Contrastive Learning for Robust Representations of Neutrino Data

Alex Wilkinson, Radi Radev, Saul Alonso-Monsalve

Neutrino Physics and Machine Learning 2024

27 June 2024

ML with a Data-MC Discrepancy

- HEP simulation provides detailed training data for ML
- But the simulation will never be perfect, ultimately data and MC will come from different distributions due to mismodelling
- Some unknowns in the model will be parametrised

  $\rightarrow$ Design models that are invariant to these parameters where possible

  $\rightarrow$ Characterise dependence on these parameters where it's not
- There will be some ”unknown unknowns”

  $\rightarrow$ Can still look for ways data can be used to mitigate the effect of this on our models

Can recent advances in computer vision help address these problems?

Self-Supervised Learning in Vision

- Computer vision tasks often have a large quantity of unlabelled data and much fewer labelled samples
- Self-supervised learning methods are used to train a model on the unlabelled data
  → These model can then be finetuned using the small labelled sample

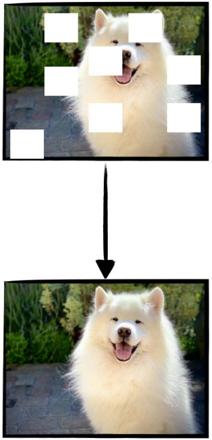


Illustration of MAE - vision foundation model

Mitigating Bias with Contrastive Learning

- We try to adapt vision's self-supervised paradigm to mitigate the effect of a data-MC discrepancy in neutrino physics
- Look at how a pretraining stage using contrastive learning can be used to generate representations of neutrino data that are robust to mismodelling of the detector simulation
- Our method is based on the SimCLR framework

A Simple Framework for Contrastive Learning of Visual Representations

---

Ting Chen [1] **Simon Kornblith** [1] **Mohammad Norouzi** [1] **Geoffrey Hinton** [1]

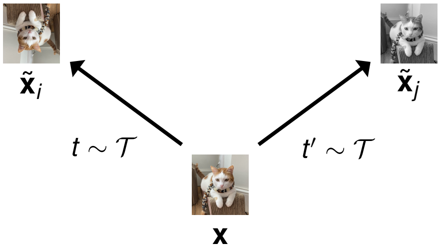
Contrastive Learning Recipe

- Ingredients:

$\mathbf{x}$

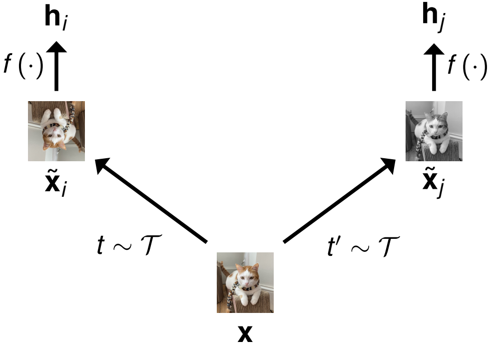
Contrastive Learning Recipe

- Ingredients:
  - $\rightarrow$ A set of augmentations $\mathcal{T}$



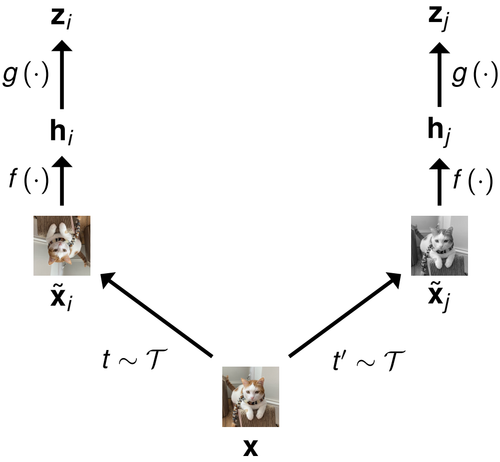
Contrastive Learning Recipe

- Ingredients:
  - $\rightarrow$ A set of augmentations $\mathcal{T}$
  - $\rightarrow$ An encoder network $f(\cdot)$ - this extracts the representation $\mathbf{h}$ we will use for downstream tasks



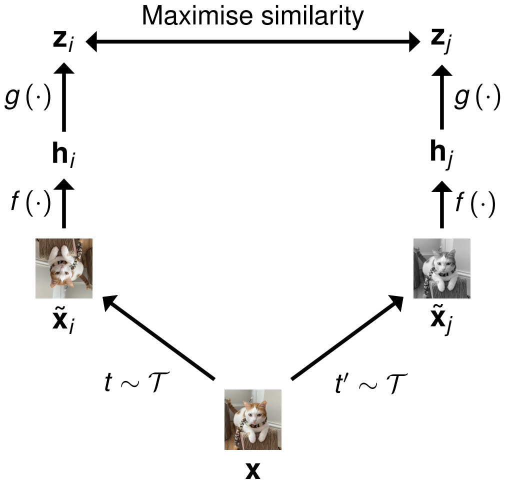
Contrastive Learning Recipe

- Ingredients:
  - → A set of augmentations $\mathcal{T}$
  - → An encoder network $f(\cdot)$ - this extracts the representation $\mathbf{h}$ we will use for downstream tasks
  - → A projection head $g(\cdot)$ - an MLP to map the representations to the space where a contrastive loss is applied



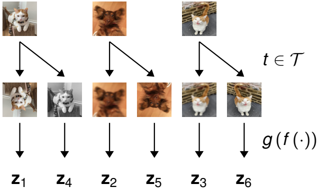
Contrastive Learning Recipe

- Ingredients:
  - → A set of augmentations $\mathcal{T}$
  - → An encoder network $f(\cdot)$ - this extracts the representation $\mathbf{h}$ we will use for downstream tasks
  - → A projection head $g(\cdot)$ - an MLP to map the representations to the space where a contrastive loss is applied
  - → A contrastive loss function



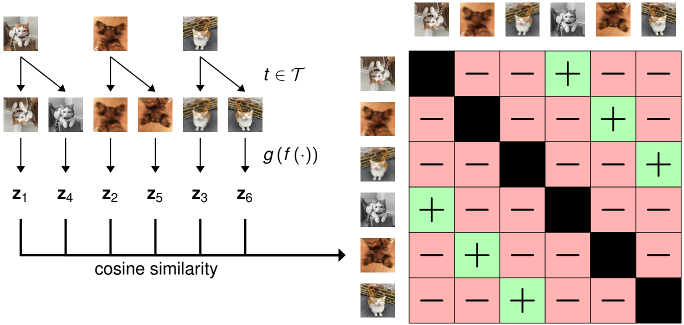
A Contrastive Loss

- In a minibatch of N unique examples, each gets augmented twice generating for each example, one positive pair and $2\,(N-1)$ negative pairs



A Contrastive Loss

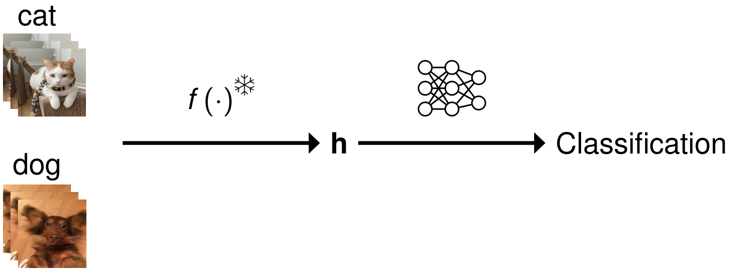
- In a minibatch of N unique examples, each gets augmented twice generating for each example, one positive pair and $2\,(N-1)$ negative pairs
- A contrastive loss is constructed that maximises the cosine similarity of positive pairs and minimises that of negative pairs



$$\mathcal{L} = \frac{1}{N_{row}} \sum_{i \in rows} -\log\left(\frac{\exp\left(\mathrm{sim}\left(\mathbf{z}_i, \mathbf{z}_+\right)/\tau\right)}{\sum_{k \in -,+} \exp\left(\mathrm{sim}\left(\mathbf{z}_i, \mathbf{z}_k\right)/\tau\right)}\right)$$

Using the Representation

- This self-supervised training provides a powerful representation of the data through the frozen weights of the encoder network: $f(\cdot)$ ❄
- We can use this representation for downstream tasks by finetuning with labelled data



Applications to Neutrino Experiments

Can we learn representations that are invariant to known detector systematics?

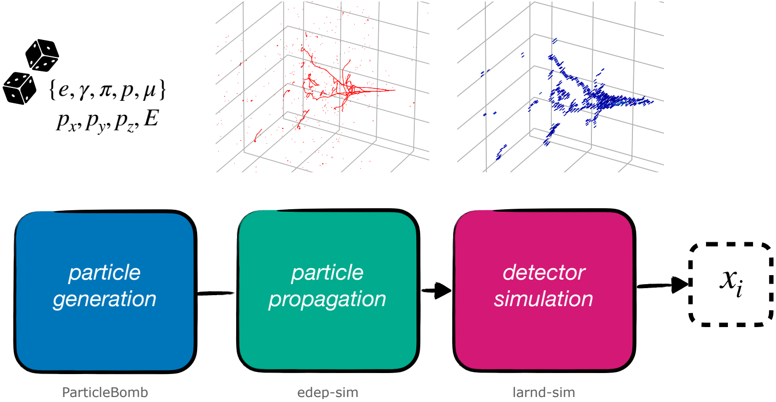
Use throws of detector systematics as augmentations for the contrastive learning

Can we use unlabelled data to learn representations that are invariant to ”unknown unknowns” caused by mismodelling of the detector simulation?

Use data in the contrastive learning stage and labelled simulation for the finetuning

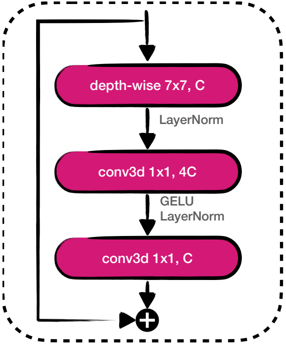
ND-LAr Simulation

- We use single particle classification of DUNE ND-LAr simulation to study these questions

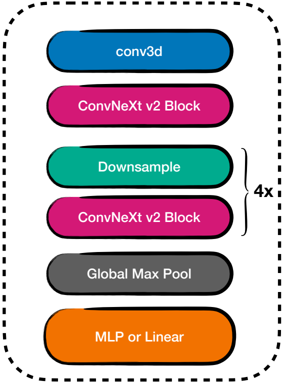


Contrastive Model

- ND-LAr input is extremely sparse, models made using the sparse library MinkowskiEngine
- Encoder network is a sparse submanifold CNN based on the ConvNeXt v2 architecture
- The learned representation, **h**, has dimension 768

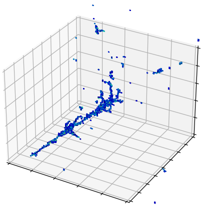


Modified ConvNeXt Block

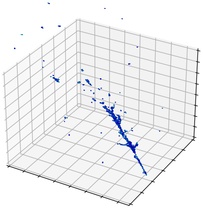


Full architecture

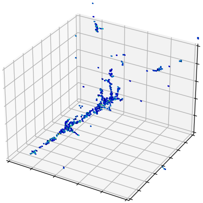
Augmentations



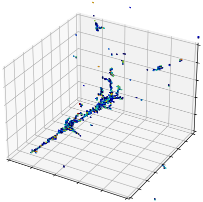
Identity



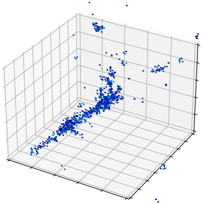
Rotate



Drop



Noise



Wiggle

Apply composition of
3 selected at random

Training for Invariance to Systematics

Can we learn representations that are invariant to known detector systematics?

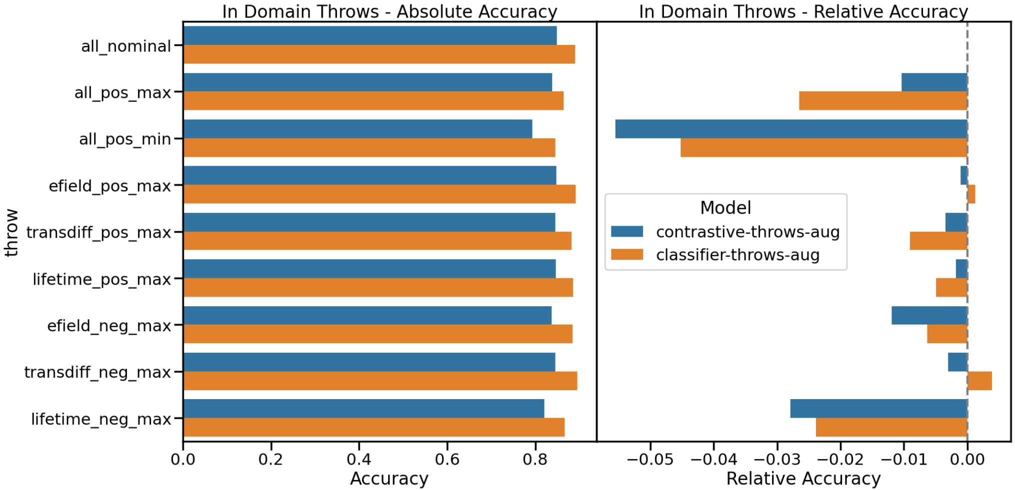
- Use systematic throws as augmentations for the contrastive learning

  $\rightarrow$ Compare to using these throws as augmentations in training of a classifier

- Look at uncertainties in LAr properties
- Simulate each sample with 10 random throws to make a set of augmentations

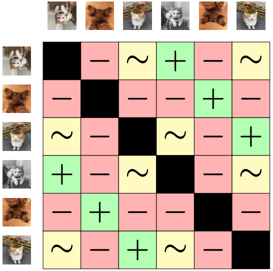
E Field	[0.45, 0.55] kV/cm
Transverse Diffusion	[4e-6, 14e-6] cm$^2/\mu$s
Electron Lifetime	[500, 5000] μs

Results - Accuracies



Negative Result!

- Some things that could improve the contrastive learning
  - → Stronger detector throws
  - → Incorporate class information in the contrastive loss function



- The bottom line: Known detector systematics should be included as augmentations in training, contrastive learning approaches probably wont help here

Contrastive Domain Adaptation

Can we use unlabelled data to learn representations that are invariant to ”unknown unknowns” caused by mismodelling of the detector simulation?

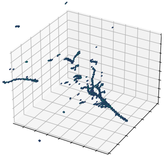
- Try using unlabelled data in the contrastive stage to learn a strong representation from the data rather than the simulation distribution
  - → Pretrain on unlabelled data
  - → Finetune the learned representation using labelled simulation
- By pretraining on the correct distribution we hope to mitigate the risk our model being sensitive to the effects of mismodelling
- Using the same encoder network architecture, compare with:
  - → Classifier trained with the same augmentations used in the contrastive stage
  - → Domain-adversarial neural network (DANN) — tries to enforce domain-invariant features by classifying the domain as well as the label

Electronics Throws

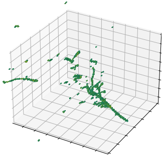
- Labelled simulation $\rightarrow$ Nominal
- Unlabelled data $\rightarrow$ Throws

Parameter	Throw 1σ
Gain	2%
Buffer Risetime	10%
Common-mode Voltage	2%
Reference Voltage	2%
Pedestal Voltage	20%
Reset Noise	10%
Uncorrelated Noise	10%
Discriminator Noise	10%
Discriminator Threshold	2%

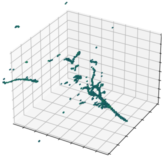
Nominal



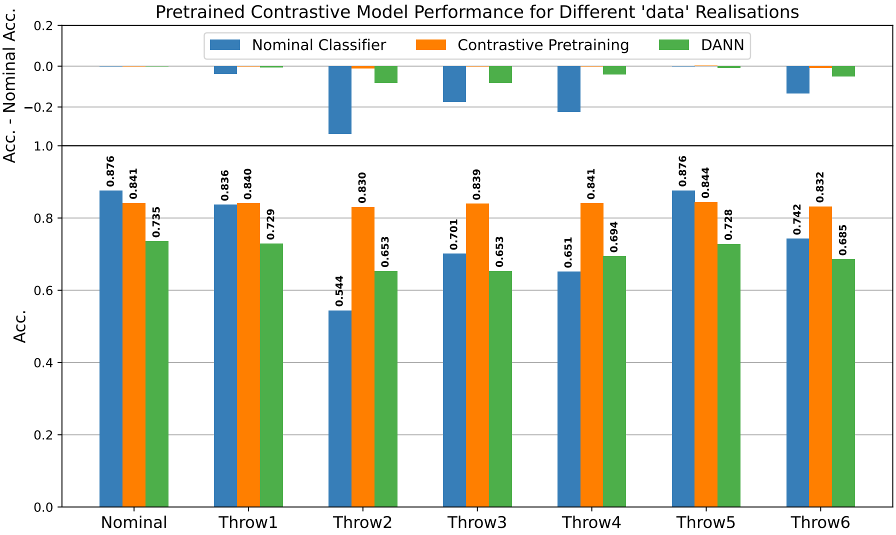
Throw 1



Throw 2



Results - Accuracies

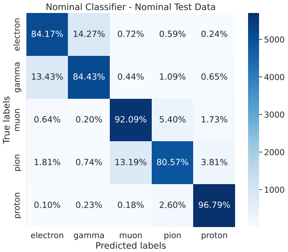


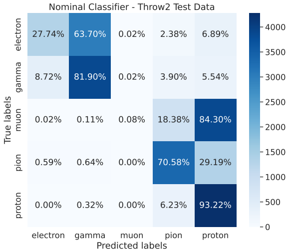
Results - Confusion Matrices

Data Domain is Nominal

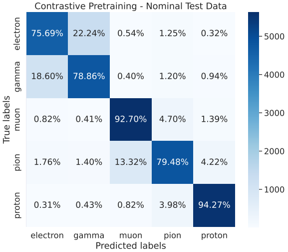
Data Domain is Throw2

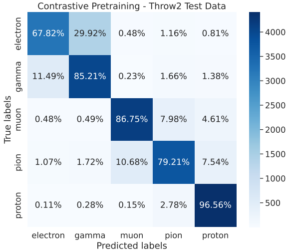
Nominal Classifier





Contrastive Pretraining





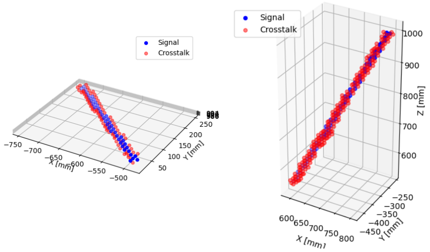
Future Work

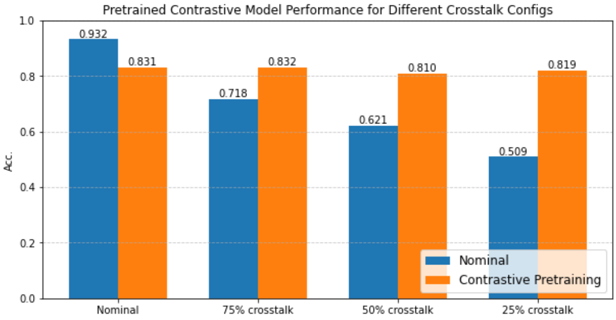
- Improvements to contrastive model to make nominal accuracy more competitive with classifier
  - $\rightarrow$ Different representation shapes and sizes
  - $\rightarrow$ Incorporate class information into contrastive training
- Testing with different systematics and on different tasks
- Applying to new datasets and detector technology…

Segmented Scintillator Cube Detector

- Starting work on applying contrastive learning to simulation of a magnetised plastic scintillator detector made of optically-isolated cubes (zenodo)

  → The simulation includes crosstalk — we vary this to study mismodelling of the detector



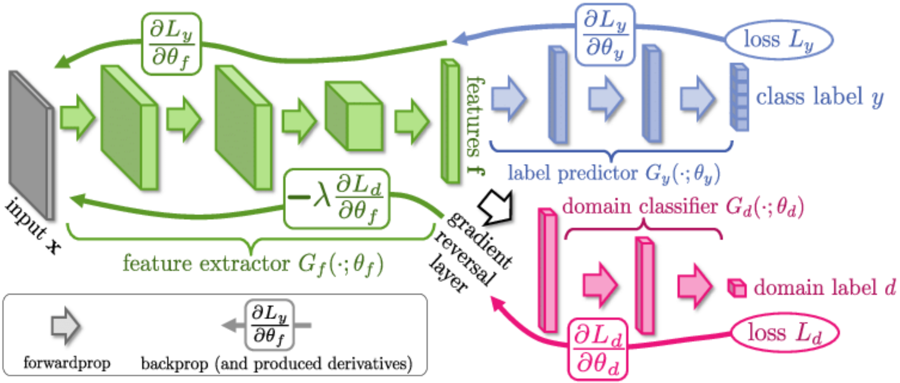


Summary

- Developed a contrastive learning framework for tackling data-MC discrepancy in neutrino physics
- Studied the framework by varying the detector simulation for single-particle LArTPC data
  - $\rightarrow$ Promising results when applied to domain adaptation
  - $\rightarrow$ Less promising when applied to learning explicit detector systematics
- Potential for contrastive learning to mitigate data-MC discrepancy is demonstrated — we need to do more studies to know!
  - $\rightarrow$ Improve discriminative power of the representation
  - $\rightarrow$ Look at new datasets and systematics

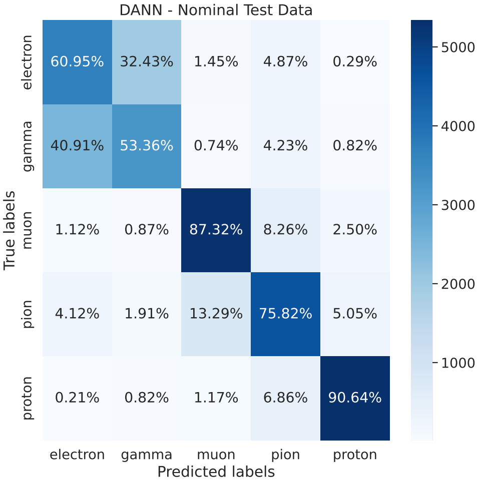
Backup

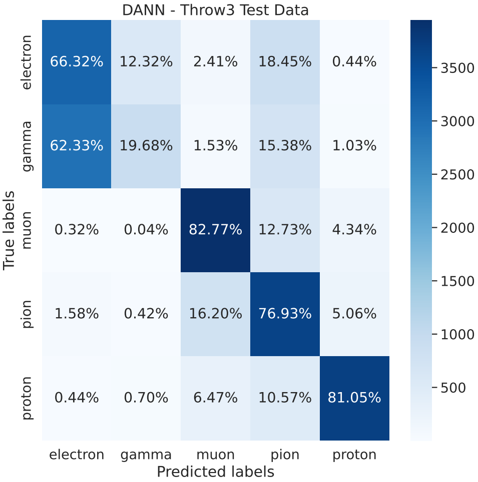
DANN



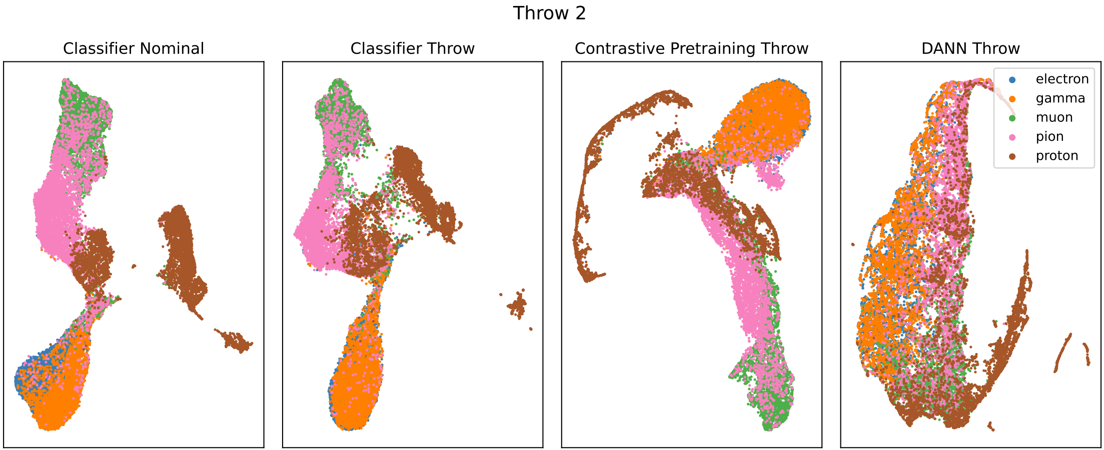
$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
features $\mathbf{f}$
class label y
label predictor $G_y(\cdot;\theta_y)$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
input x
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
feature extractor $G_f(\cdot;\theta_f)$
loss L_d
$\frac{\partial L_d}{\partial \theta_d}$
$\frac{\partial L_y}{\partial \theta_f}$
forwardprop
backprop (and produced derivatives)

DANN Confusion Matrices





Some UMAPS


Throw 2
Classifier Nominal
Classifier Throw
Contrastive Pretraining Throw
DANN Throw
electron
gamma
muon
pion
proton