

An Adversarial ML Approach to Learning PDF Parameters

Katherine Keegan¹ Mohamed Wahib² Emil Vatai² Aleksandr Drozd² Johann Rudi³ Pi-Yueh Chuang³ Xingfu Wu⁴

¹Emory University

²RIKEN

³Virginia Tech

⁴Argonne National Lab

June 26, 2024



Introduction

Overall goal: fit parameters to PDFs based on observable event data.

- For this work, consider a simple 1-D deep inelastic scattering problem with PDFs given by

$$u(x; p) = N_u x^{a_u} (1 - x)^{b_u}$$

$$d(x; p) = N_d x^{a_d} (1 - x)^{b_d},$$

$$x \in (0, 1).$$

- Observable events (x, Q^2) are sampled from some particle momentum distribution based on this $u(x; p)$ and $d(x; p)$.
- Hope: finding parameters $p = [N_u \ a_u \ b_u \ N_d \ a_d \ b_d]^\top$ such that we can generate realistic fake events \implies we know the parameters.

Possibly very related work

At this workshop (non-exhaustive):

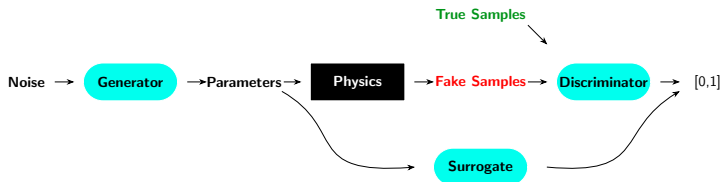
- Tim Hobbs, NN parametrizations of PDFs
- Chiara Bissolotti, PDF analysis with NNs
- Felix Ringer, diffusion models for generating events
- Yaohang Li, GANs and uncertainty quantification

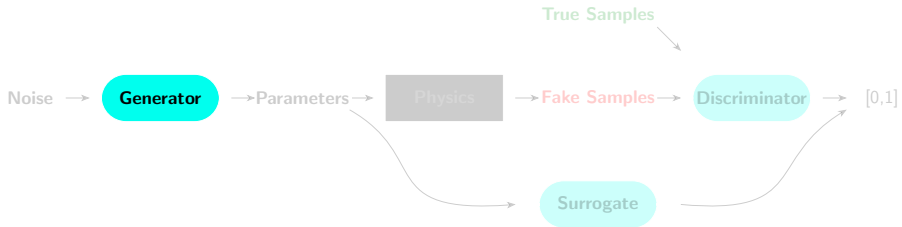
QuantOm SciDAC Project:

- Parallel to QuantOm GAN approach
- My understanding of current GAN approach: also learning parameters via (outer) GAN, but there is a surrogate event generator (inner GAN) as well?

This Work

- We propose a workflow based on **adversarial generative machine learning** for finding these parameters.
- This workflow allows for **uncertainty quantification** of predicted parameters.
 - At least for now, we're interested in aleatoric uncertainty: data quantity remains fixed.
- We demonstrate preliminary results on simplified DIS problem.

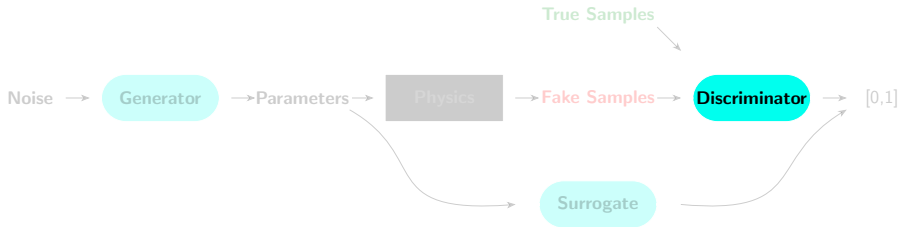




Generator

Goal: Learn to generate parameters.

- Objective function is based on discriminator output
- Want to generate parameters which construct fake events that are indistinguishable from true events to the discriminator



Discriminator

Goal: Learn to distinguish between true and generated samples.

- Output: some prediction between 0 (fake) and 1 (true).
- Training of the overall GAN model converges when the generator and discriminator reach equilibrium
- Convergence is hard! Discriminator loss worsens as generator loss improves.

Quick aside: distances between empirical distributions

To avoid all of the issues that come with GAN training, we spent a lot of time trying out classical methods to compare true and fake samples instead of training a discriminator:

- Optimal Transport-based/Wasserstein distances: worked well on 1-D distributions, not great on 2-D.
- Statistical divergences (Jensen-Shannon, Kullback-Liebler): tried for a long time, didn't work, moved on.
- Distances between empirical probability distributions/histograms with binning: can't differentiate through, could work around that with some kind of surrogate mapping maybe, in any case stopped working on this.

Permutation-Invariant Discriminator Architecture

- Input is essentially a point cloud $X \in \mathbb{R}^{\text{sample size} \times 2}$.
- Discriminator should have same prediction (true/fake data) regardless of row order.
- Need **permutation-invariant** discriminator!
- A few works have investigated this:
 - Deep Sets, 2017¹
 - PointNet, 2016²

¹Zaheer et al. 2018.

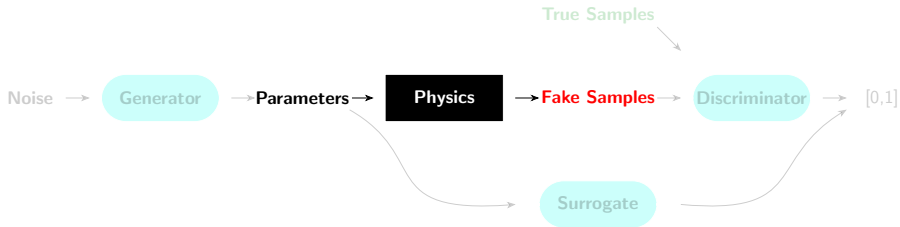
²Qi et al. 2017.

Permutation-Invariant Discriminator Architecture

- General idea of Deep Sets: we apply some (learnable) function ϕ to each of the events $x \in X$, some permutation-invariant aggregation function to the output (e.g. mean, maxpool), and then another (learnable) function ρ to the aggregated output:

$$\text{Discriminator}(X) = \rho(\max(\phi(x) \text{ for } x \in X))$$

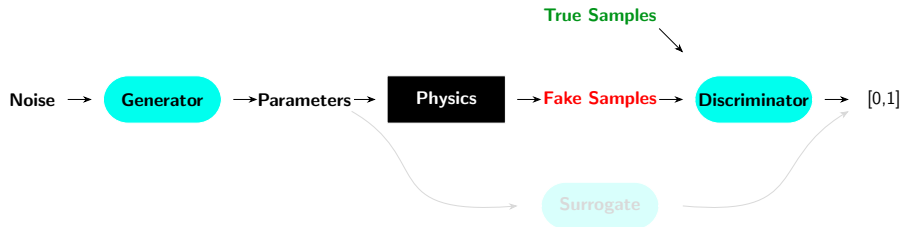
- Implementation: we use simple multi-layer perceptrons for both ρ and ϕ .



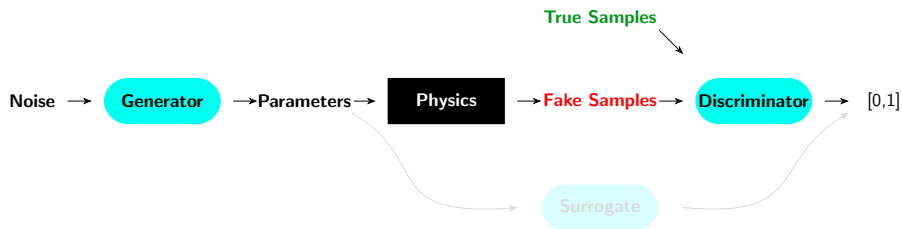
Physics

- Giant disclaimer: using a possibly ancient (?) repository (`quantom-collab/tomography_toolkit_dev`).
- Use parameters
 $p = [0.72916667 \quad 0.25 \quad 0.6 \quad 0.36458333 \quad 0.25 \quad 0.8]^T$ to generate synthetic true data.
- In training, we sample 1024 points based on generated parameters and compare with 1024 true points with the discriminator.
 - 1024 true points taken from a total bank of 102400 data points

Workflow (almost done!)

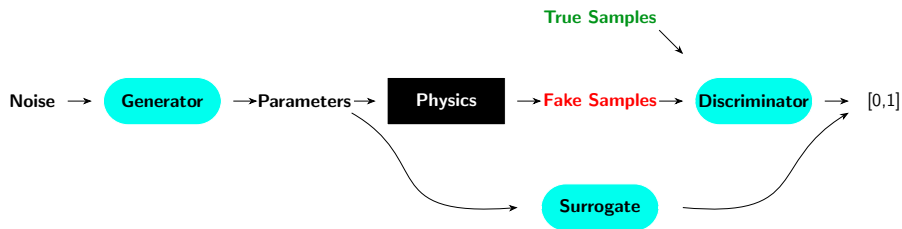


Workflow (almost done!)



⚠ Physics sampling is not invertible/differentiable!

Workflow (almost done!)

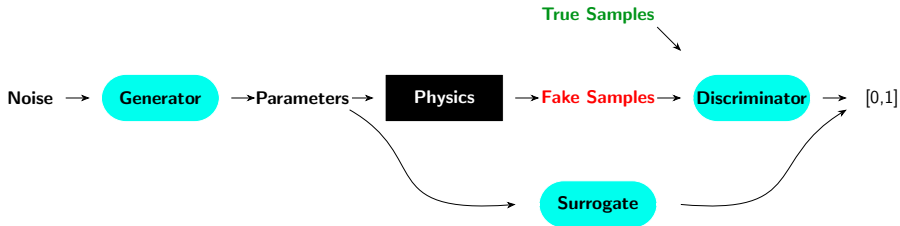


⚠ Physics sampling is not invertible/differentiable!

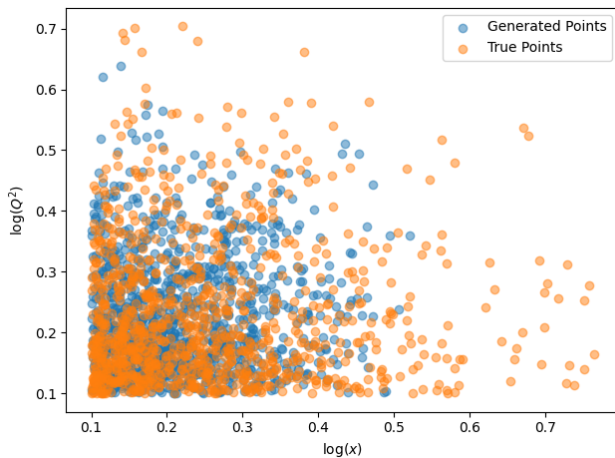
Surrogate Physics Model

Goal: Learn a surrogate mapping between the parameters and the discriminator output.

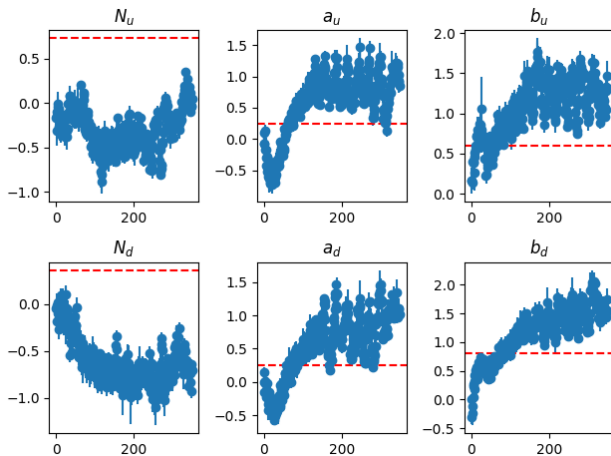
- Learn in conjunction with discriminator
- Only need during training; can ignore during inference



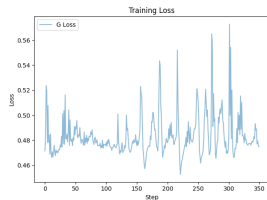
Results: Generated Events



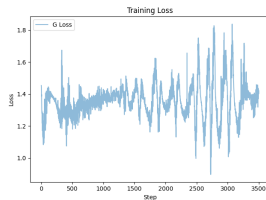
Results: Parameters



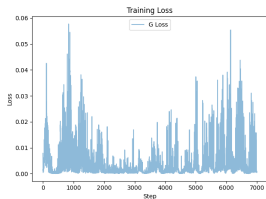
Results: Training



(a) Generator



(b) Discriminator



(c) Surrogate Physics

Conclusions

Questions:

- Is there a unique solution to finding parameters based purely on events? Is it okay if we find **one** solution?
- Is there value in being able to generate realistic events even if generated parameters are distant from true parameters?
- Importance of outliers?
- Physics constraints?



Thoughts for future work:

- Reducing uncertainty from absence of data with using more true data/investigating sensitivity of results to the fixed sample size (1024)
- GAN training improvements (Hinge/Wasserstein loss, etc.)
- Avoid GAN convergence pitfalls entirely and try to train some kind of surrogate mapping to a histogramming-based distance
- Fancier discriminator (Deep Sets with Attention, PointNet/PointNet++)

Thank you!

- This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112.
- We also acknowledge support from the INT for this presentation as well as RIKEN Center for Computational Science for sponsoring an internship and subsequent research visit.

Bibliography I

-  Qi, Charles R. et al. (2017). *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. arXiv: 1612.00593.
-  Zaheer, Manzil et al. (2018). *Deep Sets*. arXiv: 1703.06114.