Simulating the response of DarkSide-20k using GANs

ECHEP/Excalibur Workshop

Krishan Jethwa, Enrico Zammit Lonardelli, Darren Price, Stephen Menary

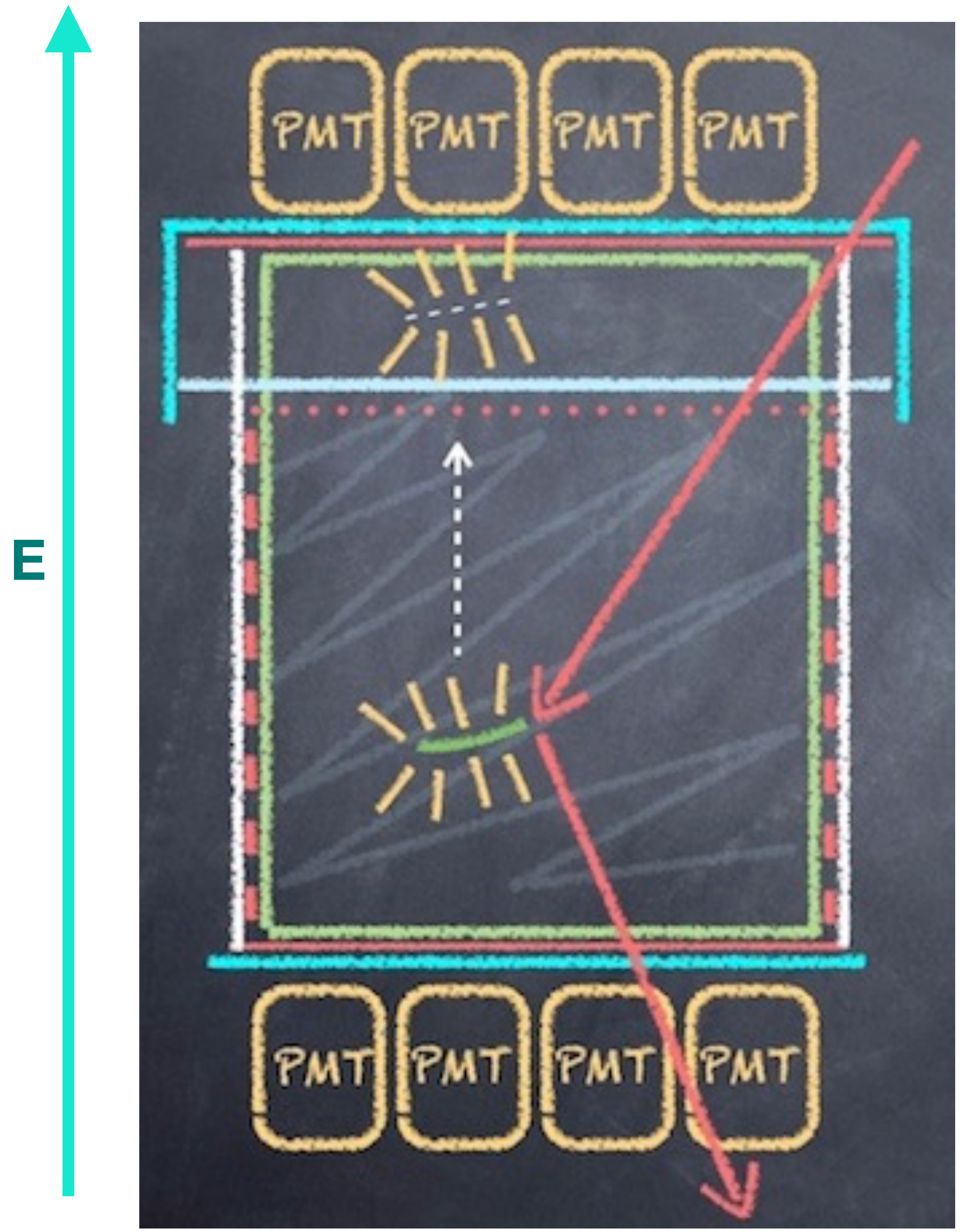
Intro

- **Question:** can we use generative adversarial networks (GANs) to accelerate detector simulation when searching for **WIMPs** using **LAr TPC** experiments (**DarkSide-20k**)?
- **Context:** MPhys project conducted by **Krishan Jethwa** and **Enrico Zammit Lonardelli** (2019/20)

- **Background:**
  - GANs invented in 2016 (arXiv:1406.2661) as a new type of generative model, and have gained much popularity, especially in context of image generation, since they can produce **high fidelity outputs** (e.g. super-resolution)
  - Have been used to model calorimeters @ LHC: arXiv:1712.10321, arXiv:1812.00879, arXiv:2005.05334, ATL-SOFT-PUB-2018-001, ATLAS-SIM-2019-004
  - Much faster than running G4 every time (but we still use G4 to train the GAN)

- **Aims:**
  - See how well we can get a GAN implementation to work in a LAr TPC setting
  - Iterate discussion on how/where ML tools can contribute to efficient simulation LAr experiments

DarkSide-20k



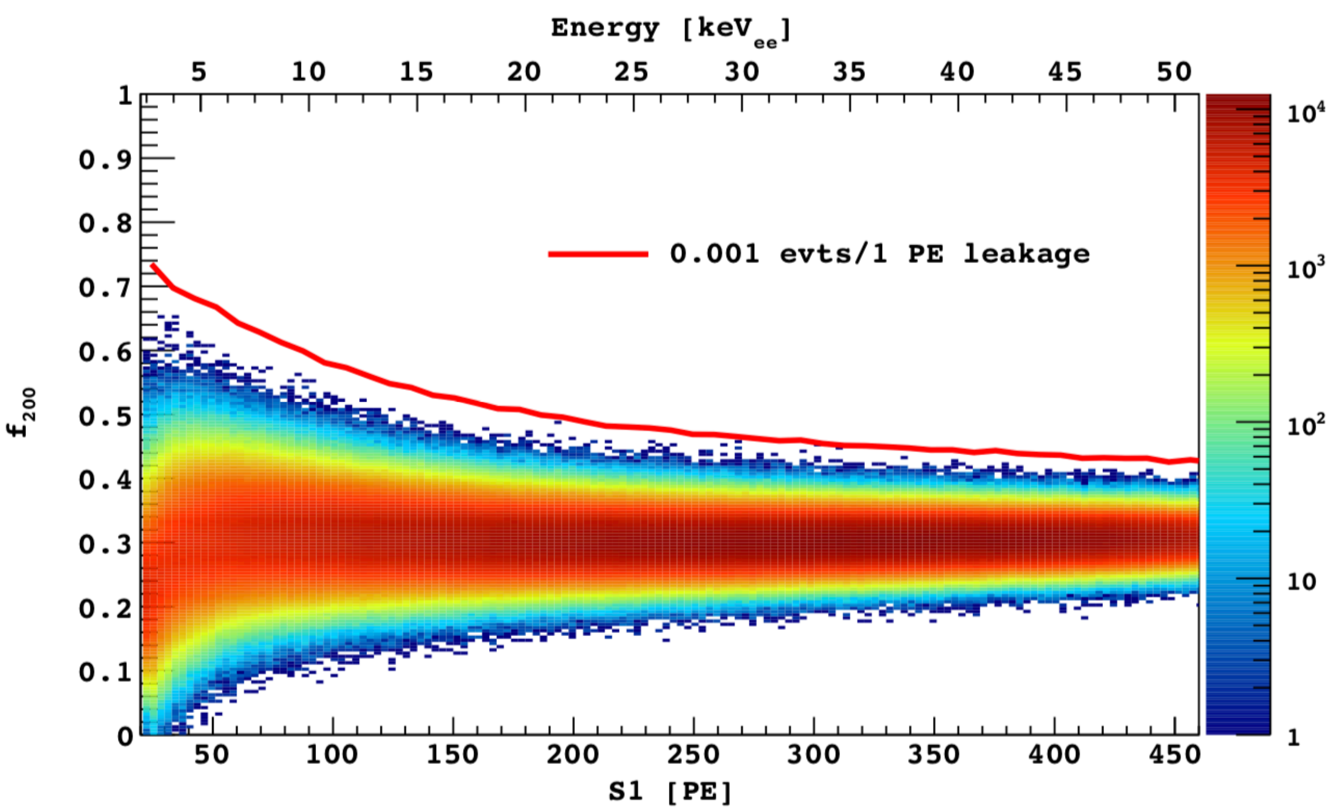
DarkSide-50
Image credit: http://darkside.lngs.infn.it/argon-tpc/

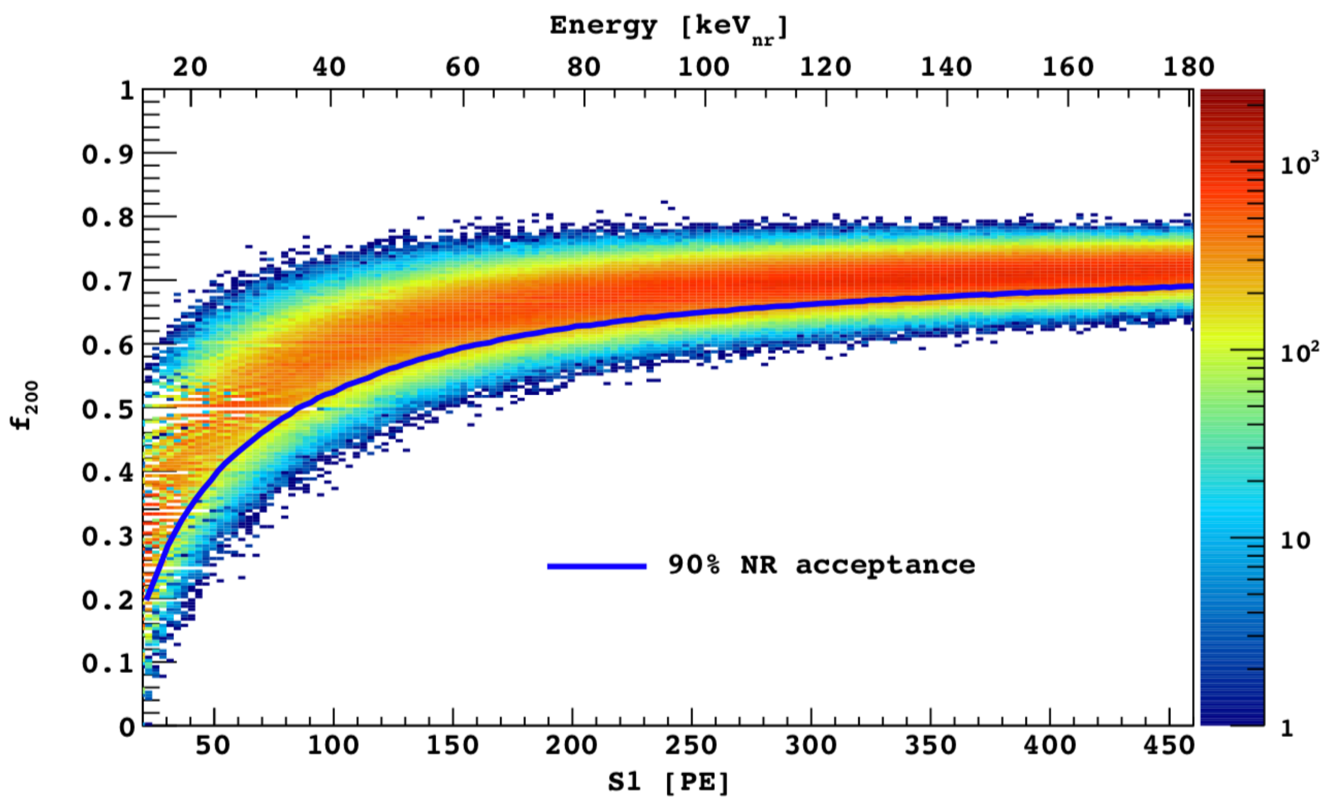
***DarkSide-20k uses SiPMs, not PMTs**

- **Dual-phase LAr time-projection-chamber (TPC)**
- **WIMP** collision causes **nuclear recoil**, releasing **ionisation electrons**
- Also create excited LAr dimers, which de-excite and create a **prompt scintillation signal (S1)**
- Free charges are accelerated in electric field until they cross the LAr liquid->gas boundary, creating a **secondary scintillation signal (S2)**
- Radiation background: e.g. electron recoil. Less ionising than DM, so can distinguish using S1/S2/f200 spectrum
  - f200 = fraction of S1 in first 200ns (pulse shape discrimination)

DarkSide-20k

Figures credit: Eur. Phys. J. Plus (2018) 133: 131, arXiv:1707.08145





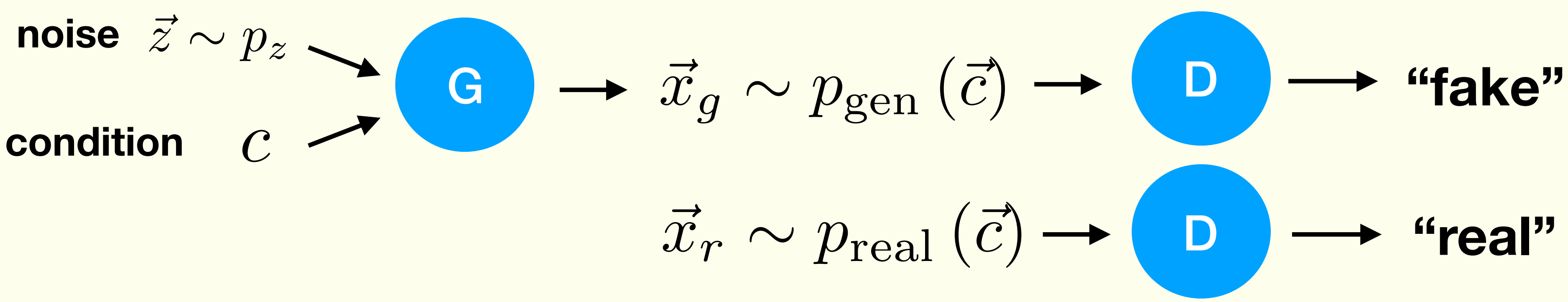
Characteristic background
(36Ar β-decay, electron recoil)

Characteristic signal
(nuclear recoil)

Important to learn joint-probability across multiple observables

Implementation

Vanilla GANs



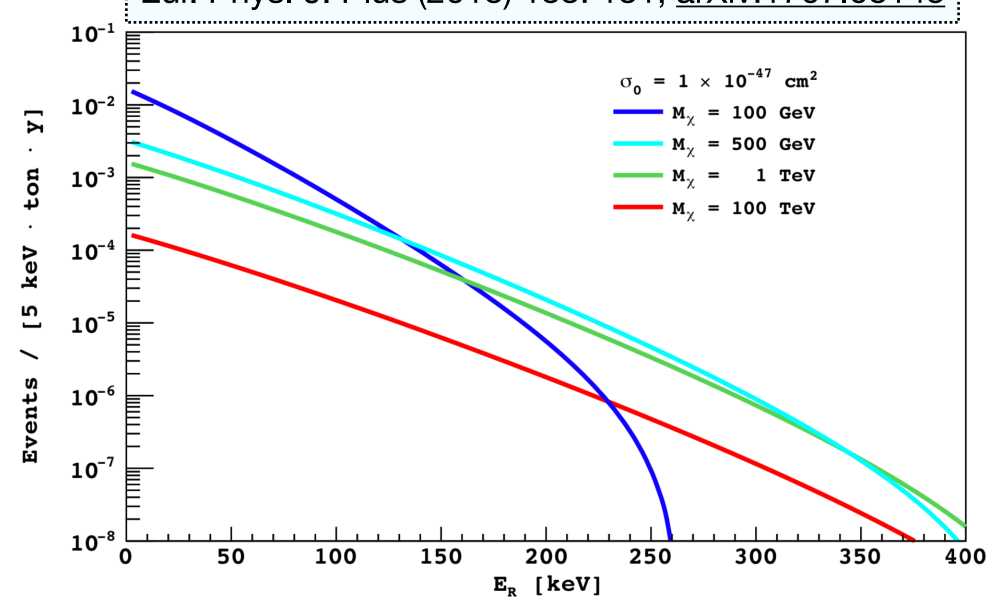
- Train G to fool D, and D to not be fooled (2-player minimax game, "adversarial")
- Optimum training —> $p_{gen} = p_{real}$
- **We use recoil energy as conditional parameter**

Auto-regressive property

$$p(A, B|C) \;=\; p(A|B, C) \;\times\; p(B|C)$$

- Use an auto-regressive GAN = a chain of GANs with 1D output(but increasing complex inputs)

Eur. Phys. J. Plus (2018) 133: 131, arXiv:1707.08145

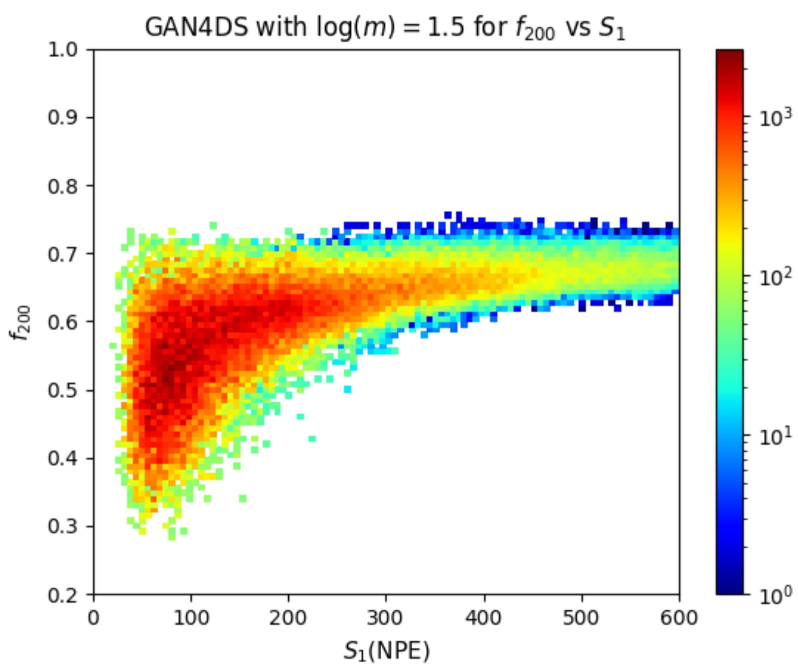


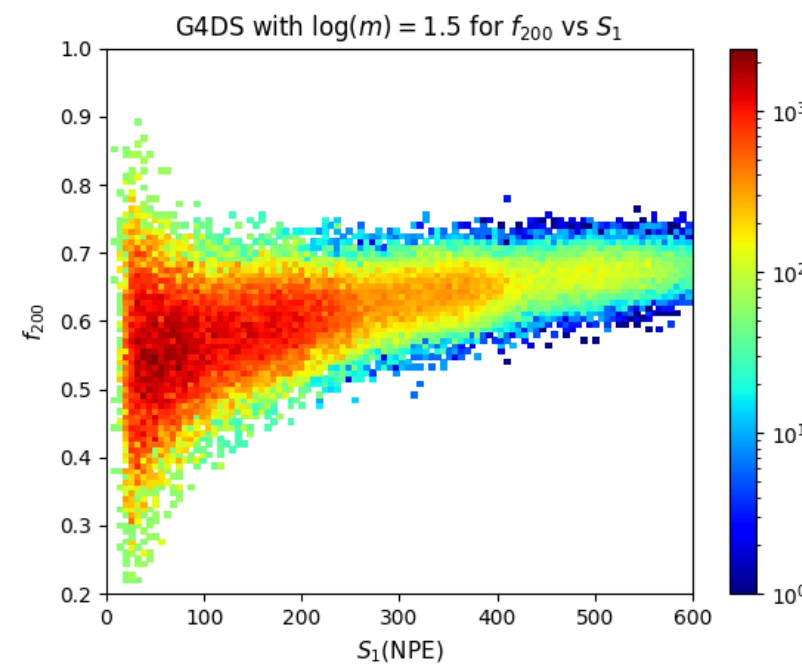
Results

GAN4DS

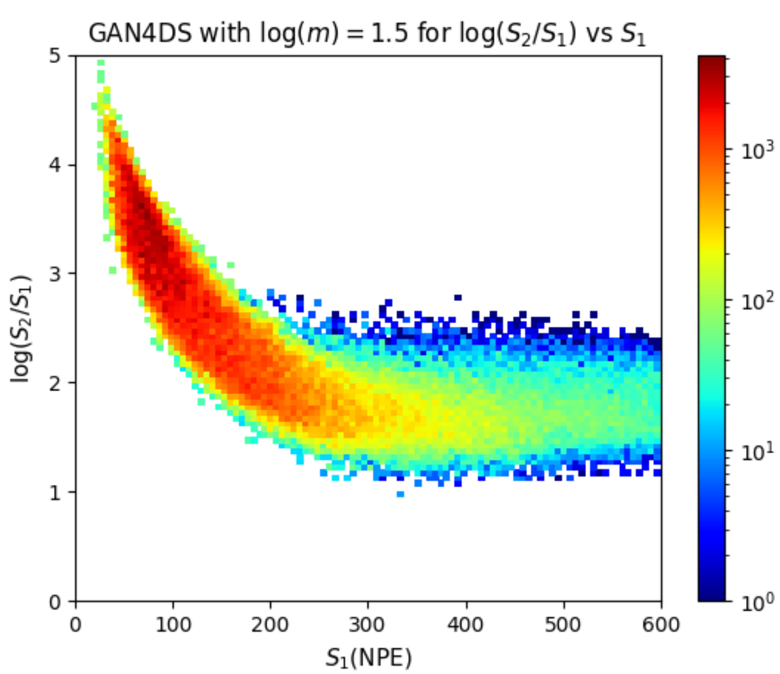
G4DS

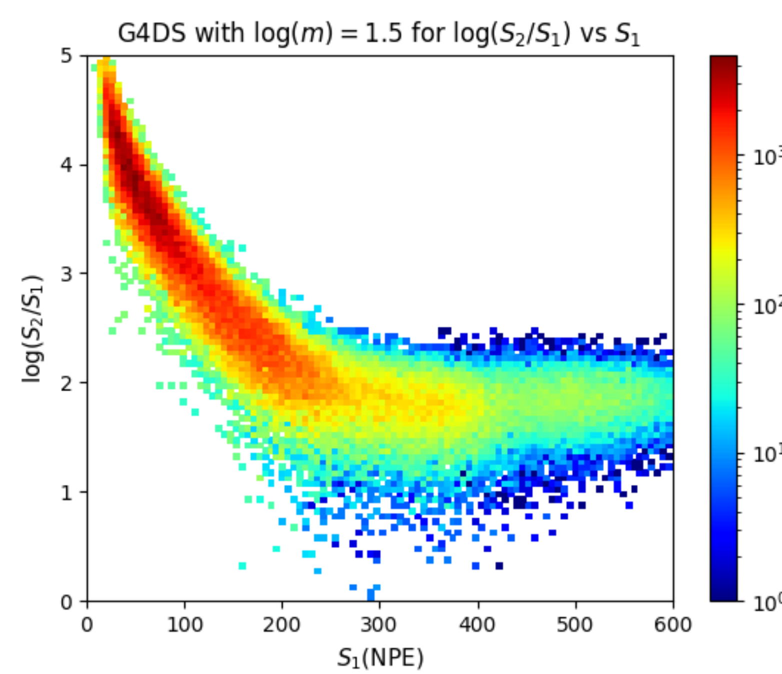
f200





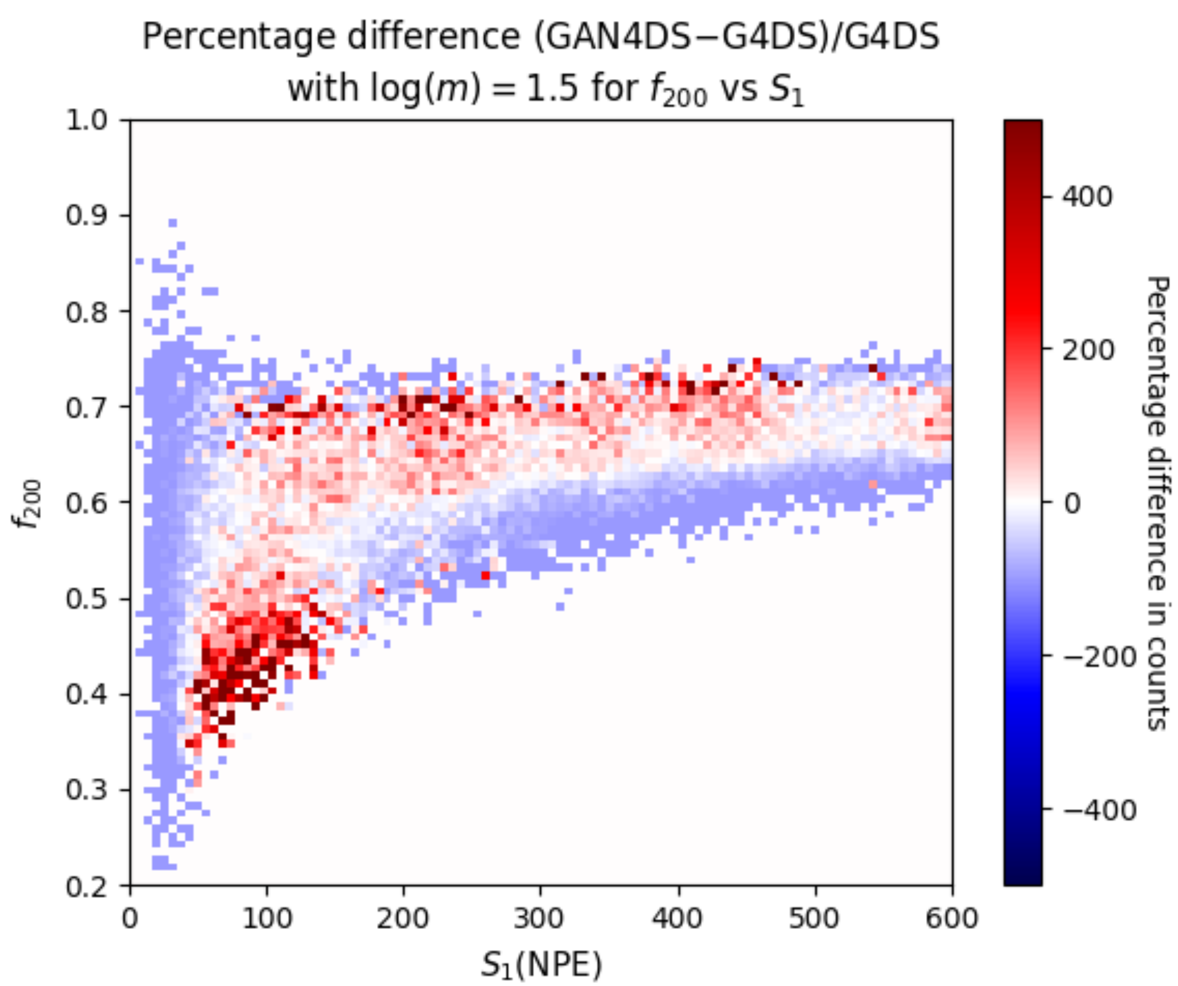
log(S1/S2)

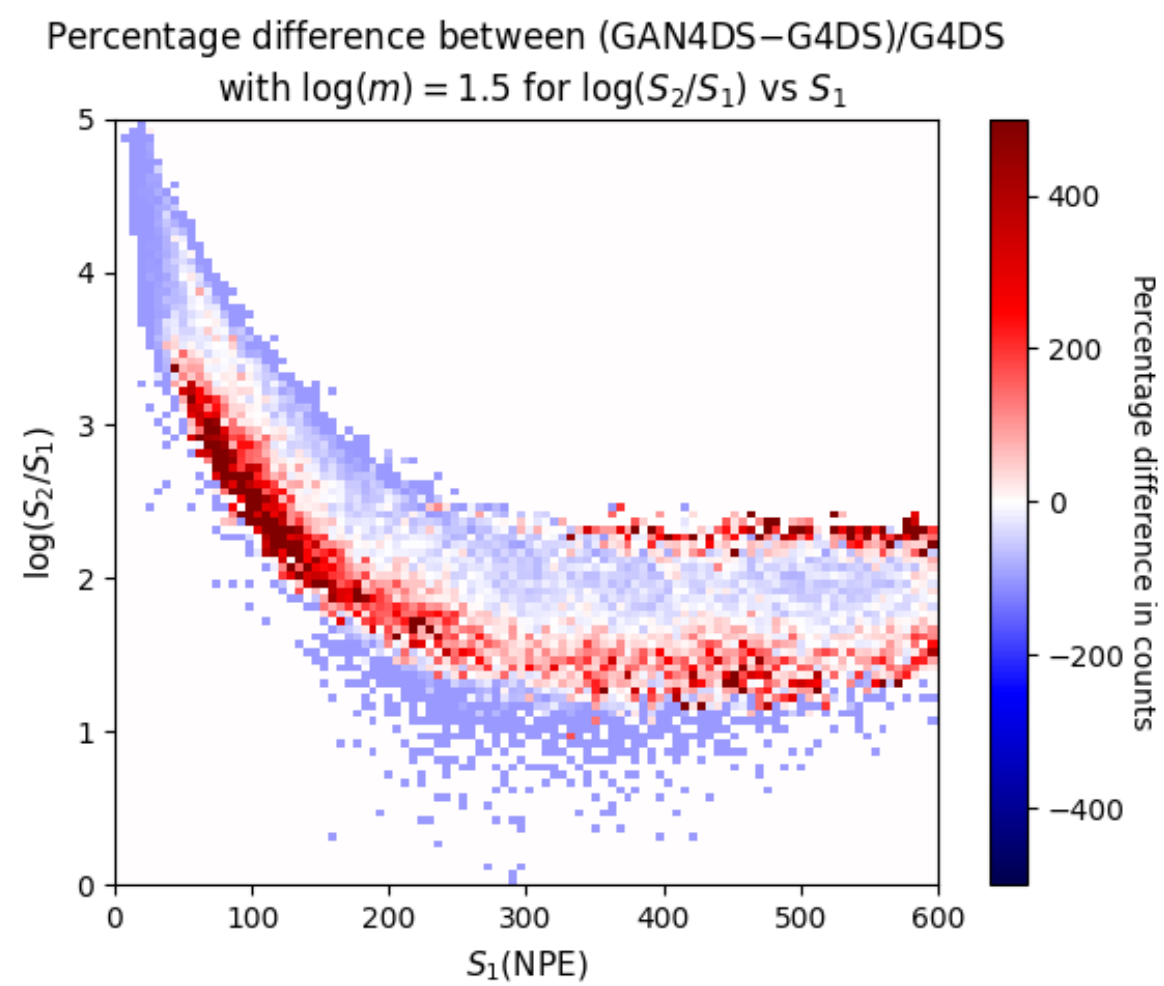




S1

Results

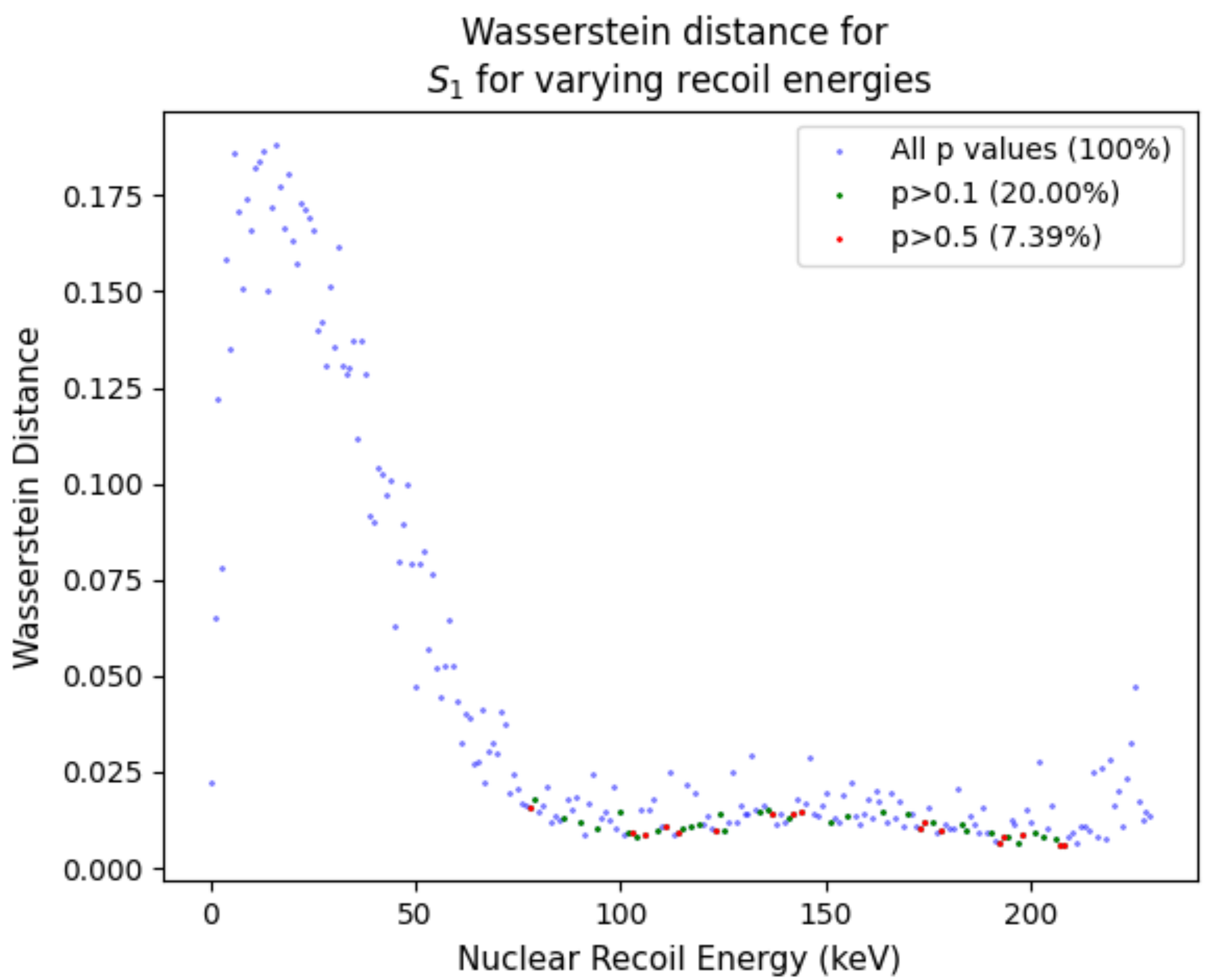




Typical accuracy ~O(50-400%)

Interesting systematic trend - can probably be improved?

Results



Conditional dependence learned with ~medium success

Wasserstein distance is metric comparing G4DS & GAN4DS

Results

Processing Times Comparison

	G4DS	GAN4DS
Trainin g		222 hrs (~9 days)
Generation	78 hrs (~3 days)	1.5 mins

- G4DS/GAN benchmarks each for 1000 events per NR energy, for energies [5,235] keV in steps of 1 keV
- GAN trained on a NVIDIA TESLA V100 16GB memory, Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 48 GB RAM
- G4DS obtained on AMD Opteron(tm) Processor 6278 @ 2.4 GHz and 132 GB RAM

Factor ~3000 improved run-time performance compared with G4DS

But not necessarily fair comparison, as people would likely use fast-sim in real world

Discussion

- **Shown a GAN can describe qualitative characteristic signatures of WIMPs in LAr TPC**
- **Improvements:** positional/directional dependence, dependence on latent variables (e.g. Rayleigh scattering length)
- **What is the best input?** Currently nuclear recoil. To generalise to more LAr TPC uses, include electron recoil. Maybe better to factor out nuclear interaction and condition on “truth level photons” at certain positions and times.
- **What is the best output?** High level observables like S1/S2/f200, or something more low-level like an image of SiPM hits? The latter might require a DCGAN
- **GAN training is very unstable,** in part because the objective functions of D and G constantly change over time. Often results in e.g. mode collapse, catastrophic forgetting, biased model, even though “fully trained” GAN would be unbiased. Could benefit from much prior work here.
- **Alternative:** neural likelihood models may be more stable and so less biased for a low-fidelity output, but usually slower/harder to sample, and not good for image generation (SiPM hit)
- **For more info:** https://indico.cern.ch/event/919221/