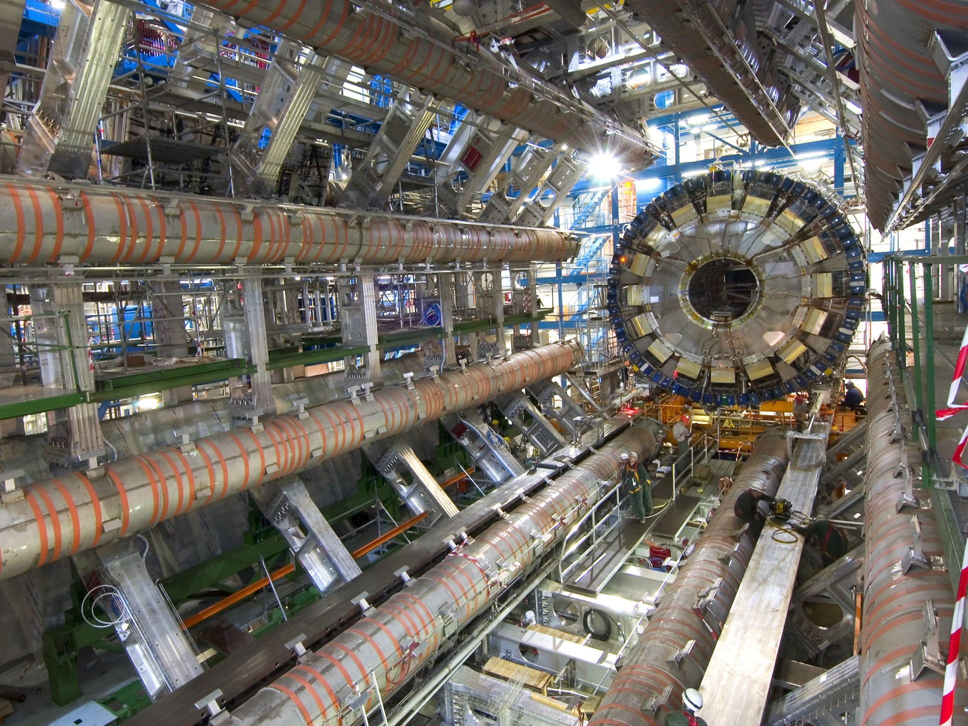
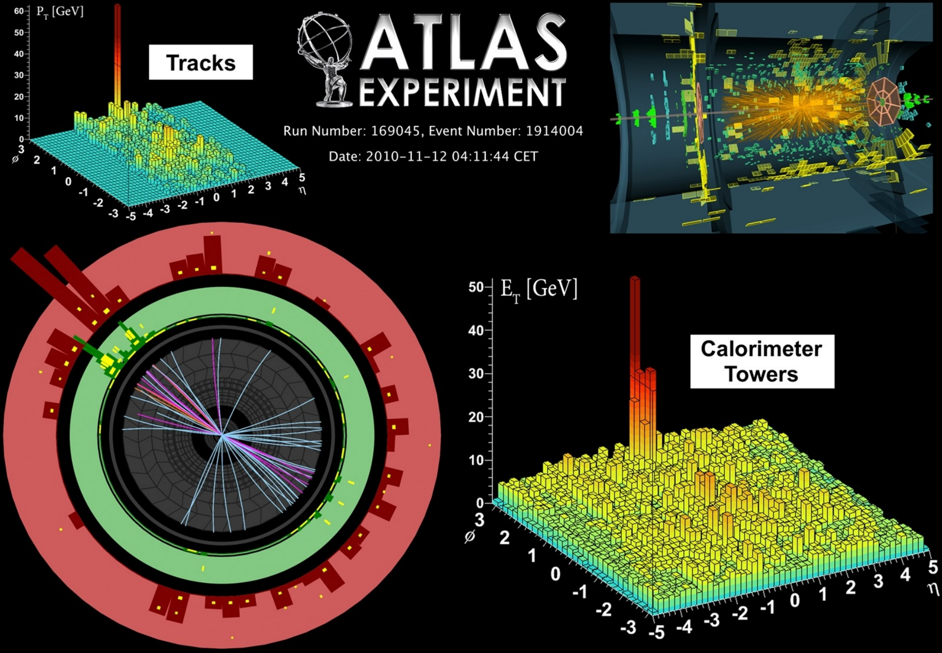
Adversarial Games for Particle Physics

Gilles Louppe

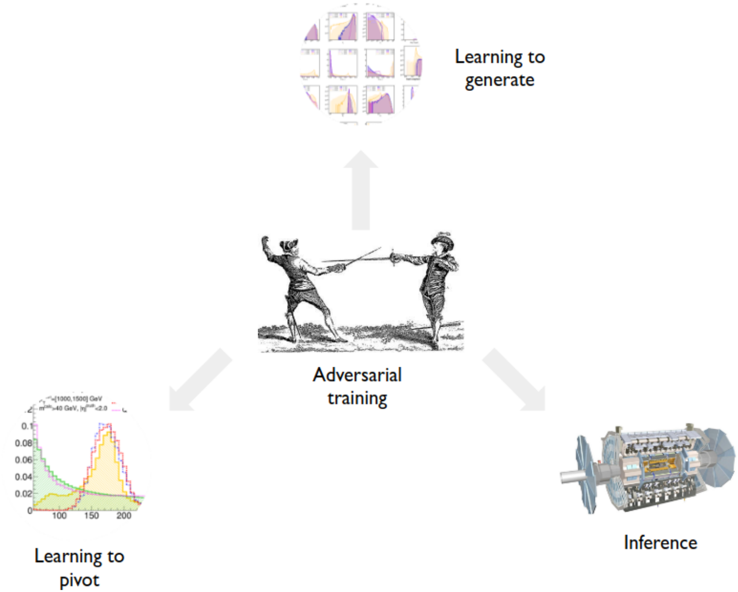
Deep Learning for Physical Sciences workshop
December 8, 2017



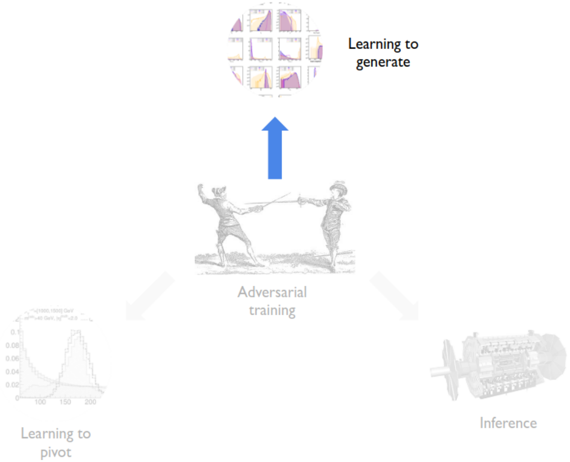


ATLAS
EXPERIMENT
Run Number: 169045, Event Number: 1914004
Date: 2010-11-12 04:11:44 CET
P_T [GeV]
Tracks
ϕ
η
E_T [GeV]
Calorimeter
Towers

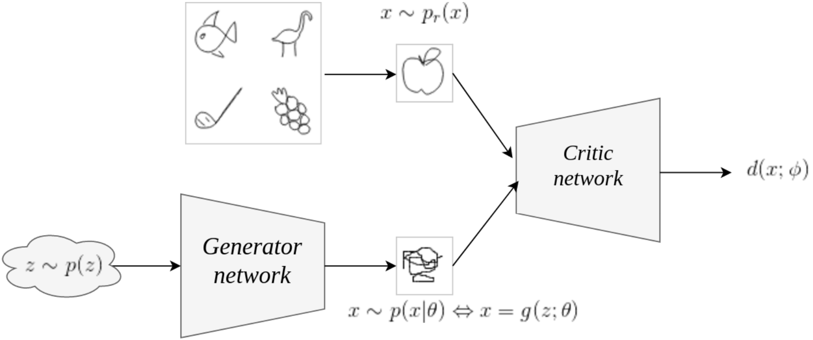
Adversarial games for particle physics



I. Fast simulation



Generative adversarial networks

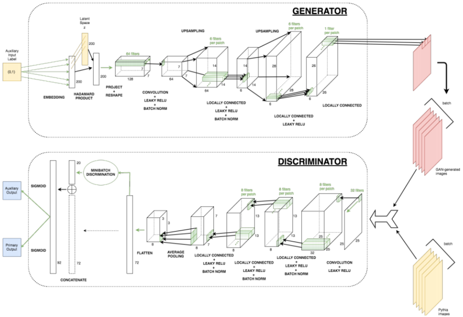


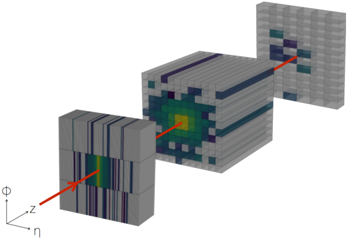
$$\mathcal{L}_d = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\boldsymbol{\theta})}[d(\mathbf{x};\phi)] - \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[d(\mathbf{x};\phi)] + \lambda\Omega(\phi)$$

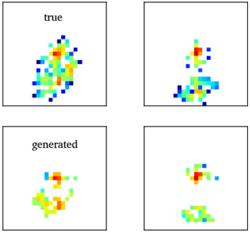
$$\mathcal{L}_g = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\boldsymbol{\theta})}[d(\mathbf{x};\phi)]$$

(Wasserstein GAN + Gradient Penalty)

Fast simulation



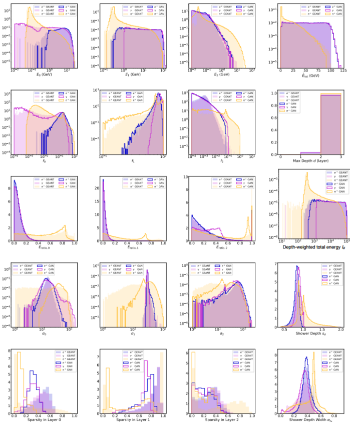




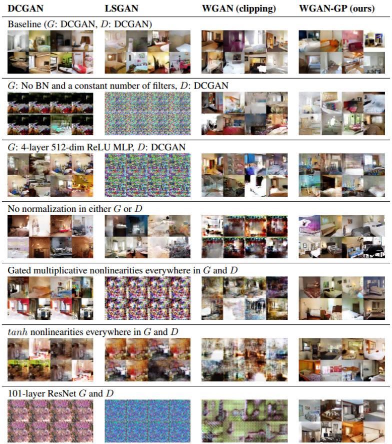
Challenges:

- How to ensure physical properties?
- Non-uniform geometry
- Mostly sparse
- How to scale to full resolution?

Evaluation



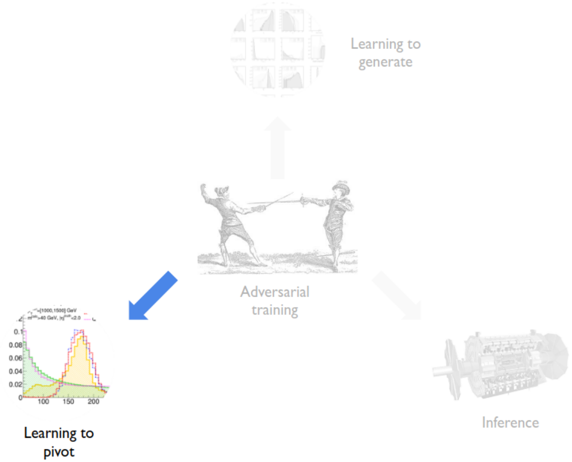
Physics: Evaluate well-known physical variates



ML: Look at generated images

How to be sure the generator is physically correct?

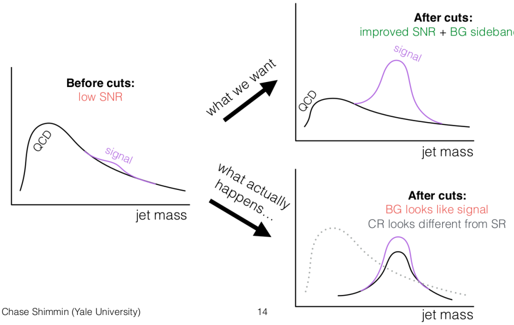
II. Learning to Pivot

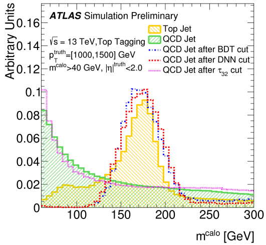


Shimmin et al, 2017, arXiv:1703.03507; ATL-PHYS-PUB-2017-004

Independence from physics variates

- Analysis often rely on the model being independent from some physics variates (e.g., mass).
- Correlation leads to systematic uncertainties, that cannot easily be characterized and controlled.

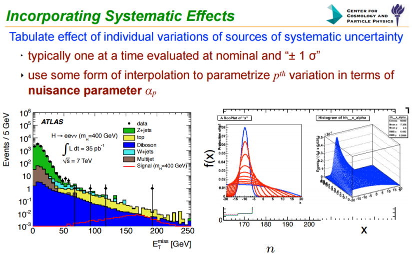




Independence from known unknowns

Credits: Kyle Cranmer

- The generation process is often not uniquely specified or known exactly, hence systematic uncertainties.
- Parametrization through nuisance parameters.
- Ideally, we would like a classifier that is robust to nuisance parameters.



$$\mathbf{f}(\mathcal{D}|\boldsymbol{\alpha}) = \mathrm{Pois}(n|\nu(\boldsymbol{\alpha})) \prod_{e=1}^{n} f(x_e|\boldsymbol{\alpha})$$

Problem statement

- Assume a family of data generation processes $p(X, Y, Z)$ where
  - X are the data (taking values $x \in \mathcal{X}$),
  - Y are the target labels (taking values $y \in \mathcal{Y}$),
  - Z is an auxiliary random variable (taking values $z \in \mathcal{Z}$).
    - Z corresponds to physics variates or nuisance parameters.
- Supervised learning: learn a function $f(\cdot; \theta_f) : \mathcal{X} \mapsto \mathcal{Y}$.
- We want inference based on $f(X; \theta_f)$ to be robust to the value $z \in \mathcal{Z}$.
  - E.g., we want a classifier that does not change with systematic variations, even though the data might.

Pivot

- We define robustness as requiring the distribution of $f(X;\theta_f)$ conditional on Z to be invariant with Z. That is, such that

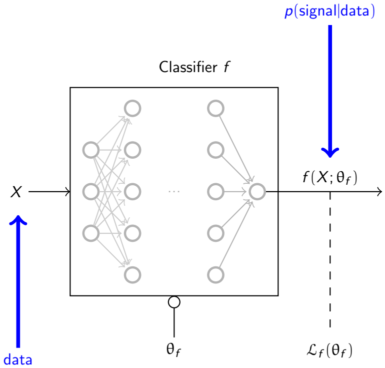
$$p(f(X;\theta_f) = s|z) = p(f(X;\theta_f) = s|z')$$

for all $z, z' \in \mathcal{Z}$ and all values $s \in \mathcal{S}$ of $f(X;\theta_f)$.

- If f satisfies this criterion, then f is known as a pivotal quantity.

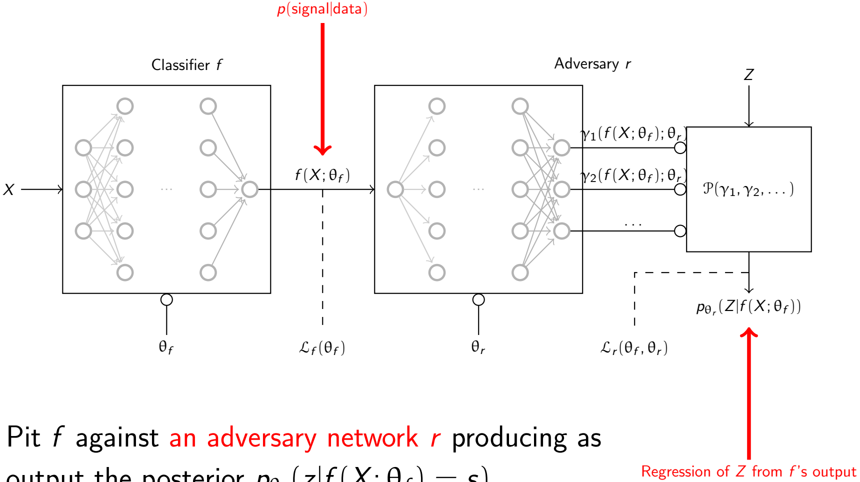
- Same as requiring $f(X;\theta_f)$ and Z to be independent random variables.

Adversarial game



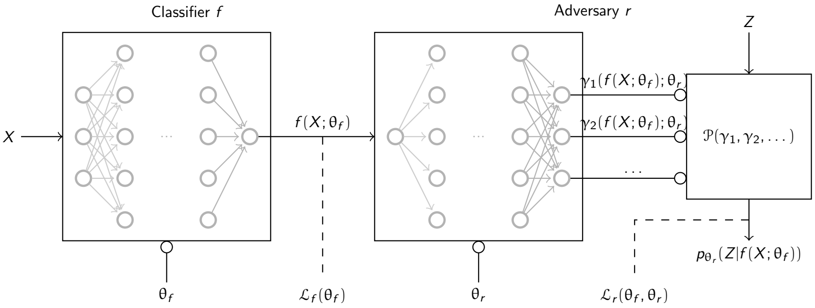
Consider a classifier f built as usual, minimizing the cross-entropy $\mathcal{L}_f(\theta_f) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim Y|x}[-\log p_{\theta_f}(y|x)]$.

Adversarial game



Pit f against an adversary network r producing as output the posterior $p_{\theta_r}(z|f(X;\theta_f) = s)$.
Set r to minimize the cross entropy
$\mathcal{L}_r(\theta_f, \theta_r) = \mathbb{E}_{s\sim f(X;\theta_f)}\mathbb{E}_{z\sim Z|s}[-\log p_{\theta_r}(z|s)]$.

Adversarial game



Goal is to solve: $\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f} \max_{\theta_r} \mathcal{L}_f(\theta_f) - \mathcal{L}_r(\theta_f, \theta_r)$

Intuitively, r penalizes f for outputs that can be used to infer Z.

In practice

- The assumption of existence of a classifier that is both optimal and pivotal may not hold.

- However, the minimax objective can be rewritten as

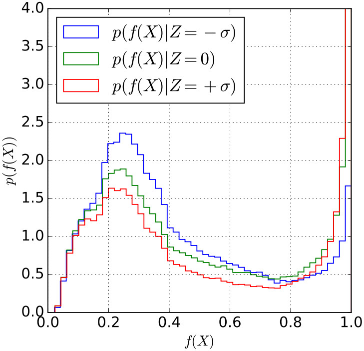
$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r)$$

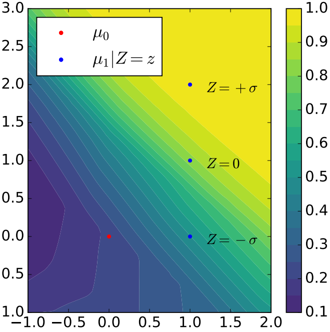
  where λ controls the trade-off between the performance of f and its independence w.r.t. Z.

  - Setting λ to a large value enforces f to be pivotal.
  - Setting λ close to 0 constraints f to be optimal.

- Tuning λ is guided by a higher-level objective (e.g., statistical significance).

Toy example (without adversarial training)

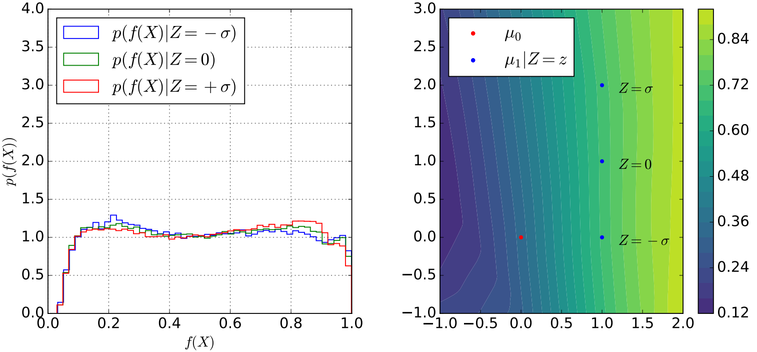




(Left) The conditional probability distributions of $f(X;\theta_f)|Z=z$ changes with z.

(Right) The decision surface strongly depends on X_2.

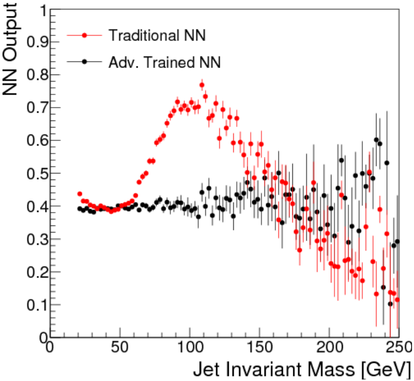
Toy example (with adversarial training)



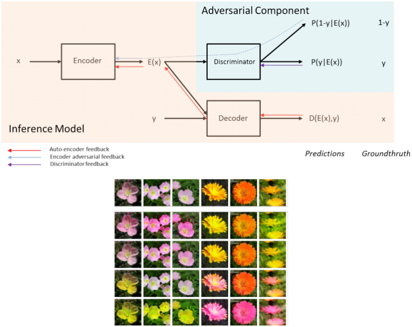
(Left) The conditional probability distributions of $f(X;\theta_f)|Z=z$ are now (almost) invariant with z!

(Right) The decision surface is now independent of X_2.

Applications



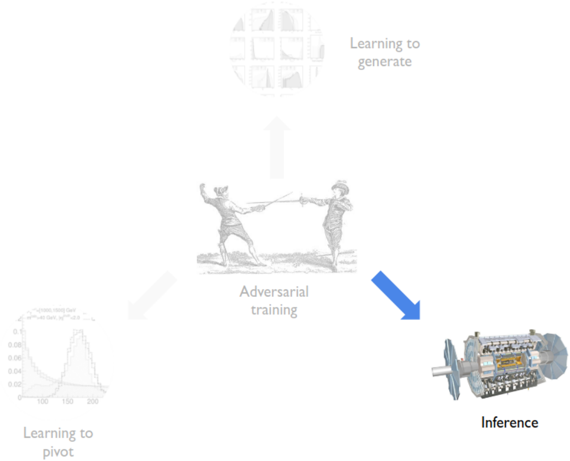
Decorrelated Jet Substructure Tagging using Adversarial Neural Networks



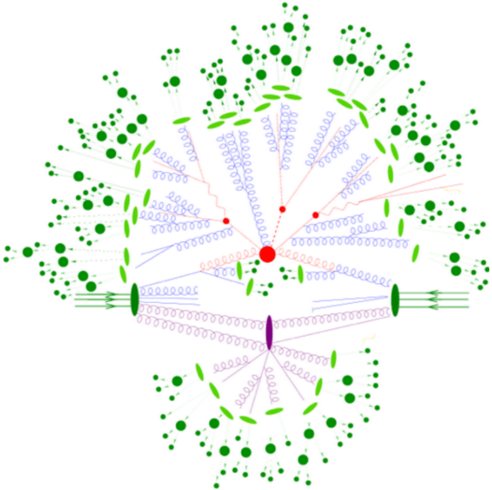
Fader networks

Louppe and Cranmer, 2017, arXiv:1707.07113

III. Adversarial Variational Optimization



Microscopic picture

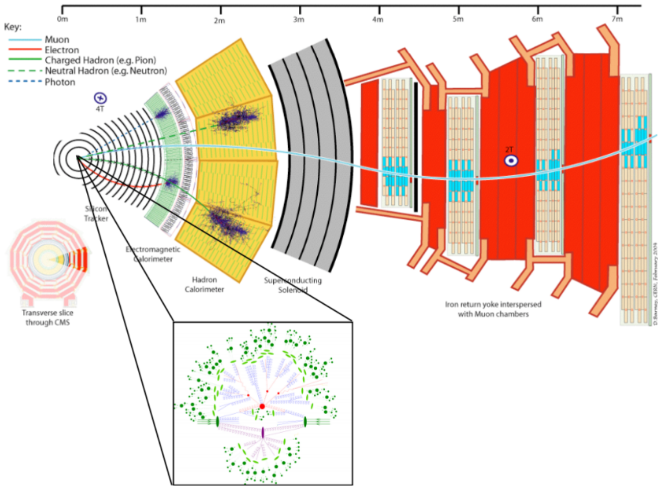


Pencil and paper calculable from first principles.

Controlled approximation of first principles.

Phenomenological model.

Macroscopic picture



Simulate interactions of outgoing particles with the detector.

Likelihood-free assumptions

Operationally,

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

where

- $\mathbf{z}$ provides a source of randomness;
- g is a non-differentiable deterministic function (e.g. a computer program).

Accordingly, the density $p(\mathbf{x}|\boldsymbol{\theta})$ can be written as

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int_{\{\mathbf{z}: g(\mathbf{z};\theta) = \mathbf{x}\}} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}$$

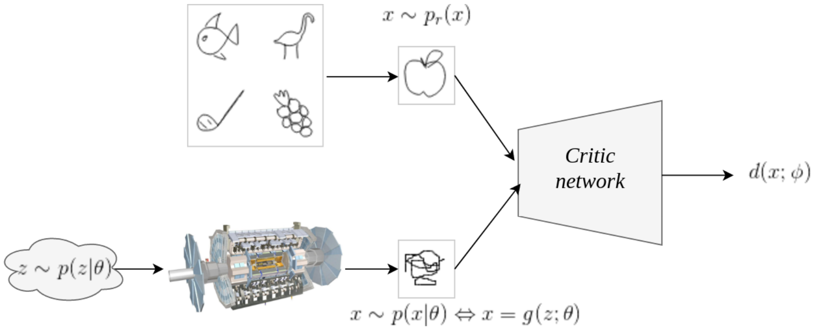
Evaluating the integral is often intractable.

Inference

Given observations $\mathbf{x} \sim p_r(\mathbf{x})$, we seek:

$$\boldsymbol{\theta}^* = \arg\min_{\theta} \rho(p_r(\mathbf{x}), p(\mathbf{x}|\boldsymbol{\theta}))$$

Adversarial game

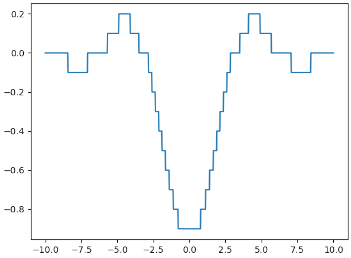


Replace g with an actual scientific simulator!

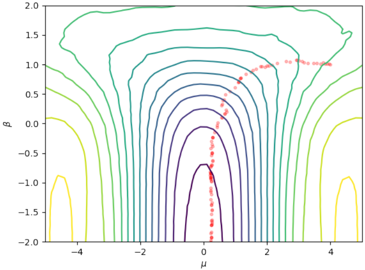
Variational Optimization

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

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



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



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

(Similar to REINFORCE gradient estimates)

Adversarial Variational Optimization

- Replace the generative network with a non-differentiable forward simulator $g(\mathbf{z}; \boldsymbol{\theta})$.
- With VO, optimize upper bounds of the adversarial objectives:

$$U_d = \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\psi})}[\mathcal{L}_d] \quad (1)$$

$$U_g = \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\psi})}[\mathcal{L}_g] \quad (2)$$

respectively over ϕ and ψ.

Operationally,

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

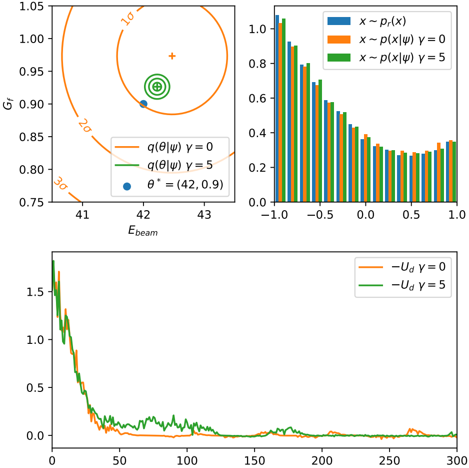
Therefore, $q(\mathbf{x}|\boldsymbol{\psi})$ is the marginal $\int p(\mathbf{x}|\boldsymbol{\theta})q(\boldsymbol{\theta}|\boldsymbol{\psi})d\boldsymbol{\theta}$.

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

Preliminary results

Simplified simulator for electron–positron collisions resulting in muon–antimuon pairs.

- Parameters: E_{beam}, G_f.
- Observations: $\mathbf{x} = \cos(A) \in [-1, 1]$, where A is the polar angle of the outgoing muon wrt incoming electron.



Ongoing work

- Benchmark against alternative methods (e.g., ABC).
- Scale to a full scientific simulator.
- Control variance of the gradient estimates.

Summary

- Adversarial training = indirectly specifying complicated loss functions.
  - For generation
  - For enforcing constraints
- Directly useful in domain sciences, such as particle physics.

Questions?

Joint work with