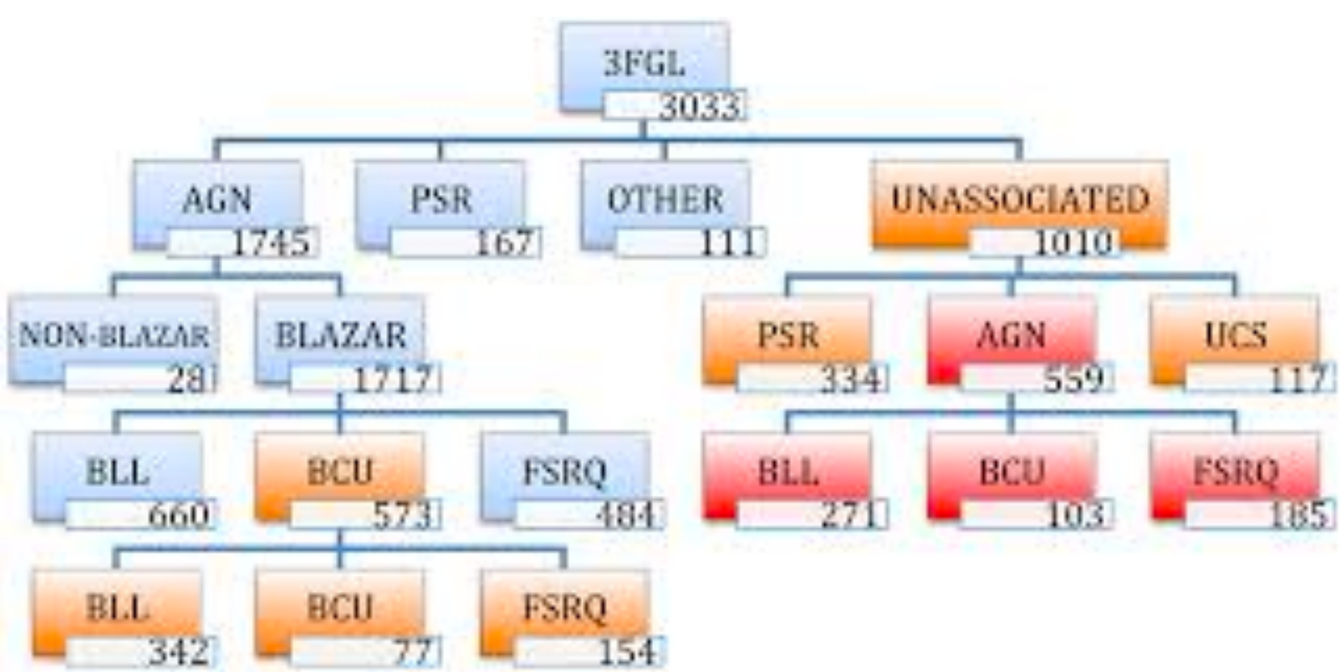
γ-RAYS AND MACHINE LEARNING

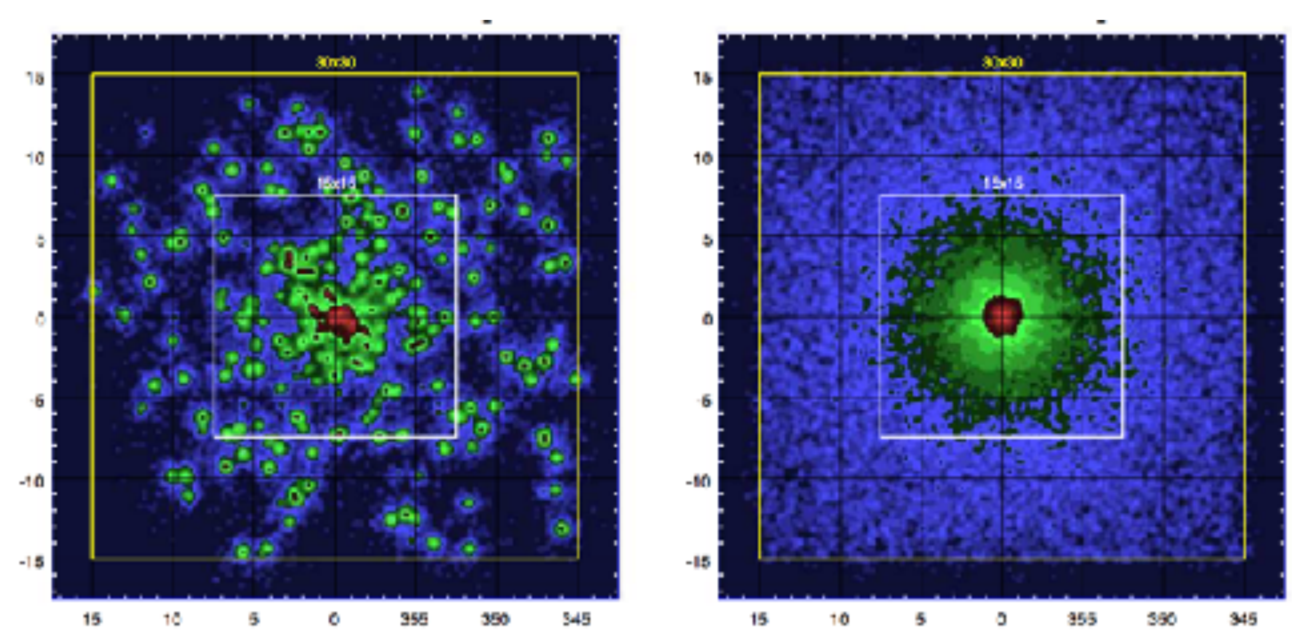
LUC HENDRIKS

RADBOUD UNIVERSITY, NIJMEGEN (NL)

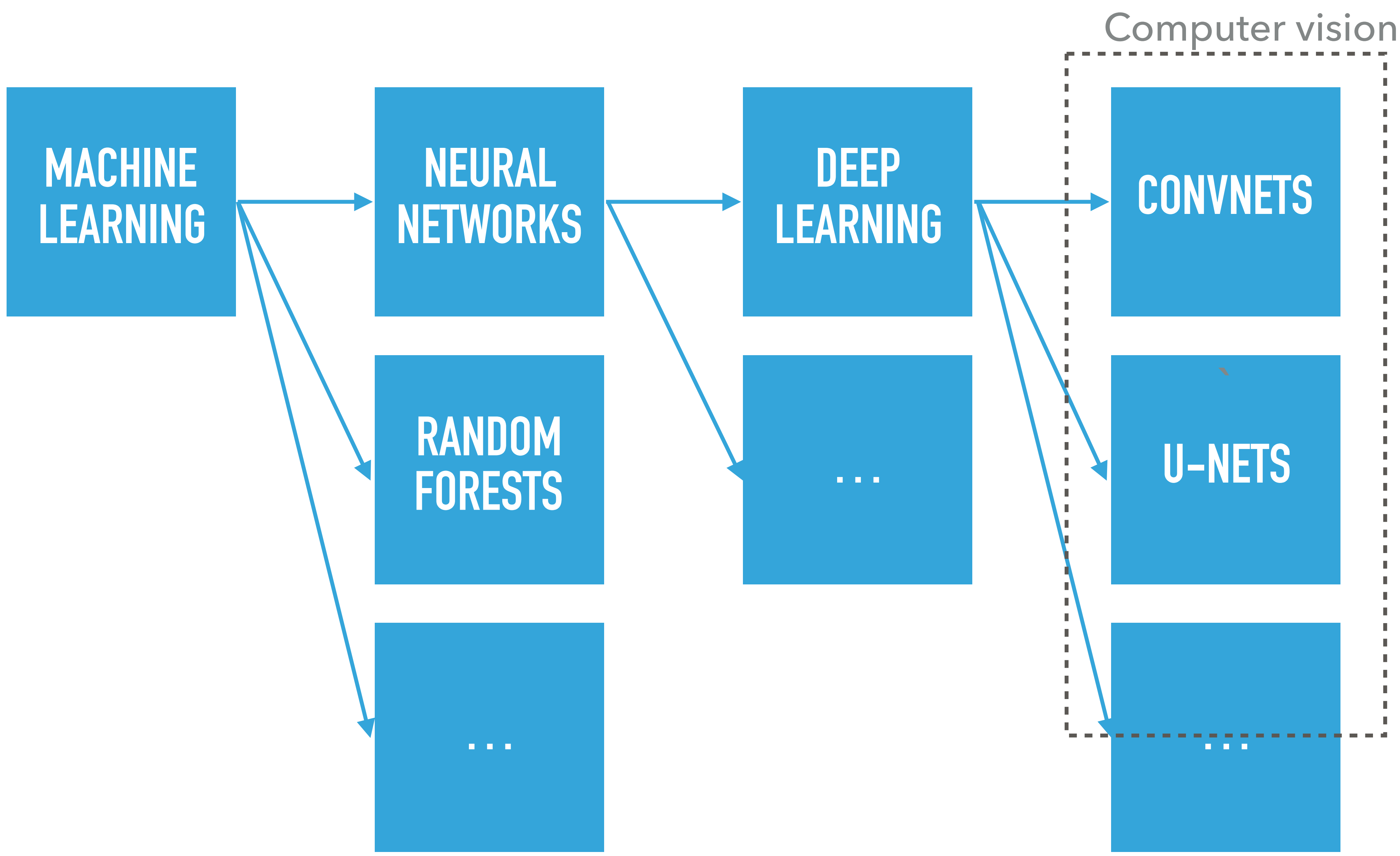
INTRODUCTION

- Discuss two questions
  - Is the GC excess mainly due to a collection of point sources or a truly diffuse source like annihilating DM?
  - What source classes are the ~33% of point sources found in the 3FGL catalog?
- Can we answer these questions with the help of machine learning?
  - Satellite data is basically image data, which means we can use Facebook & Google's stuff!
- I will give a few examples of how computer vision could help γ-ray astronomy and how we apply it



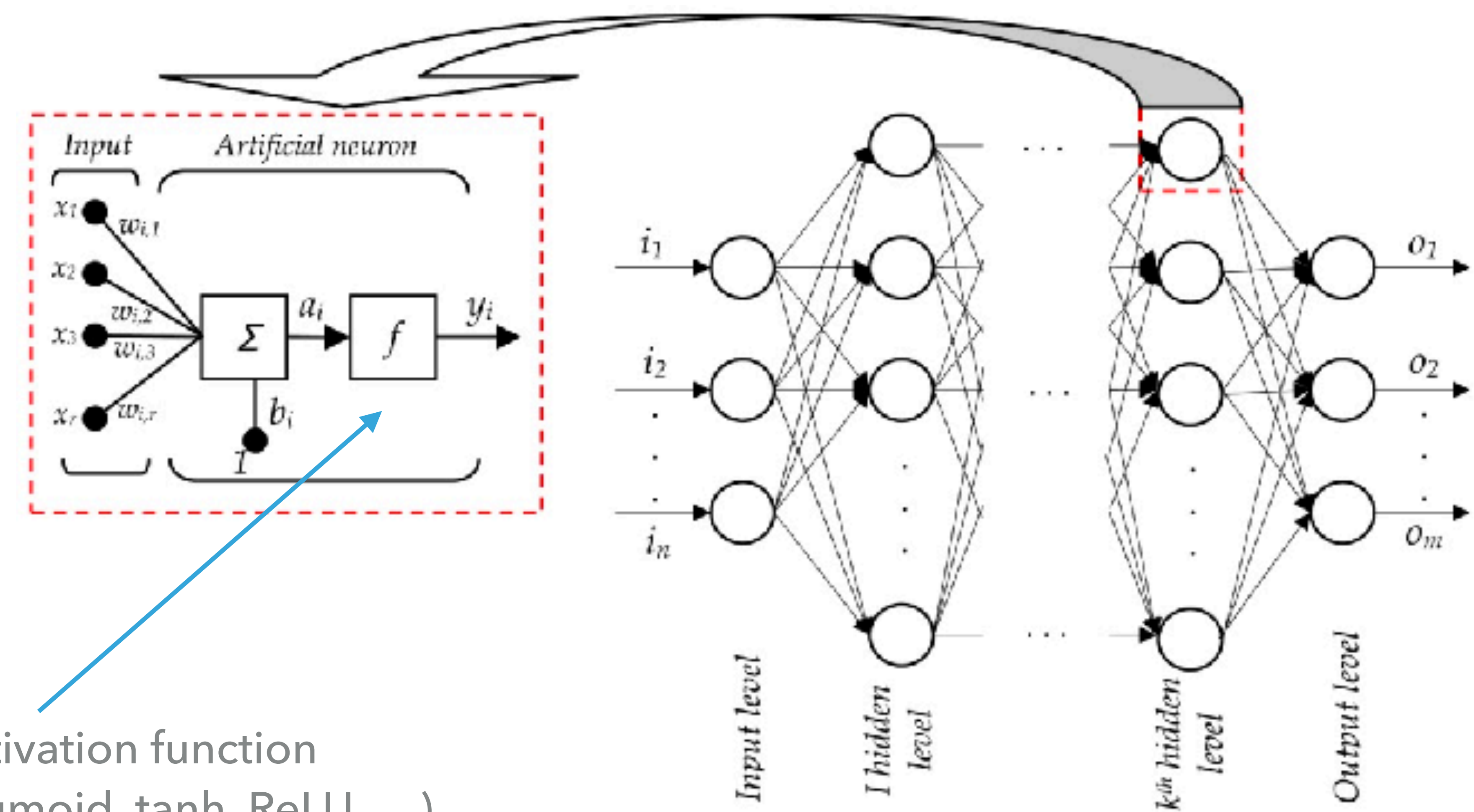


WHAT IS COMPUTER VISION?



Instead of the programmer defining what the computer should do (IFTTT), supply an objective and learn from data

NEURAL NETWORKS & DEEP LEARNING

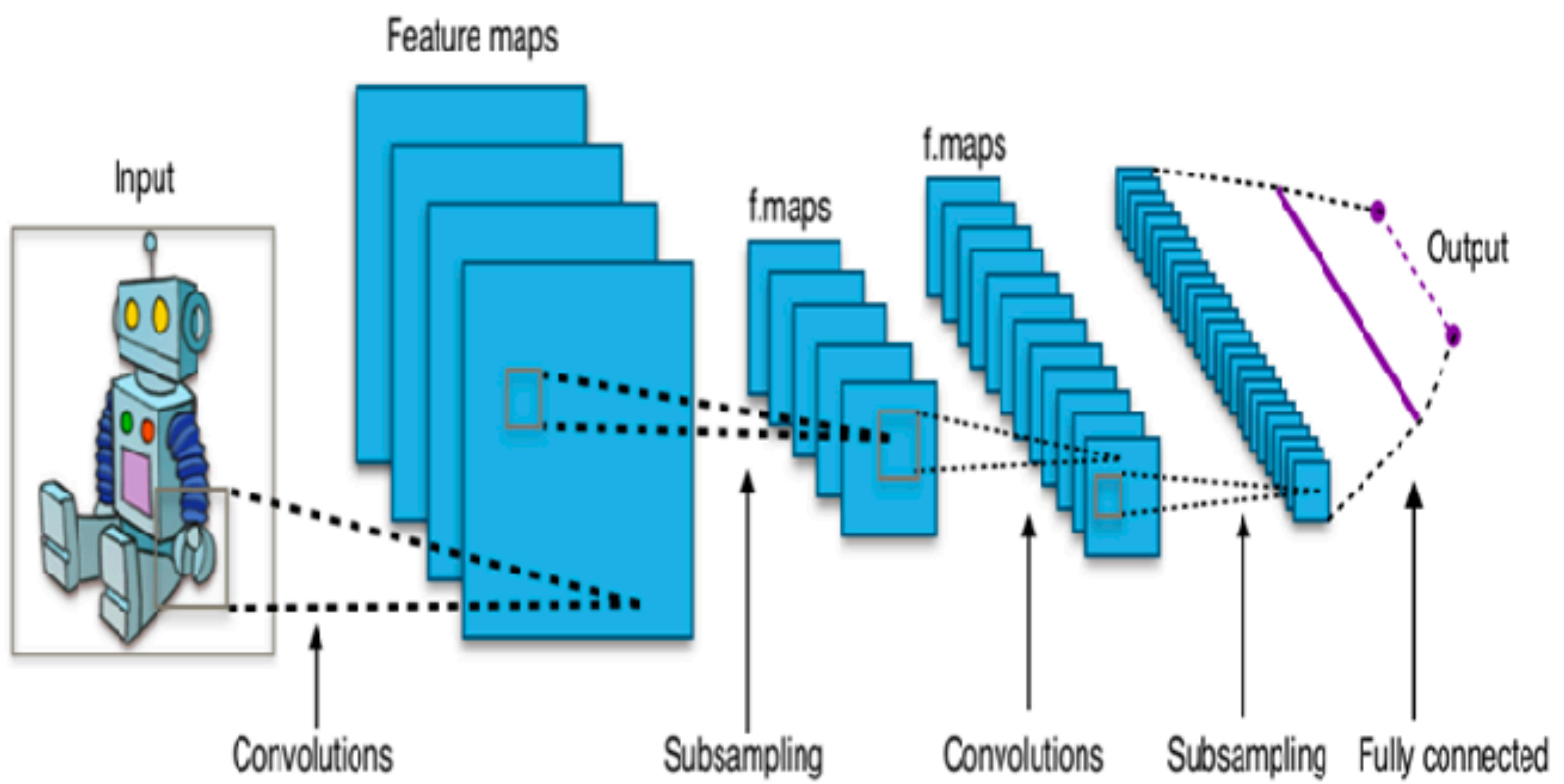


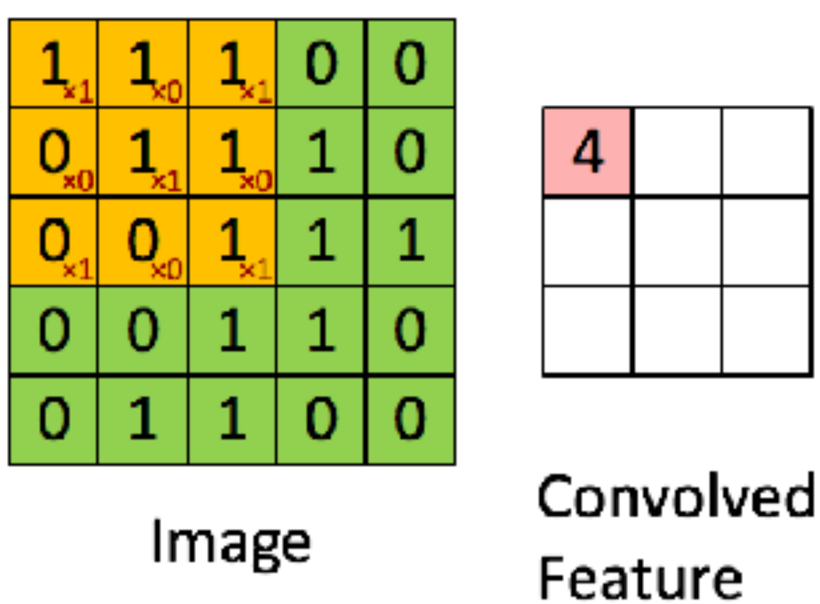
Activation function
(sigmoid, tanh, ReLU, …)

During training:

Backpropagate error & update weights using gradient descent

CONVOLUTIONAL NEURAL NETWORKS



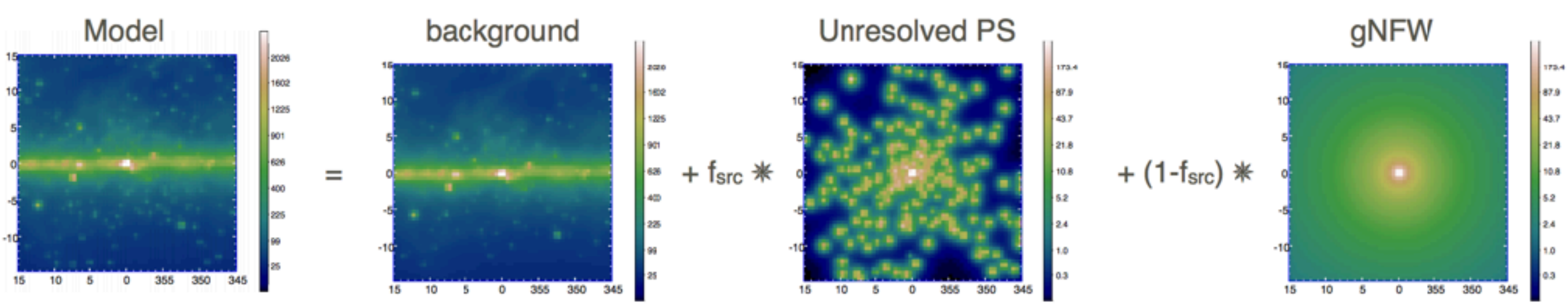


- Weights are fixed per kernel
  - Translational invariance
  - Greatly reduced weight parameter space (=faster training convergence / more accuracy / less data necessary)

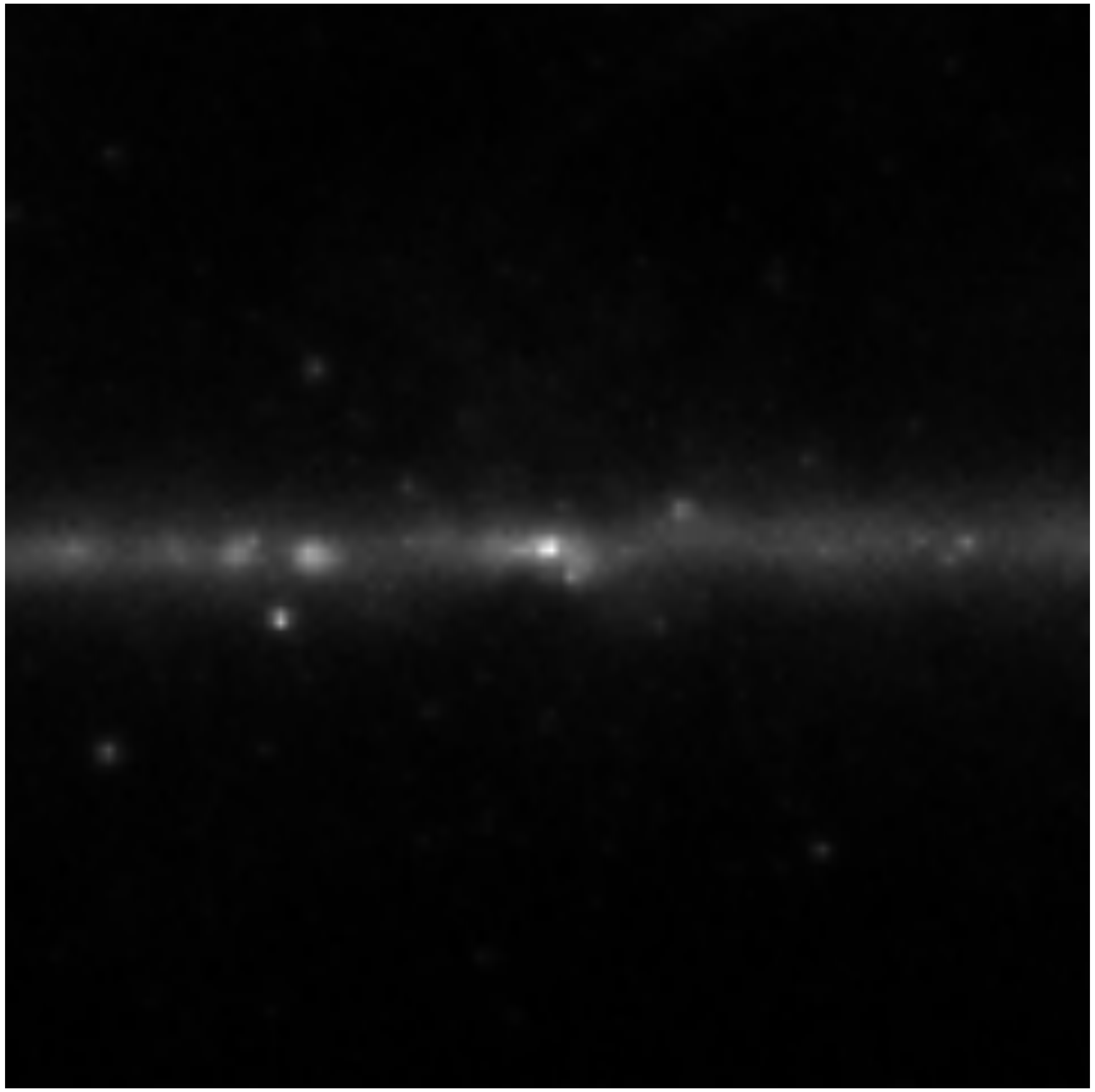
HOW TO APPLY THIS TO γ-RAY DATA?

- Goal: determine the component of point sources vs diffuse source of the GC excess – f_{src}
- **Proof of concept: arXiv:1708.06706**
- Difficulty here: there is only 1 image of the GC, in contrast to galaxy classification for example
- Simulate GC using Fermi tools (3 parameters (f_{src}, BG model used, unresolved PS flux distribution))
  - Output is photon count map of photons between 1-6GeV
    (no spectrum information, will be improved in new version)
- Generate training + validation data from simulations
  - Sample the same point multiple time because of randomisations
    (like point source coordinates & added noise)
- Train network to predict f_{src} accurately in all scenarios of the other components
- Apply to real image to get prediction of f_{src}

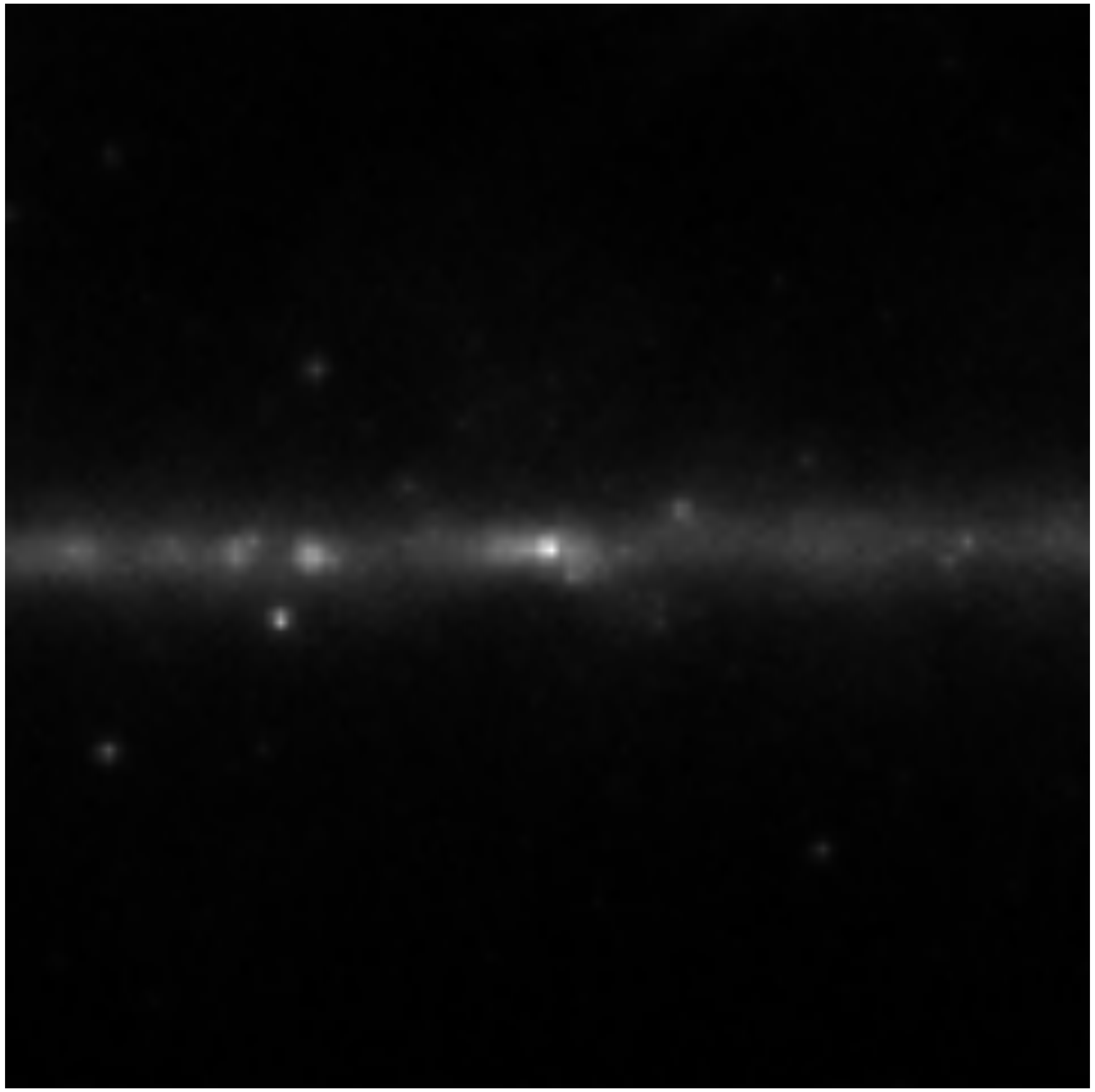
GC Excess



EXAMPLE OF TWO SIMULATIONS

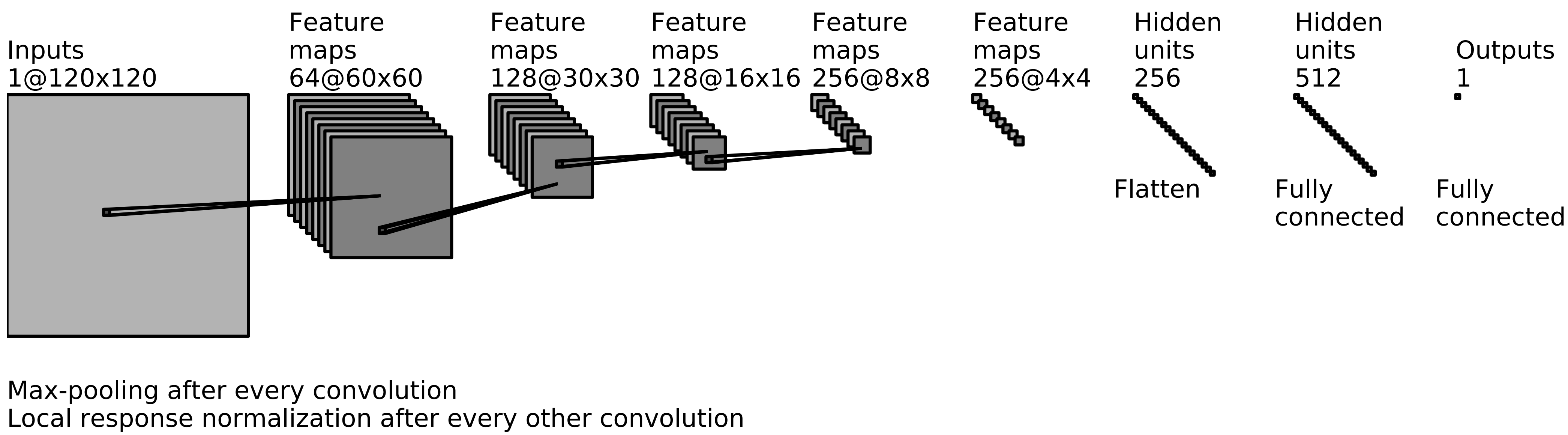


$f_{src} = 0.9883$



$f_{src} = 0.0275$

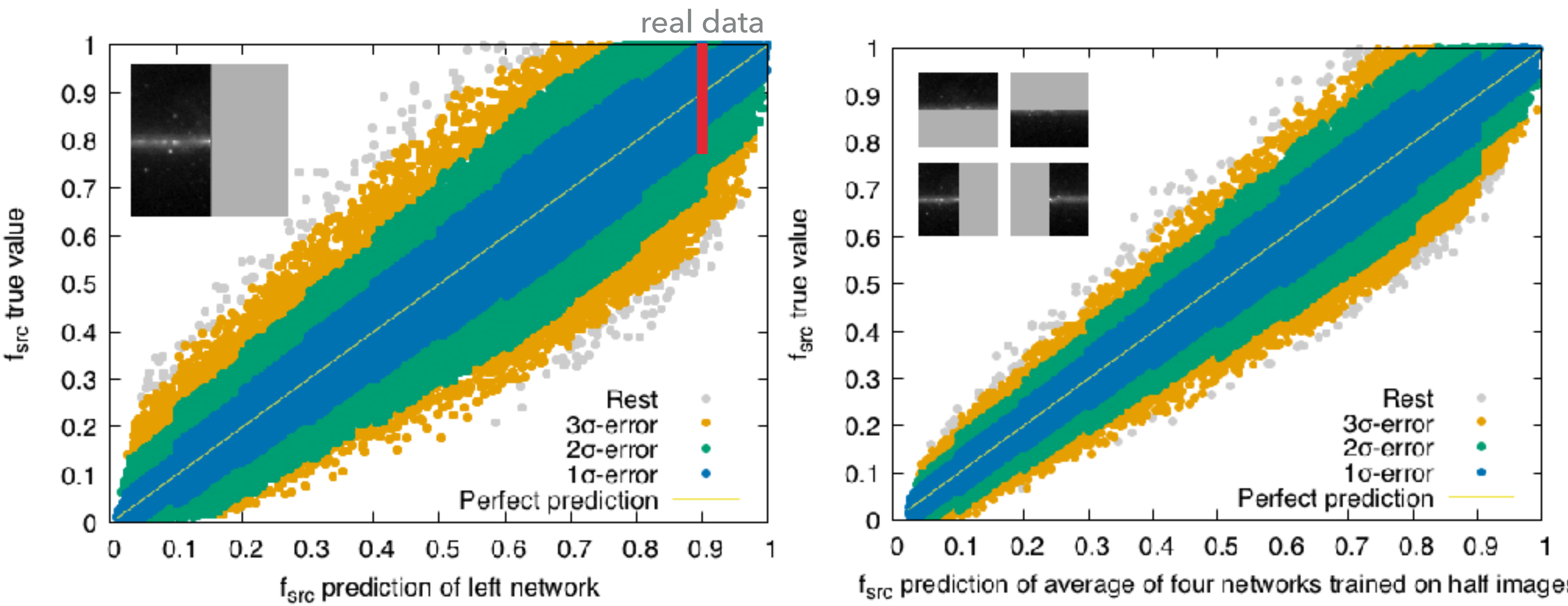
CONVOLUTIONAL NEURAL NETWORK



- Every layer has L2 regularisation (penalise high weights to prevent overfitting)
- 1.2 million images of 120x120 values
- ~10 million internal parameters
- 1 day to train each network (TensorFlow, 2x GTX1080, >5000 cores, ~16 TFLOPs)

RESULTS

- Train using 3 background models, test on 2 others
- Test data: 2x30000 test points



Because we are doing a followup study, real result is only evaluated on the left network to not bias ourselves

CAVEATS

- This was a proof of concept, because of simulation limitations:
- Reality may be outside the sampling space for generating images:
  - The range of flux distribution is not large enough
  - The background models used are probably too similar
  - The γ parameter for the gNFW profile was fixed
  - –> Accuracy is hard to define if the network needs to extrapolate

CAVEATS

- This was a proof of concept, because of simulation limitations:
- Reality may be outside the sampling space for generating images:
    - The range of flux distribution is not large enough
    - The background models used are probably too similar
    - The γ parameter for the gNFW profile was fixed
    - –> Accuracy is hard to define if the network needs to extrapolate
- Only returns f_{src}, no prediction on other simulation parameters
    - f_{src} is not known, so no way to do sanity-check on known parameters
- Input data has no distinction between 1 or 6 GeV – data can be enriched by adding energy bins

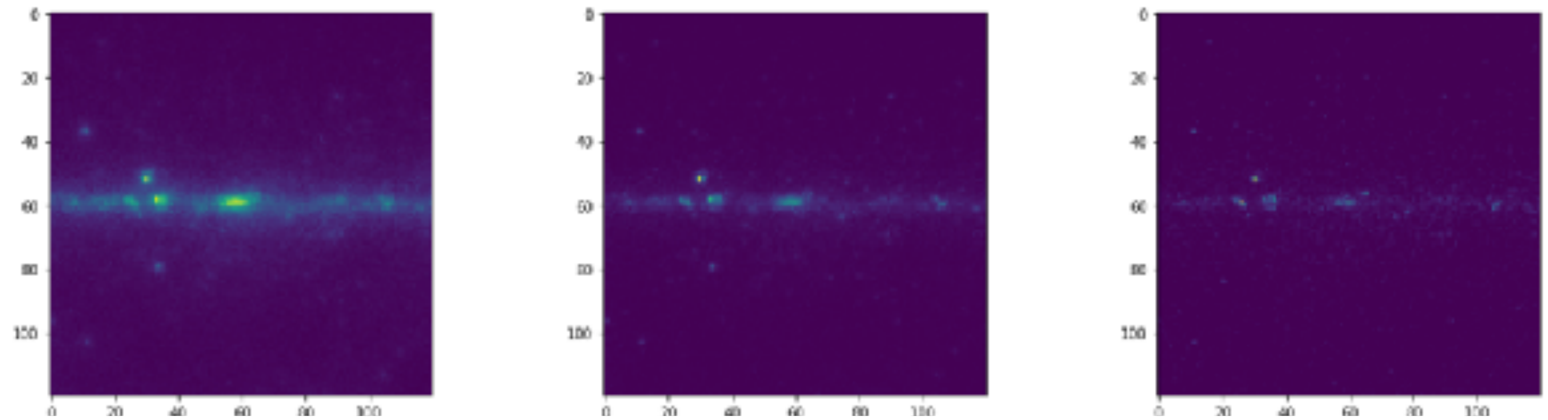
FERMIAI V2

- We attempt to address all these issues in a new analysis:
  - Instead of using 5 fixed background models, randomise background model generation – much larger input parameter space
  - Add 3 energy bins for the photon counts to the data instead of 1
  - Update the ranges of the parameters to wider values during dataset creation
  - Predict not only f_{src}, but all (16) input parameters

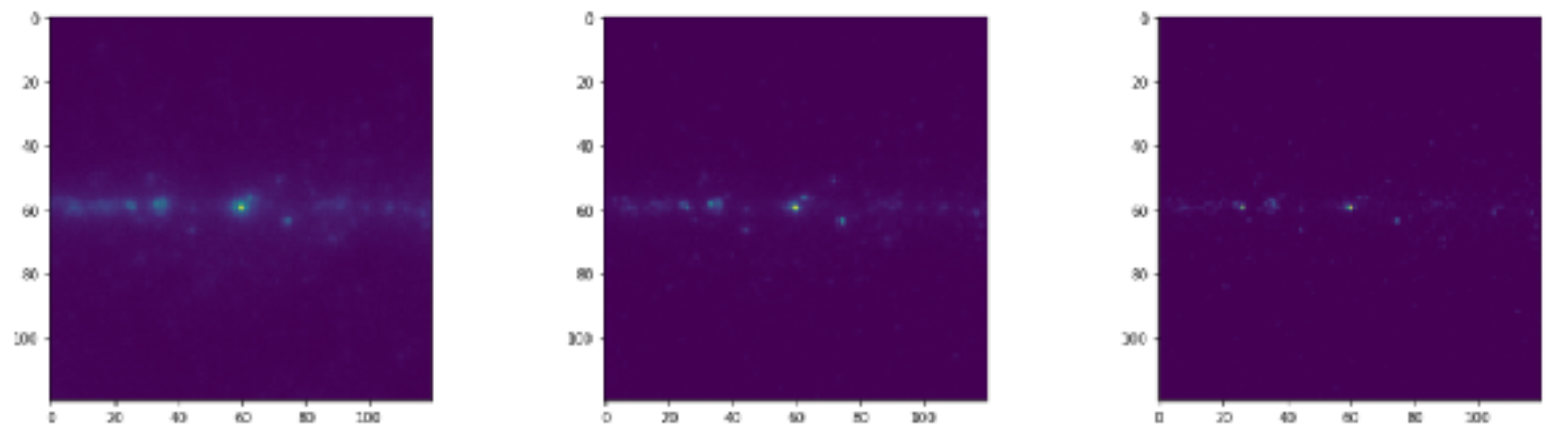
FERMIAI V2

- Example simulations

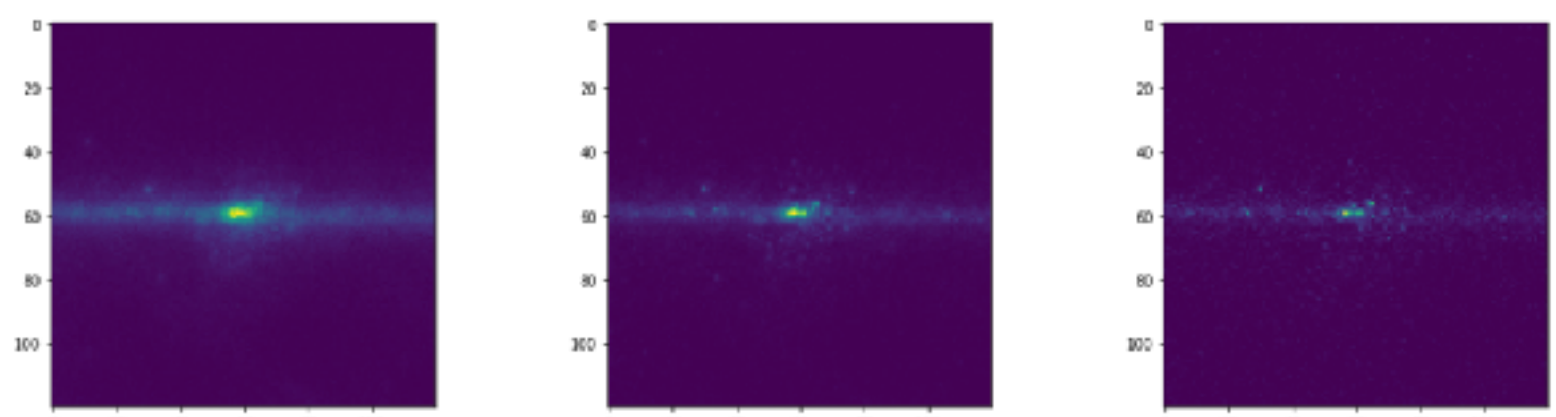
```
Example  4
Y [2.0110815  0.0499763  0.22974858 0.25071636 0.20901702 0.45731387
 1.15165    2.7110274  0.8225395  1.4593923  0.2322307  0.83295983
 0.91459596 1.0291971  0.57494676 1.05      ]
```




```
Example  5
Y [0.20034094 0.37943304 0.0232188  1.6287282  0.34669346 0.68575513
 1.5606146  1.4767014  0.08627474 2.632246   1.6390889  0.6207119
 0.4194314  1.3014951  0.05485215 1.14      ]
```

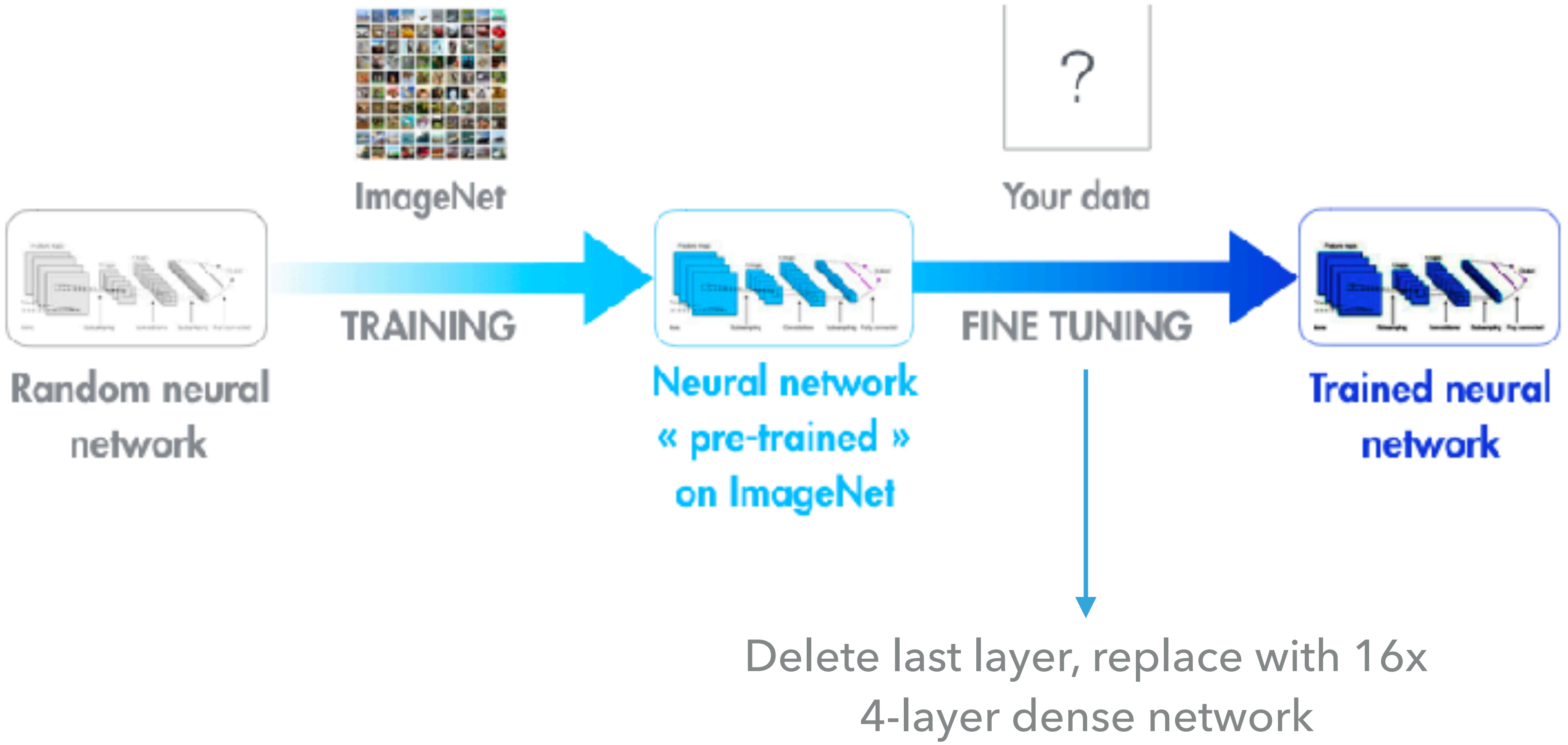



```
Example  6
Y [2.5529957  0.49496025 1.7866129  0.644916   0.18359113 1.0066106
 0.6741637  2.2655401  0.23621954 0.48079348 0.19954418 1.89356
 2.57494    1.5297246  0.18912439 1.07      ]
```



FERMIAI V2

- Problem is much more difficult (16D regression instead of 1D), use a different technique: **transfer learning**

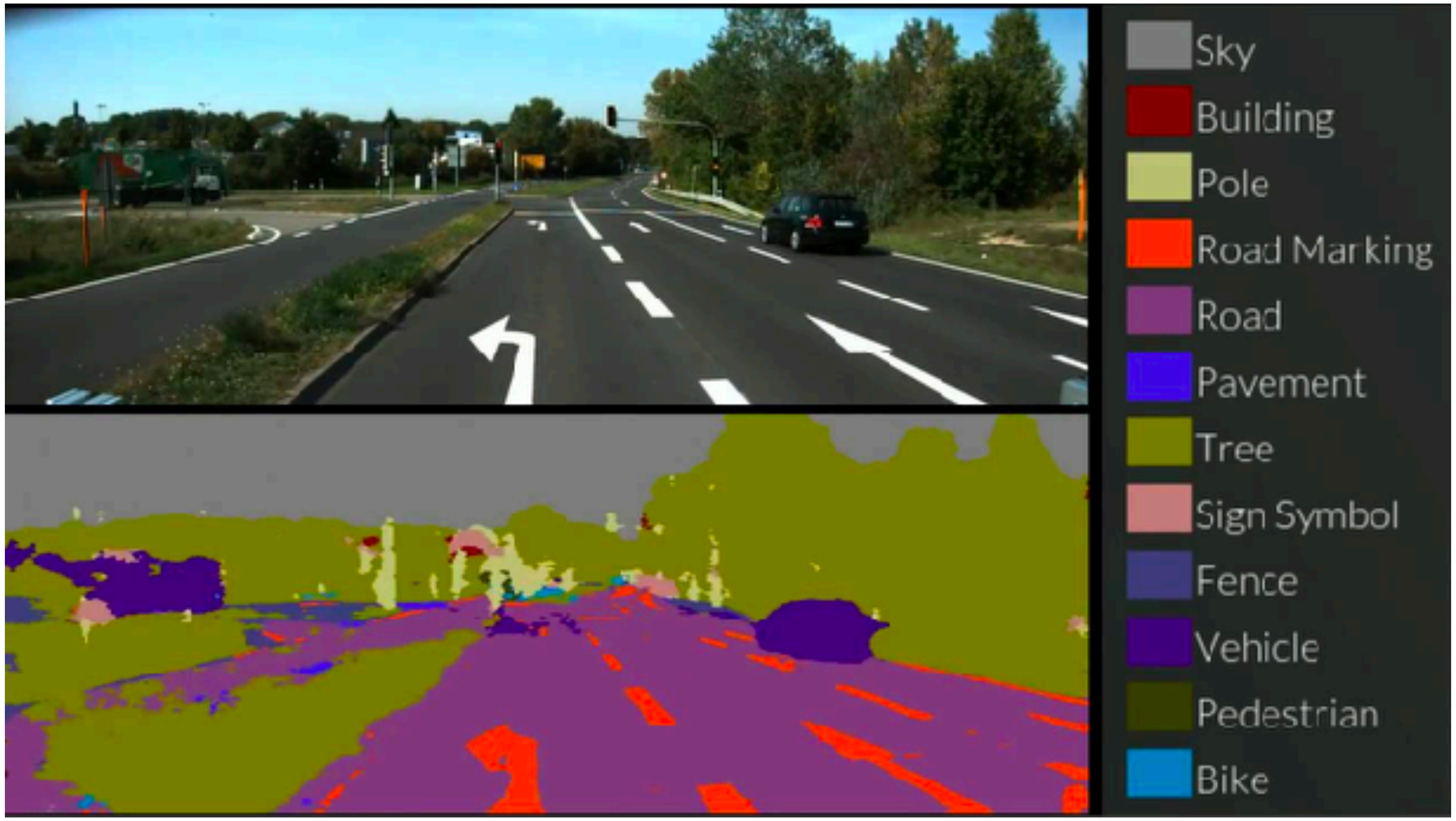


FERMIAI V2

- Seems to work extremely well
    - Even though the parent network (trained on imagenet) is completely unrelated to γ-rays
    - But: if random weight initialisation (default), every ConvNet needs to learn the concept of “lines”, “squares” etc.
    - Like you learn a baby Italian, it’s easier if the baby first learns another language and then Italian. Even if it’s Chinese.

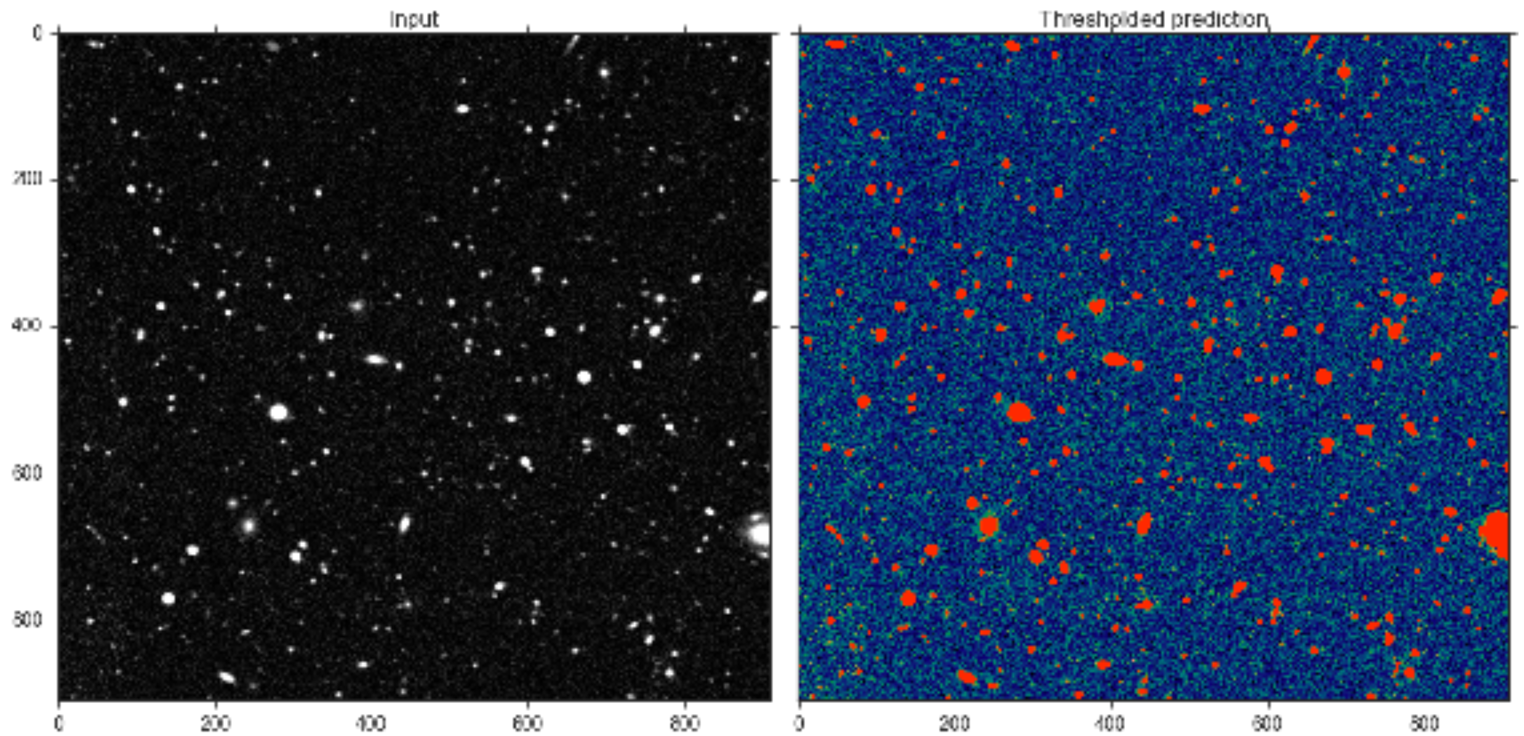
POINT SOURCE CLASSIFICATION

- Another topic where ML can be very useful is in the localising and classification of point sources
- Copy from eg self-driving car research (segmentation maps):



POINT SOURCE CLASSIFICATION

- Label every pixel as "point source" or "background"
- Or even label every pixel as "pulsar", "blazar", …, or "background"
- Catalog generator
- U-nets, from the tensorflow example page: segmentation of galaxies

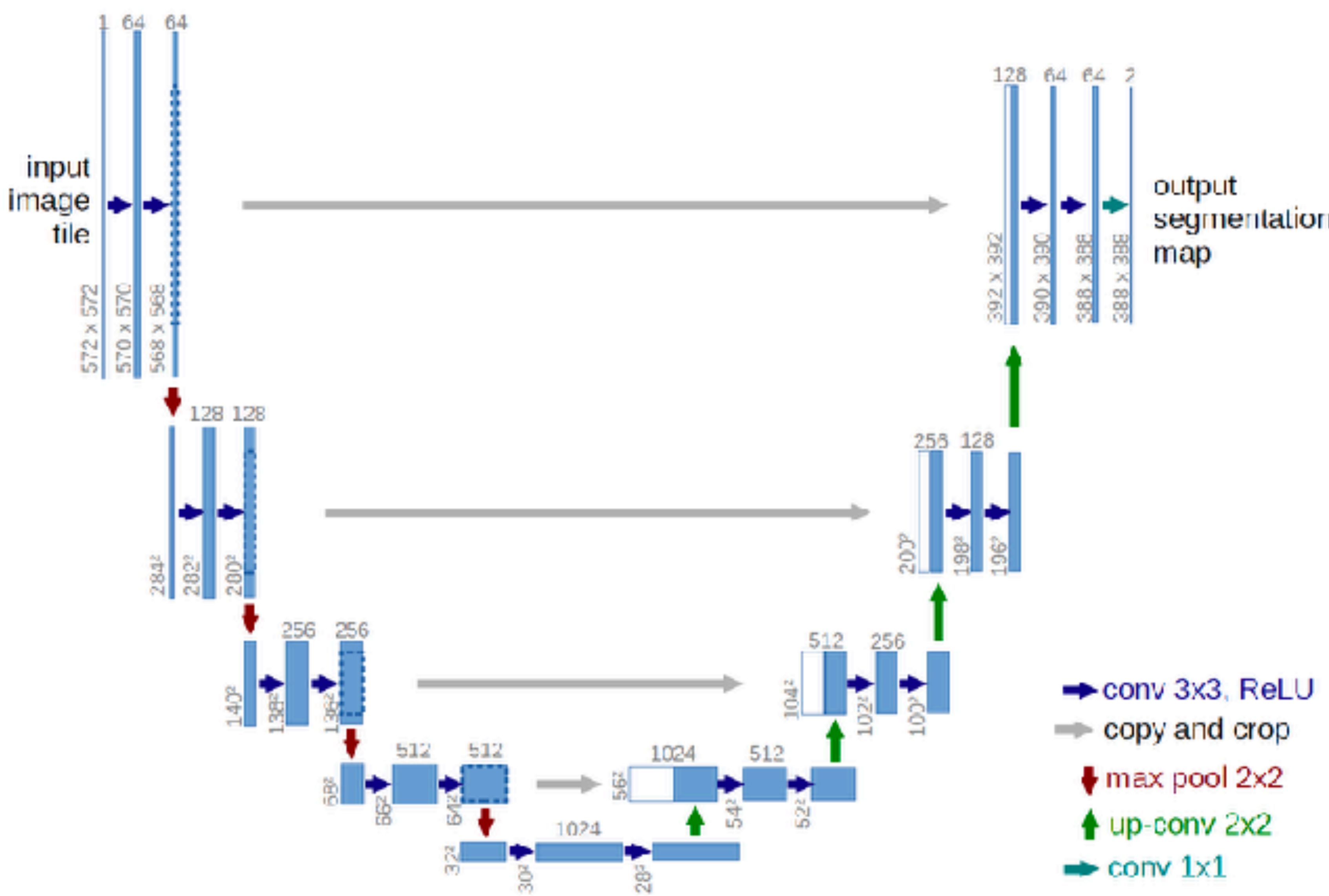


POINT SOURCE CLASSIFICATION

>2000 citations since 2015!

- U-nets

U-Net - Computer Vision Group, Freiburg
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/ ▾ Vertaal deze pagina
door O Ronneberger - Geciteerd door 2422 - Verwante artikelen
U-Net: Convolutional Networks for Biomedical Image Segmentation. The u-net is convolutional network architecture for fast and precise segmentation of images.

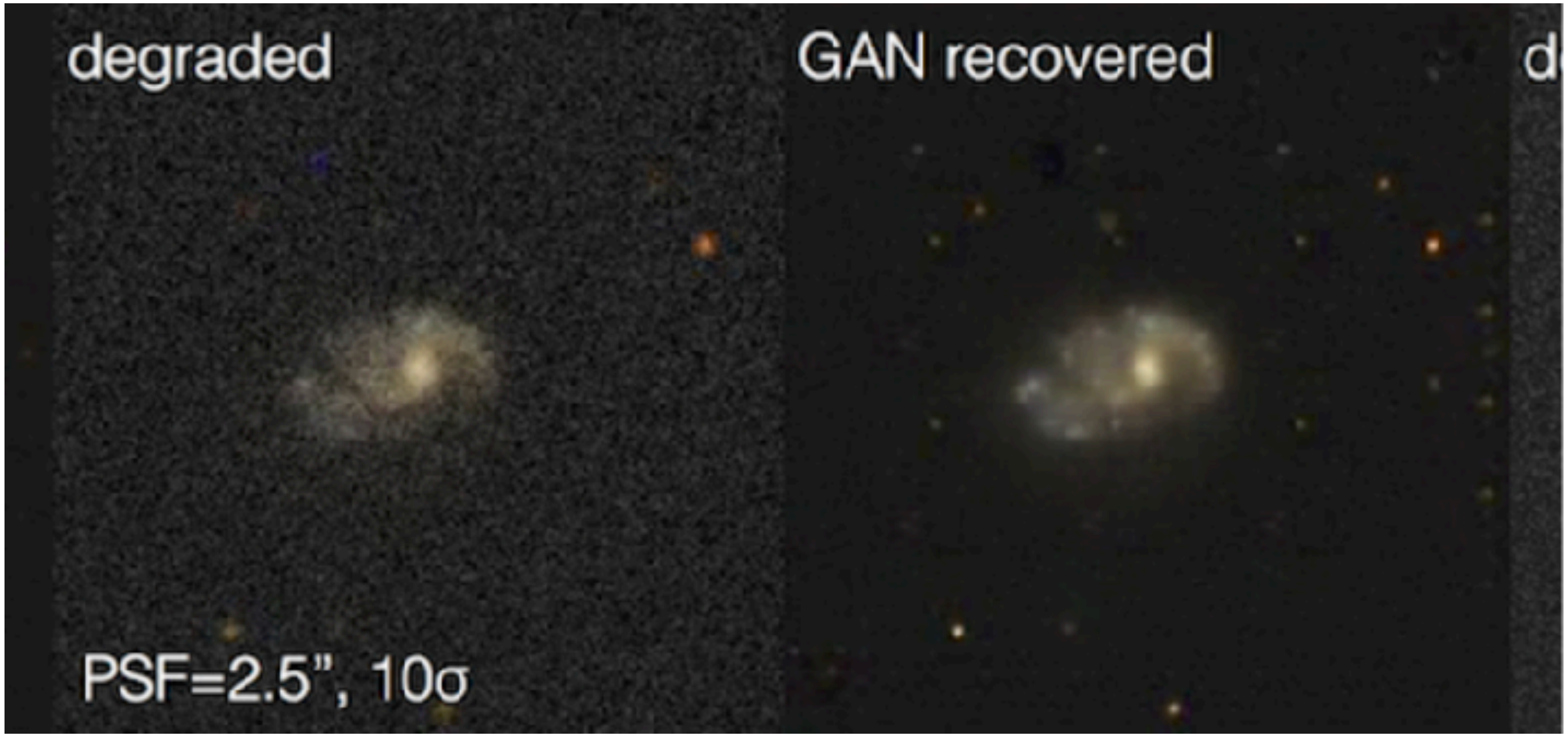


GANS

- GANs can learn to generate things. You know the faces example

GANS

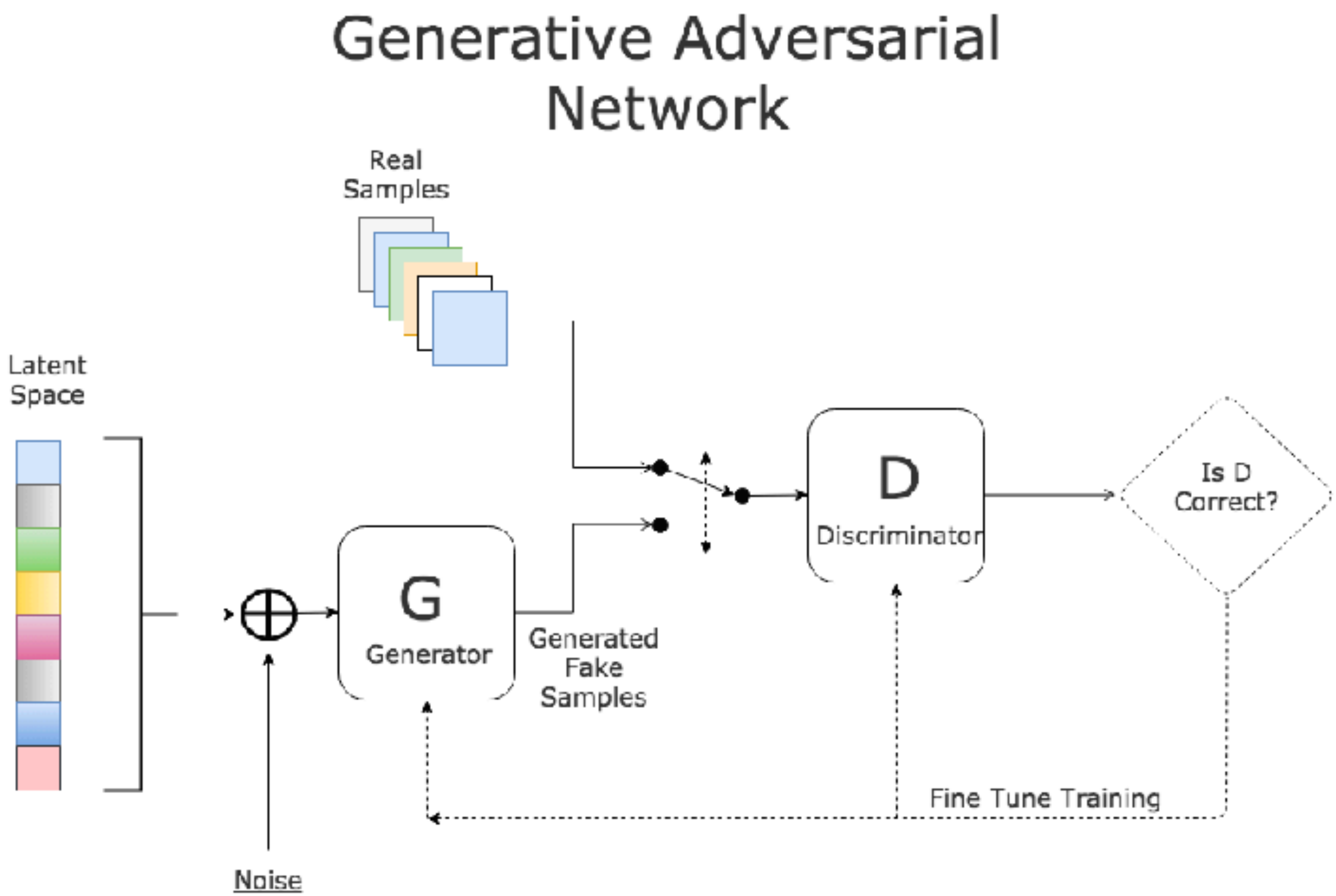
- Already used in denoising astro images
- Could be used to greatly speed up background model generation
- Or any other type of simulation (output doesn't have to be images)



DARKMACHINES

- FermiAI2 still under construction, but seems a feasible approach with good outlook
- Many use cases for ML
- Interested in applications of machine learning in the field of dark matter?
- www.darkmachines.org
  Inclusive analysis of Fermi-LAT point sources
  (but also gravitional lensing / collider experiments / …)
- ~100 researchers in field of DM/ML working together
- 8-12 April workshop in ICTP, Trieste, Italy

GANS


Generative Adversarial Network
Real Samples
Latent Space
G
Generator
Generated Fake Samples
D
Discriminator
Is D Correct?
Fine Tune Training
Noise