

$$\nabla_\psi U(\psi) = \mathbb{E}_{\theta \sim q(\theta|\psi)}[f(\theta) \nabla_\psi \log q(\theta|\psi)]$$



Piecewise constant $-\frac{\sin(x)}{x}$



$q(\theta|\psi = (\mu, \beta)) = \mathcal{N}(\mu, e^\beta)$

(Similar to REINFORCE gradient estimates)

Variational optimization can be used to bypass the non-differentiability of the adversarial game by optimizing upper bounds of the adversarial objectives

$$U_d(\phi) = \mathbb{E}_{\theta \sim q(\theta;\psi)} \left[\mathcal{L}_d(\phi) \right]$$
$$U_g(\psi) = \mathbb{E}_{\theta \sim q(\theta;\psi)} \left[\mathcal{L}_g(\theta) \right]$$

respectively over ϕ and ψ.

Algorithm 1 Adversarial variational optimization (AVO).

Inputs: Observed data $\{\mathbf{x}_i \sim p_r(\mathbf{x})\}_{i=1}^N$, simulator g.
Outputs: Proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\psi})$, such that $q(\mathbf{x}|\boldsymbol{\psi}) \approx p_r(\mathbf{x})$.
Hyper-parameters: The number k of training iterations of the discriminator d (default: $k = 1$),
The size M of a mini-batch (default: $M = 32$),
The R_1 regularization coefficient λ (default: $\lambda = 10$),
The entropy penalty coefficient γ (default: $\gamma = 0$).
The baseline strategy b in REINFORCE estimates (default: Eqn. 13).

$q(\boldsymbol{\theta}|\boldsymbol{\psi}) \leftarrow$ prior on $\boldsymbol{\theta}$ (with differentiable and known density)
while $\boldsymbol{\psi}$ has not converged **do**
  for $i = 1$ to k **do** ▷ Update d
    Sample true data $\{\mathbf{x}_m \sim p_r(\mathbf{x}), y_m = 1\}_{m=1}^{M/2}$.
    Sample synthetic data $\{\boldsymbol{\theta}_m \sim q(\boldsymbol{\theta}|\boldsymbol{\psi}), \mathbf{z}_m \sim p(\mathbf{z}|\boldsymbol{\theta}_m), \tilde{\mathbf{x}}_m = g(\mathbf{z}_m; \boldsymbol{\theta}_m), y_m = 0\}_{m=M/2+1}^{M}$.
    $\nabla_{\boldsymbol{\phi}} U_d \leftarrow \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{\phi}} \left[-y_m \log(d(\mathbf{x}_m; \boldsymbol{\phi})) - (1 - y_m) \log(1 - d(\mathbf{x}_m; \boldsymbol{\phi}))\right]$
    $\nabla_{\boldsymbol{\phi}} R_1 \leftarrow \frac{1}{M/2} \sum_{m=1}^{M/2} \nabla_{\boldsymbol{\phi}} \left[||\nabla_{\boldsymbol{\phi}} d(\mathbf{x}_m; \boldsymbol{\phi})||^2\right]$
    $\boldsymbol{\phi} \leftarrow \textsc{RmsProp}(\nabla_{\boldsymbol{\phi}} U_d + \lambda \nabla_{\boldsymbol{\phi}} R_1)$
  end for
  Sample synthetic data $\{\boldsymbol{\theta}_m \sim q(\boldsymbol{\theta}|\boldsymbol{\psi}), \mathbf{z}_m \sim p(\mathbf{z}|\boldsymbol{\theta}_m), \tilde{\mathbf{x}}_m = g(\mathbf{z}_m; \boldsymbol{\theta}_m)\}_{m=1}^{M}$. ▷ Update $q(\boldsymbol{\theta}|\boldsymbol{\psi})$
  $\nabla_{\boldsymbol{\psi}} U_g \leftarrow \frac{1}{M} \sum_{m=1}^{M} \left[\nabla_{\boldsymbol{\psi}} \log q(\boldsymbol{\theta}_m|\boldsymbol{\psi})(\log(1 - d(\tilde{\mathbf{x}}_m; \boldsymbol{\phi})) - b)\right]$
  $\nabla_{\boldsymbol{\psi}} H \leftarrow \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{\psi}} \left[-q(\boldsymbol{\theta}_m|\boldsymbol{\psi}) \log q(\boldsymbol{\theta}_m|\boldsymbol{\psi})\right]$
  $\boldsymbol{\psi} \leftarrow \textsc{RmsProp}(\nabla_{\boldsymbol{\psi}} U_g + \gamma \nabla_{\boldsymbol{\psi}} H)$
end while

Operationally,

$$\mathbf{x} \sim q(\mathbf{x}|\psi) \Leftrightarrow \theta \sim q(\theta|\psi), \mathbf{z} \sim p(\mathbf{z}|\theta), \mathbf{x} = g(\mathbf{z}; \theta)$$

Therefore, $q(\mathbf{x}|\psi)$ is the marginal $\int p(\mathbf{x}|\theta)q(\theta|\psi)d\theta$ and the AVO procedure results in solving

$$\psi^* = \arg\min_{\psi} \mathrm{JSD}(p_r(\mathbf{x})||q(\mathbf{x}|\psi)).$$

- If the simulator $p(\mathbf{x}|\theta)$ is misspecified, $q(\mathbf{x}|\psi)$ will to attempt to smear the simulator to approach $p_r(\mathbf{x})$.
- If not, $q(\theta|\psi)$ will concentrate its mass around the true data-generating parameters.

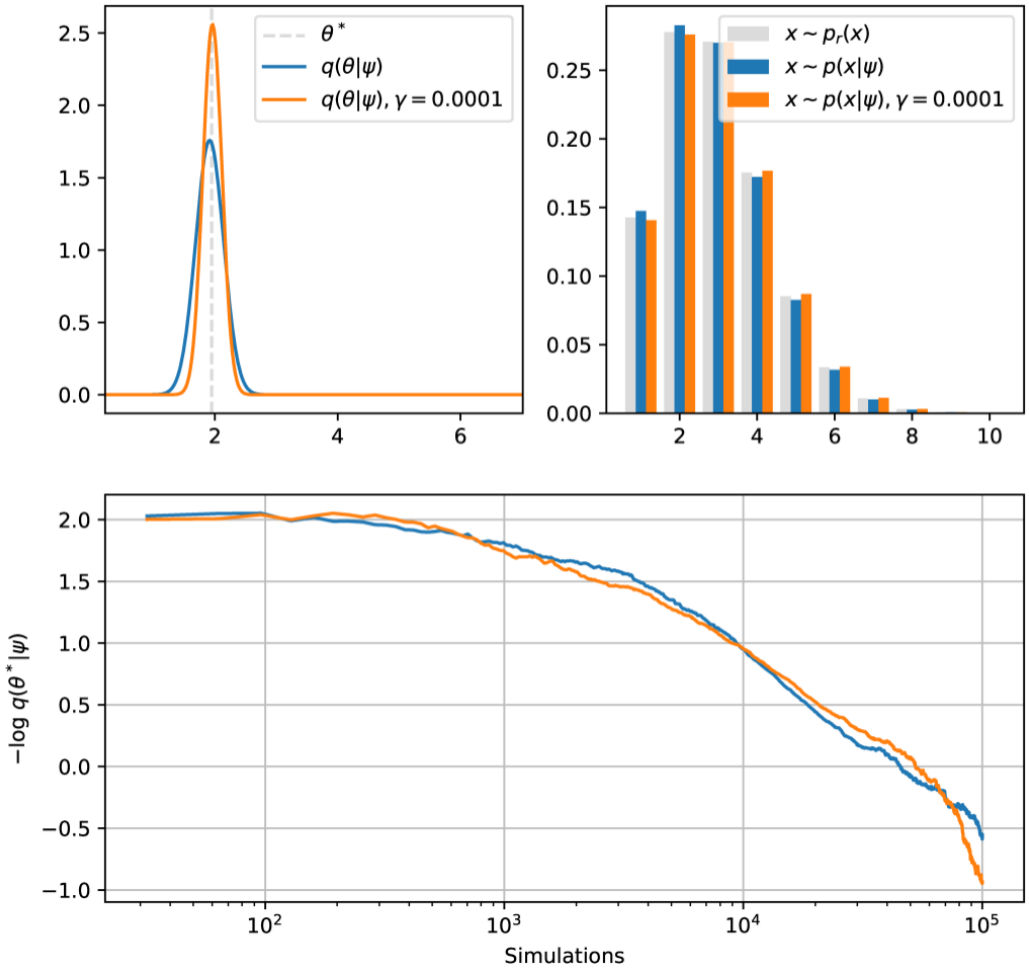
Weaknesses

AVO can been seen through the lens of empirical Bayes, where the data are used to optimize a prior within the family $q(\theta|\psi)$.

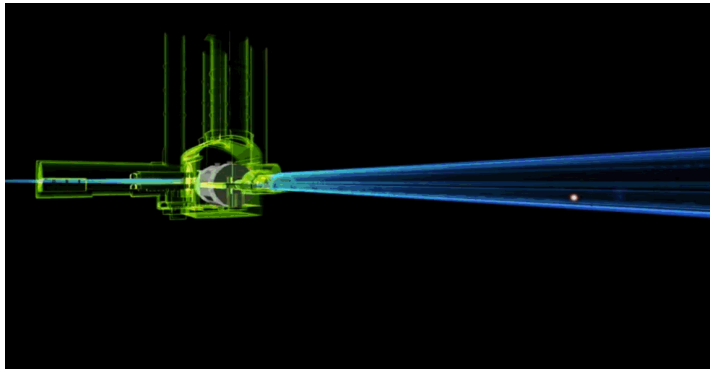
- Therefore, if the simulator is well specified, AVO can only be used to determine a point estimate of the MLE. It does not enable likelihood-free posterior inference.
- The solution may change from one run to the next in case of local minima.

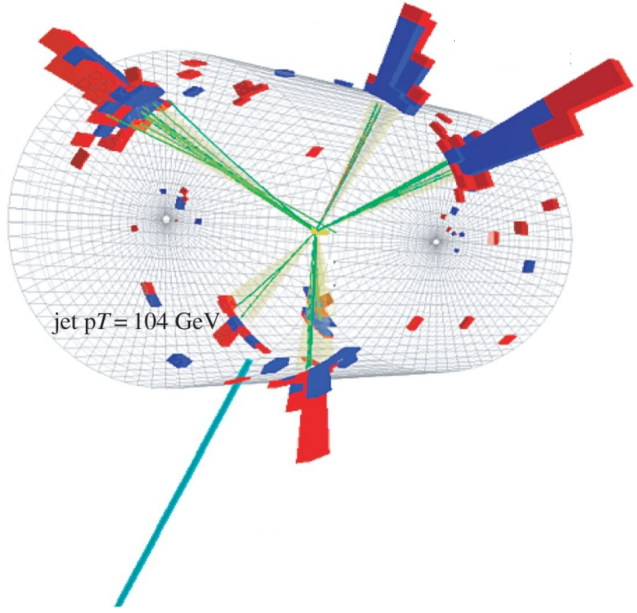
Experiments

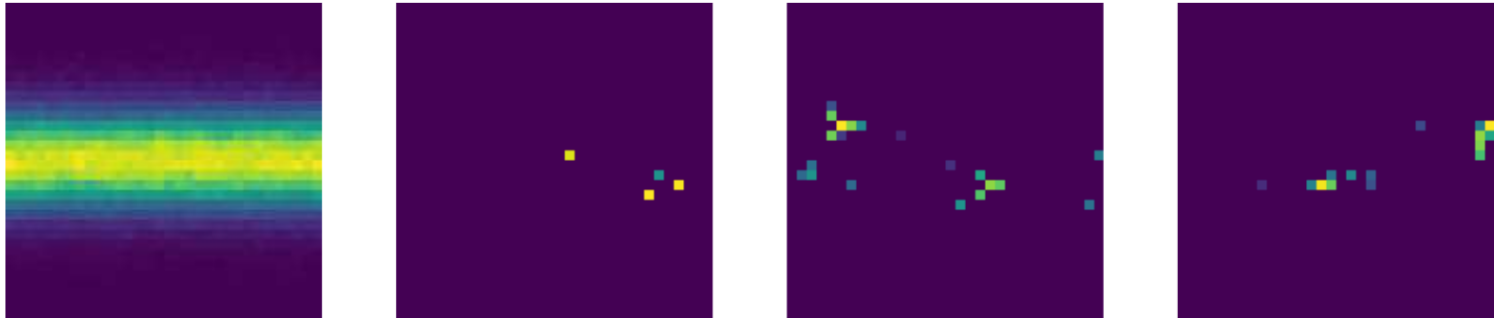
Fitting a discrete Poisson distribution



High-energy particle collisions

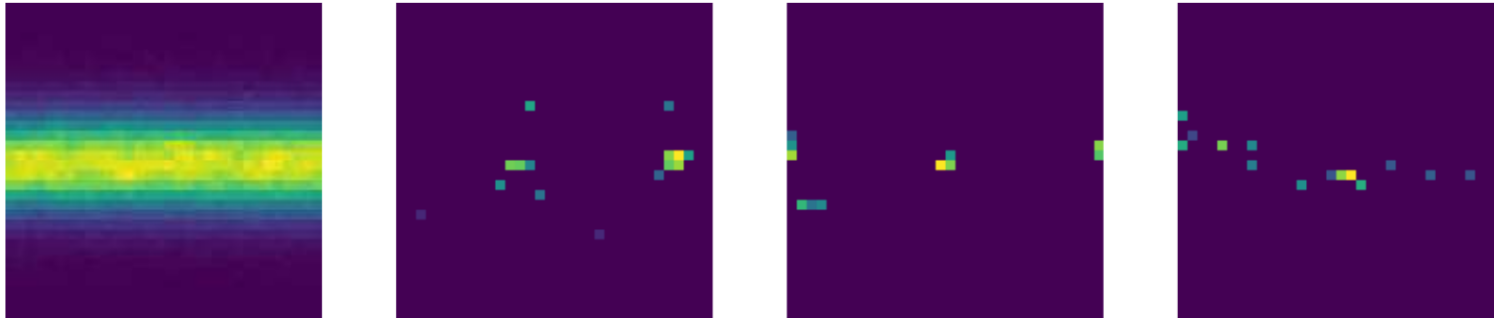




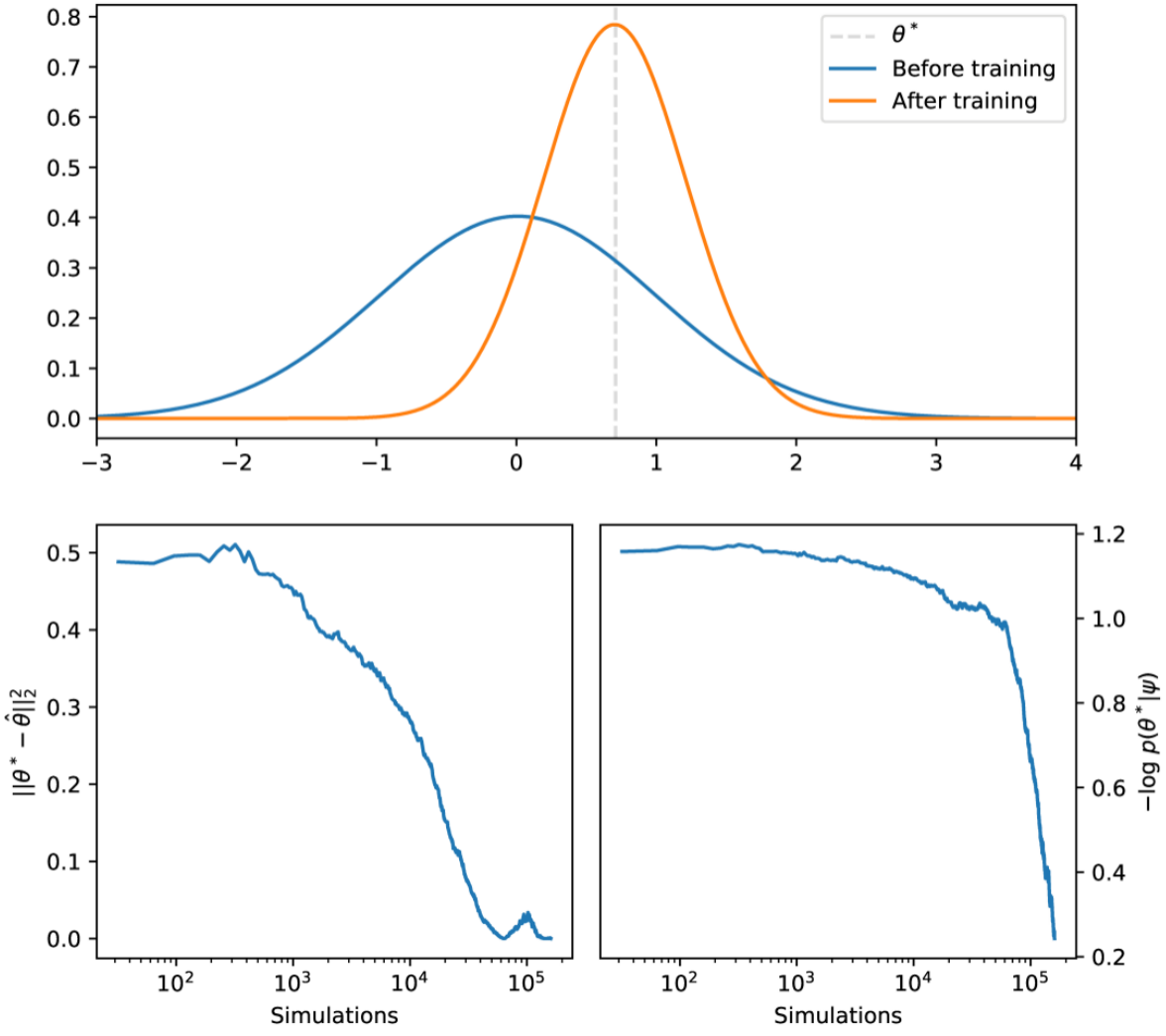


$z = 0$

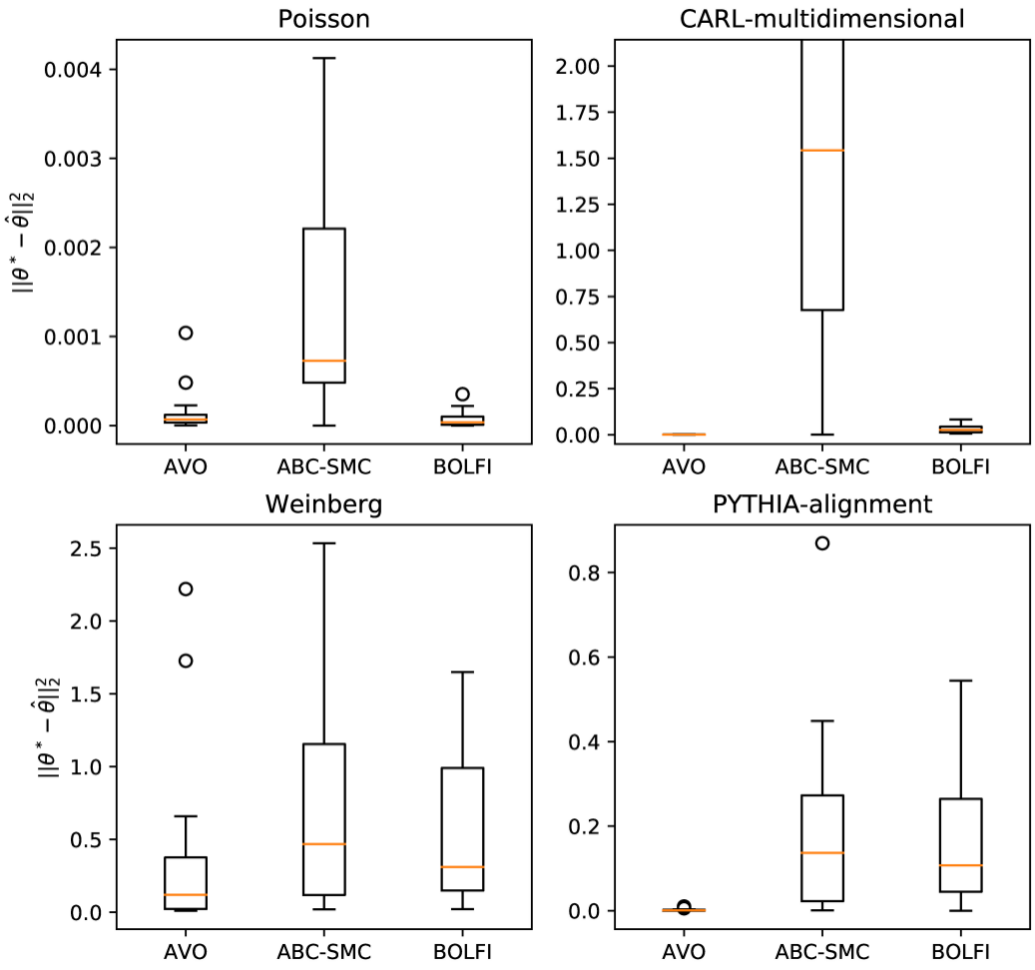
vs



$z = 1$


θ^*
Before training
After training
$||\theta^* - \hat{\theta}||_2^2$
$-\log p(\theta^*|\psi)$
Simulations
Simulations

Benchmarks


Poisson
CARL-multidimensional
Weinberg
PYTHIA-alignment
$||\theta^* - \hat{\theta}||_2^2$
0.004
0.003
0.002
0.001
0.000
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
2.5
2.0
1.5
1.0
0.5
0.0
0.8
0.6
0.4
0.2
0.0
AVO
ABC-SMC
BOLFI

Summary

In this work, we develop a likelihood-free inference algorithm for non-differentiable, implicit generative models. The algorithm combines adversarial training with variational optimization to minimize variational upper bounds on the otherwise non-differentiable adversarial objectives. The AVO algorithm enables empirical Bayes through variational inference in the likelihood-free setting. This approach does not incur the inefficiencies of an ABC-like rejection sampler nor the disadvantages of likelihood-free inference algorithms that rely on ad hoc summary statistics. When the model is well-specified, the AVO algorithm provides point estimates for the generative model, which asymptotically corresponds to the data generating parameters. Experimental results highlight the good performance of AVO in comparison to the well-established ABC-SMC and BOLFI algorithms.

The end.