Anomaly aware unsupervised learning for dark matter direct detection

Andre Scaffidi

In collaboration with with Juan Herrero-Garcia and Riley Patrick

September 9, 2021

Istituto Nazionale di Fisica Nucleare, Sezione di Torino, via P. Giuria 1, I–10125 Torino, Italy

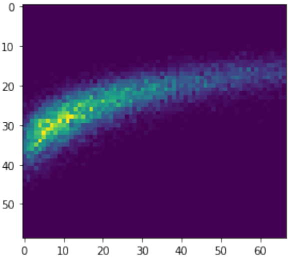
INFN PRIN

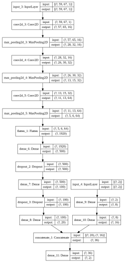
The Dark Universe: A Synergic Multimessenger Approach

BAM

Overview

- Apply Machine Learning (ML) to direct detection of dark matter? Long thought difficult due to low statistics.





$m_\chi, \sigma,$ Bkg/Sig?

Machine Learning

One slide summary...

'Learn' **characteristic features** of data in order to make predictions or decisions.

Dark Matter Direct Detection

XENON1T as a test-bed: Two types of events

- Nuclear Recoil (NR) $\rightarrow$ WIMPs
- Background dominated by Electron Recoil (ER).

XENON1T as a test-bed: Pax simulation

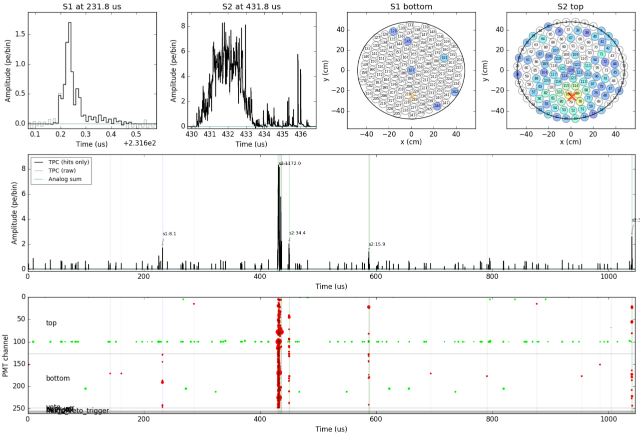


Figure 1: Nuclear recoil (NR) event example image from arxiv:1911.09210.

- S1: Prompt scintillation signal from recoil event.
- S2: Electron charges produced during ionization drift upwards → extracted into gaseous phase creating larger scintillation.

Supervised classification

Classification: Training data generation

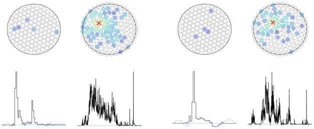
Khosa, Mars , Richards, Sanz, 1911.09210

recoil_type	x	y	depth	s1_photons	s2_electrons	t
NR	-29.45	-9.25	59.78	309	134	225461
NR	19.46	25.34	60.88	103	58	393624
NR	-9.55	-36.81	33.23	255	100	139897
NR	-33.56	-12.19	64.44	151	83	292325
NR	25.04	-26.73	63.96	270	85	295540

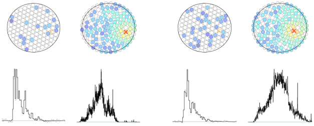
PAX
Processor for Analyzing XENON

Generate image training set with 4×10^4 images total. **Fixed mass = 500 GeV, $\boldsymbol{\sigma} = \mathbf{10^{-45}}$ cm^2:**

$$\mathbf{x}_{\mathrm{NR}}^{\mathrm{Train}} = \left(\quad , \quad , \ldots \right)$$

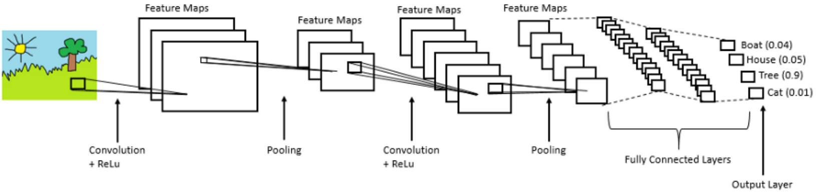


$$\mathbf{x}_{\mathrm{ER}}^{\mathrm{Train}} = \left(\quad , \quad , \ldots \right)$$



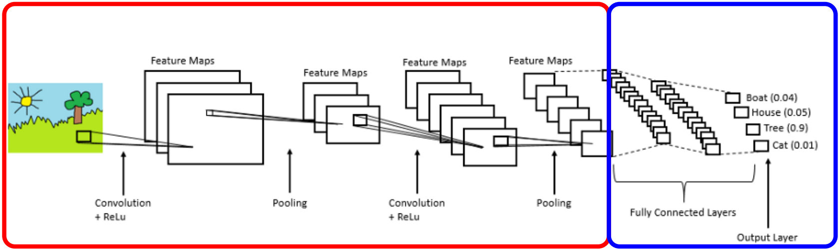
Classification: Convolutional Neural Network (CNN)

- Used when input data are pixels



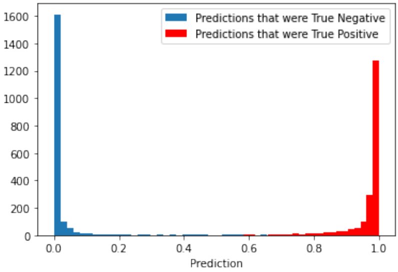
Classification: Convolutional Neural Network (CNN)

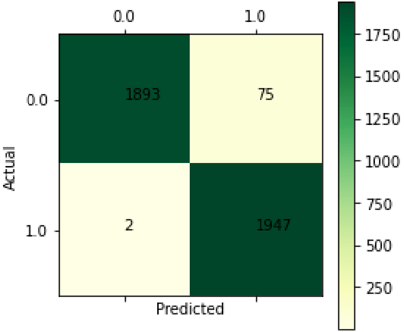
- Convolution and pooling layers: Decompose RGB information in image to be interpereted by 'fully connected' neural net
- Regular neural net performs classification



Classification: Signal vs. Background Results

- Train on $\sim$ 40000 images. Take testing sub-sample of $\sim$ 40% of MC set
- Run 'testing set' through trained network $\rightarrow$ check performance!

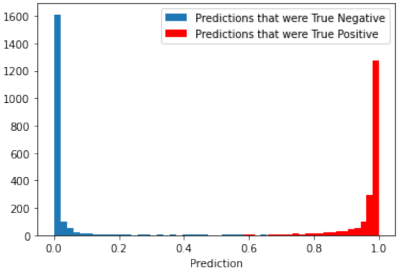


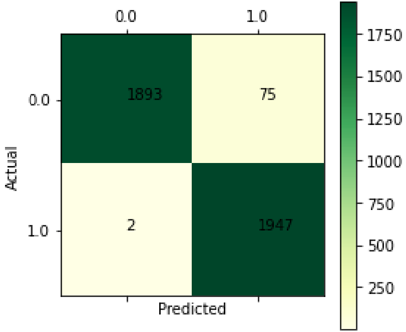


- Takeaway $\Rightarrow$ **98.03% accuracy**. (Recall = 98.07%, Precision = 96.39%)
- **We find this works regardless mass and cross-section**

Classification: Signal vs. Background Results

- Train on $\sim$ 40000 images. Take testing sub-sample of $\sim$ 40% of MC set
- Run 'testing set' through trained network $\rightarrow$ check performance!

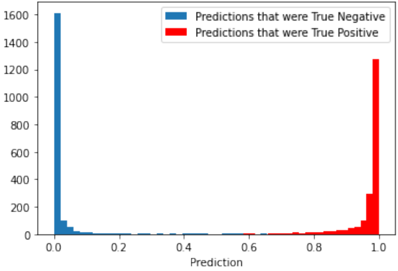


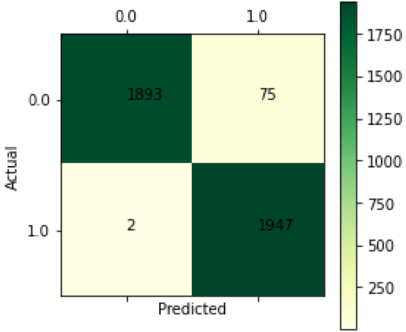


- Takeaway $\Rightarrow$ **98.03% accuracy**. (Recall = 98.07%, Precision = 96.39%)
- **We find this works regardless mass and cross-section**

Classification: Signal vs. Background Results

- Train on $\sim$ 40000 images. Take testing sub-sample of $\sim$ 40% of MC set
- Run 'testing set' through trained network $\rightarrow$ check performance!

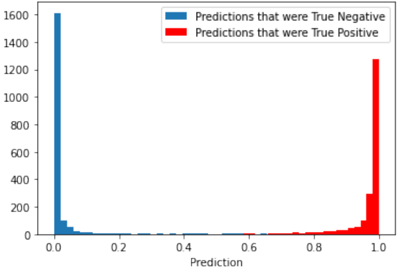


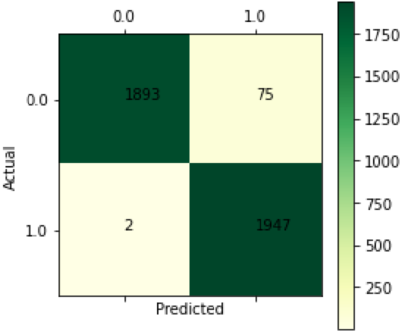


- Takeaway $\Rightarrow$ **98.03% accuracy**. (Recall = 98.07%, Precision = 96.39%)
- **We find this works regardless mass and cross-section**

Classification: Signal vs. Background Results

- Train on $\sim$ 40000 images. Take testing sub-sample of $\sim$ 40% of MC set
- Run 'testing set' through trained network $\rightarrow$ check performance!

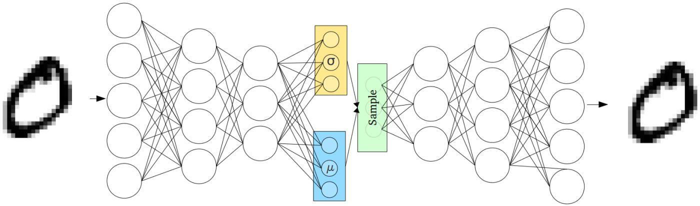




- Takeaway $\Rightarrow$ **98.03% accuracy**. (Recall = 98.07%, Precision = 96.39%)
- **We find this works regardless mass and cross-section**

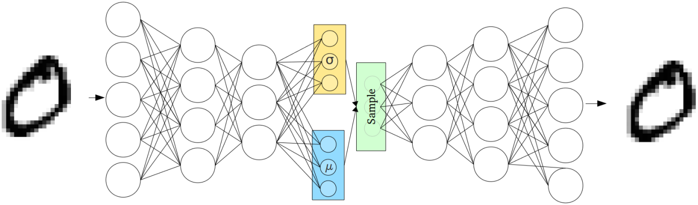
Generative Deep Learning: The Variational Auto-Encoder

- Unsupervised $\Rightarrow$ No labels!
- Goal: Learn low dimensional representation (encoding) of data via dimentionality reduction.



Generative Deep Learning: The Variational Auto-Encoder

- Unsupervised $\Rightarrow$ No labels!
- Goal: Learn low dimensional representation (encoding) of data via dimentionality reduction.



Variational-Auto-Encoder: Method

- Generate image data as with supervised CNN.
- Train CVAE on just* ER background data. Reconstruction loss function:

$$L = \frac{1}{N}\sum_{i=0}^{N}(x_i - y_i)^2 + \beta\sum_{i=0}^{K}\left[\sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1\right]$$

K = number of latent space normal distributions.

x = Input .

y = Reconstructed output.

β = Regularization parameter.

- **Anomaly Detection**: Run data the network has never seen before through trained network.
- Loss distribution of anomalous data (new physics) will show as an **excess** over background only loss distribution.

Variational-Auto-Encoder: Method

- Generate image data as with supervised CNN.
- Train CVAE on just* ER background data. Reconstruction loss function:

$$L = \frac{1}{N}\sum_{i=0}^{N}(x_i - y_i)^2 + \beta\sum_{i=0}^{K}\left[\sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1\right]$$

K = number of latent space normal distributions.

x = Input .

y = Reconstructed output.

β = Regularization parameter.

- **Anomaly Detection**: Run data the network has never seen before through trained network.
- Loss distribution of anomalous data (new physics) will show as an **excess** over background only loss distribution.

Variational-Auto-Encoder: Method

- Generate image data as with supervised CNN.
- Train CVAE on just* ER background data. Reconstruction loss function:

$$L = \frac{1}{N}\sum_{i=0}^{N}(x_i - y_i)^2 + \beta\sum_{i=0}^{K}\left[\sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1\right]$$

K = number of latent space normal distributions.

x = Input .

y = Reconstructed output.

β = Regularization parameter.

- **Anomaly Detection**: Run data the network has never seen before through trained network.
- Loss distribution of anomalous data (new physics) will show as an **excess** over background only loss distribution.

Variational-Auto-Encoder: Method

- Generate image data as with supervised CNN.
- Train CVAE on just* ER background data. Reconstruction loss function:

$$L = \frac{1}{N}\sum_{i=0}^{N}(x_i - y_i)^2 + \beta\sum_{i=0}^{K}\left[\sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1\right]$$

K = number of latent space normal distributions.

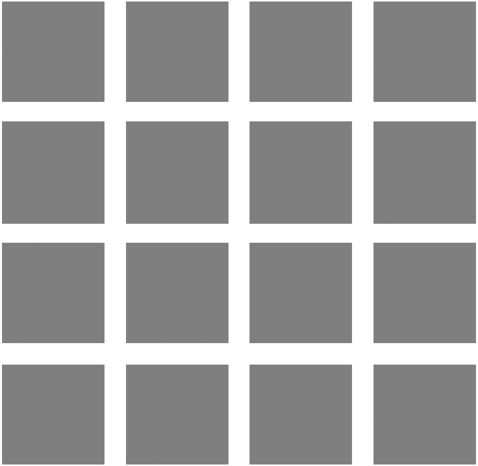
x = Input .

y = Reconstructed output.

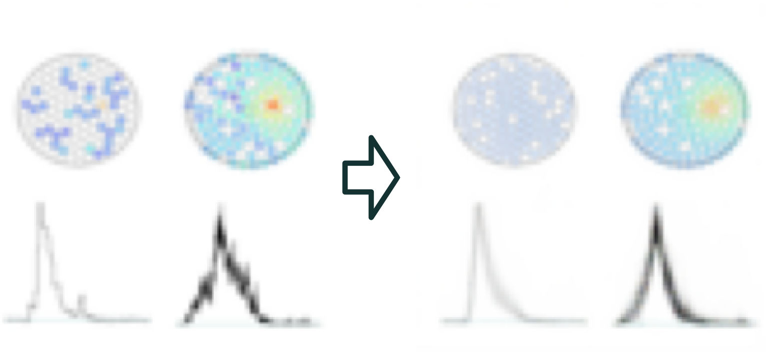
β = Regularization parameter.

- **Anomaly Detection**: Run data the network has never seen before through trained network.
- Loss distribution of anomalous data (new physics) will show as an **excess** over background only loss distribution.

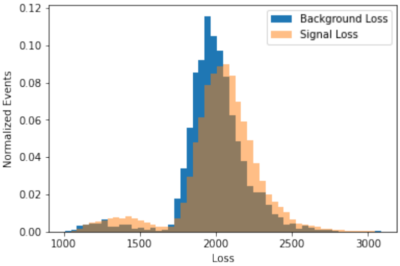
Generative Deep Learning: Results

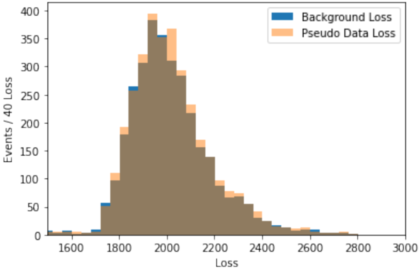


If gif didn't work...



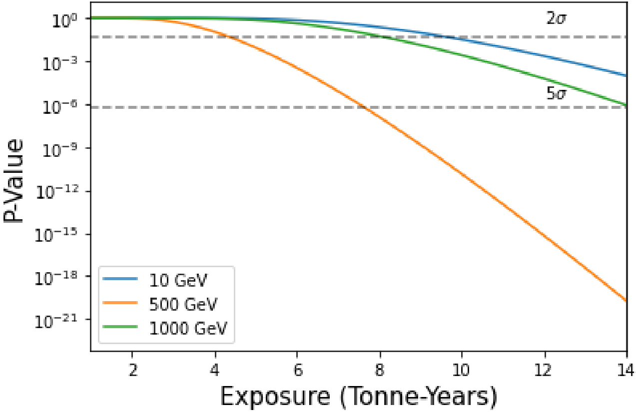
Variational-Auto-Encoder: Results





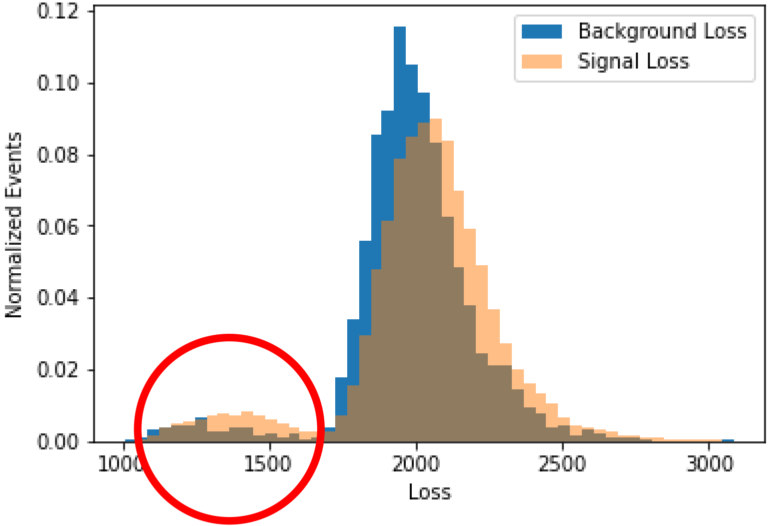
- Left: Background loss distribution + just* signal loss distribution.
- Right: Inject signal into background signal, run whole data set through network.
- Any* anomalous signal will show up as statistical deviation in (pseudo)data loss vs. (known) background loss.

Variational-Auto-Encoder: Results

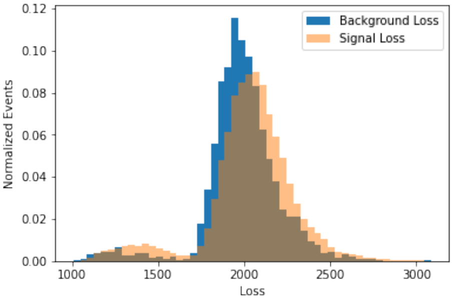


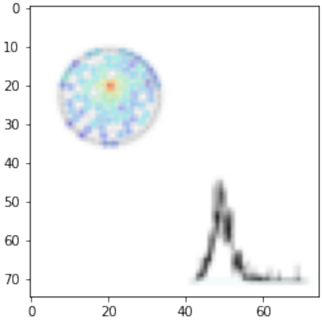
- Float exposure to get discovery significance for some sample masses.

Variational-Auto-Encoder: Interesting discovery...



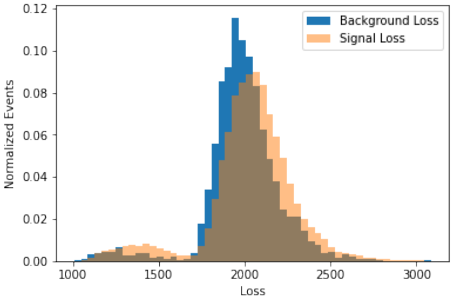
Variational-Auto-Encoder: Interesting discovery...

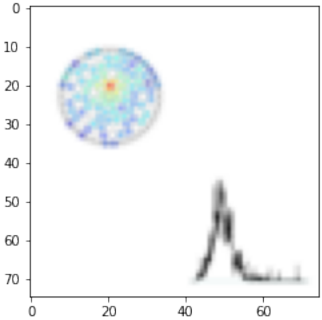




- Actually events with no S1 (S2 only)
- S2-only analyses been done before. Could extract more power from such an analysis with ML?

Variational-Auto-Encoder: Interesting discovery...





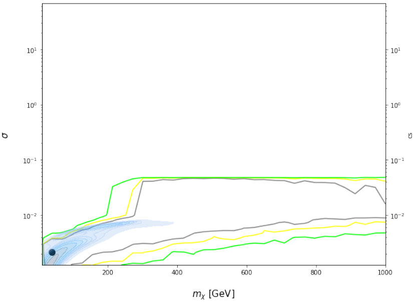
- Actually events with no S1 (S2 only)
- S2-only analyses been done before. Could extract more power from such an analysis with ML?

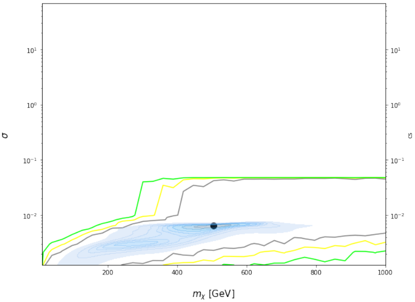
Just the tip...

Currently also looking at :

- Non-adversarial supervision
- Parameter regression with semi-unsupervised methods.
- Need better classifier to forecast uncertainty in network performance.
- Discriminate between DM models using anomaly detection methods?
- S2-only as discriminator?

Anomaly guided supervised regression:





Thank you!