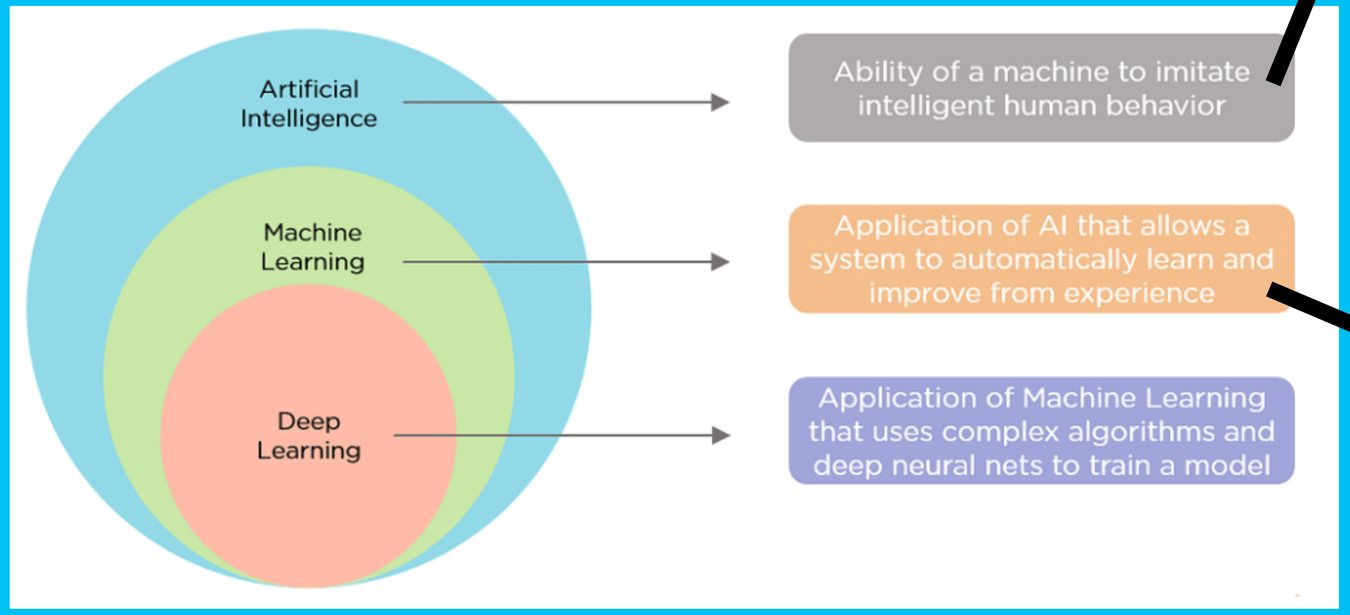
ML-REDSHIFTS

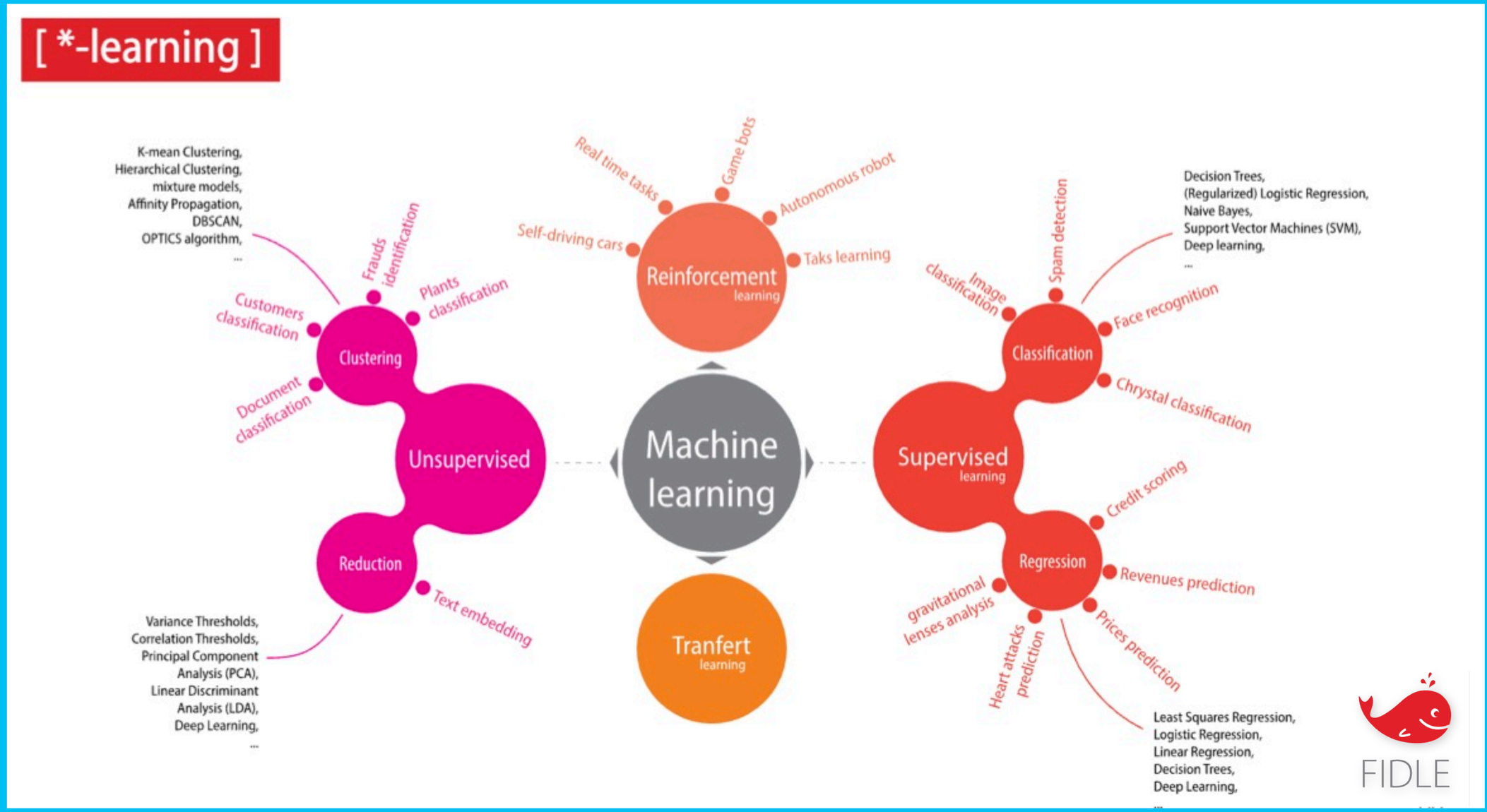


Non-ML AI methods operate on a set of instructions and logical constructs provided by human programmers (e.g. early email spam filters, original core of Google search). They don’t “learn".

Learn from data. Paradigm shift for photo-z: don't model the physics, learn the complex non linear mapping between observations and redshift directly from data

MARIE TREYER,
WITH GREAT HELP FROM
FIDLE.CNRS.FR AND AI...

ML ZOO


[*-learning]
Machine learning
Supervised learning
Classification
Regression
Unsupervised
Clustering
Reduction
Reinforcement learning
Tranfert learning
Spam detection
Image classification
Face recognition
Chrystal classification
Credit scoring
Revenues prediction
Prices prediction
gravitational lenses analysis
Heart attacks prediction
Decision Trees,
(Regularized) Logistic Regression,
Naive Bayes,
Support Vector Machines (SVM),
Deep learning,
...
Least Squares Regression,
Logistic Regression,
Linear Regression,
Decision Trees,
Deep Learning,
Frauds identification
Plants classification
Customers classification
Document classification
Text embedding
K-mean Clustering,
Hierarchical Clustering,
mixture models,
Affinity Propagation,
DBSCAN,
OPTICS algorithm,
...
Variance Thresholds,
Correlation Thresholds,
Principal Component
Analysis (PCA),
Linear Discriminant
Analysis (LDA),
Deep Learning,
...
Real time tasks
Game bots
Autonomous robot
Self-driving cars
Taks learning
FIDLE

NOMENCLATURE

machine learning (ML) : not a neural network

neural networks (NN) : mimic learning neurons

deep learning (DL) : a multi-layered NN

training set : data to learn from

labels : the true values used to train a model (e.g. spec-z)

features : fundamental units of information

supervised model : learns from a labeled training set

unsupervised : no notion of a "right answer" or a prediction task

self-supervised : a “pretext task” forces the model to learn meaningful representations of the data

SUPERVISED VS UNSUPERVISED

Photometric redshift estimation is a **supervised task**: we need known spectroscopic redshifts **(labels)** to train a model to map photometry to redshift. No method is 100% label-free.

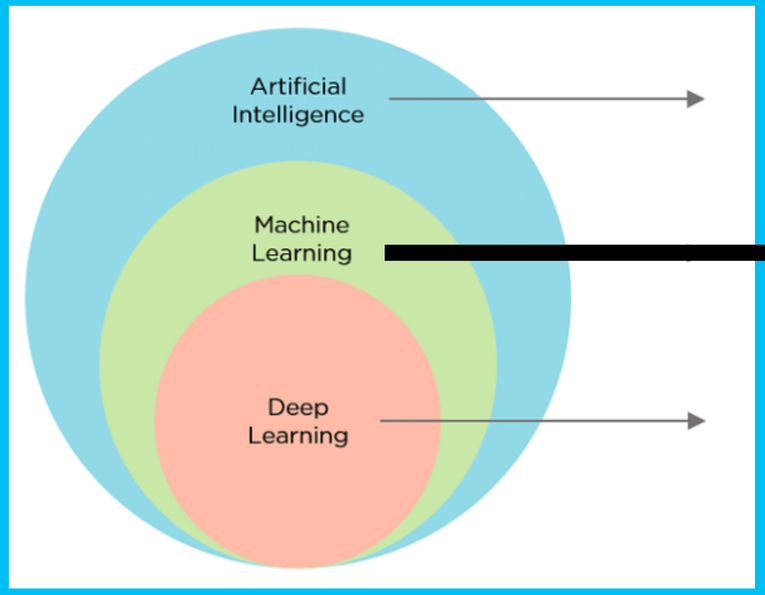
But there are un/self/semi-supervised methods that can reduce the amount of labeled data required or leverage physical priors to supplement labels.

TRAINING SET FOR PHOTO-Z

A representative sample of galaxies with known redshifts (the labels) and multi-band photometry (the input data)

N.B. : A ML model is only as good as its training data!

Trained models can't generally be trusted in regions of the parameter space that are not well sampled by the training data (but new techniques are emerging)



"TRADITIONAL" ML METHODS

Not Neural Networks (or not deep ones): primary differentiator, based on statistical and mathematical theory

Feature Engineering: rely on pre-extracted features (magnitudes, colors, morphology, etc.)

EXAMPLES OF TRADITIONAL* ML FOR PHOTO-Z

Support Vector Machines (SVMs): For both classification and regression tasks (Heinis+2016)

Instance-Based Models: k-Nearest Neighbors (k-NN) (Beck+2016)

Tree-Based Models: Decision Trees, Random Forests (RF) (Taewan+2025), Gradient Boosting Machines like XGBoost (Li+2022), CatBoost (Li+2024)

*DOESN'T MEAN OUTDATED

(Gradient boosting, Automated Feature Engineering)

MODELING NEURONS

1943: McCulloch (a neuroscientist) & **Pitts** (a logician) : provided the first mathematical neuron model as a computational threshold unit that could perform logical operations. No learning mechanism.

1949: Hebb (a psychologist) : the efficiency of communication between two neurons increases when they are activated simultaneously. This synaptic plasticity is the neurophysiological basis for learning and memory formation: our experiences reshape our brain's structure and function over time.

1958: Rosenblatt (a psychologist) : added the critical component of error correction (target minus prediction) that made his **“perceptron”** the first practical, **learnable neuron model.**

THE PERCEPTRON

1958

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN[1]

F. ROSENBLATT

Cornell Aeronautical Laboratory

If we are eventually to understand the capability of higher organisms for perceptual recognition, generalization, recall, and thinking, we must first have answers to three fundamental questions:

1. How is information about the physical world sensed, or detected, by the biological system?
2. In what form is information stored, or remembered?
3. How does information contained in storage, or in memory, influence recognition and behavior?

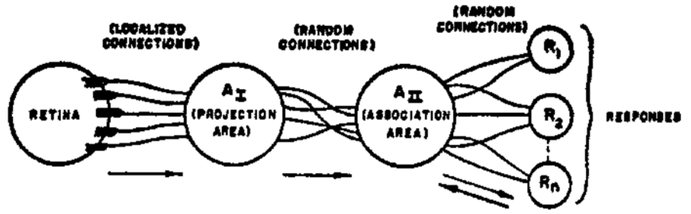
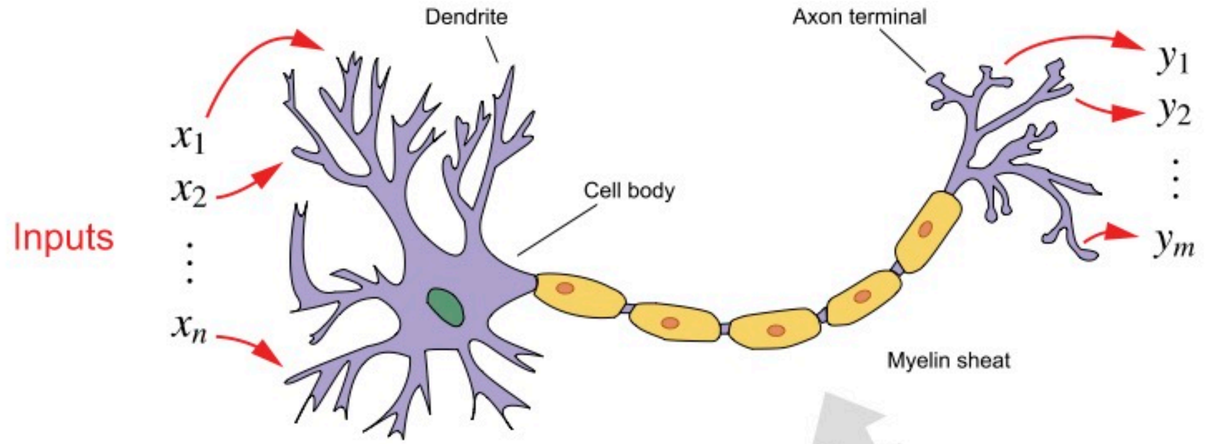
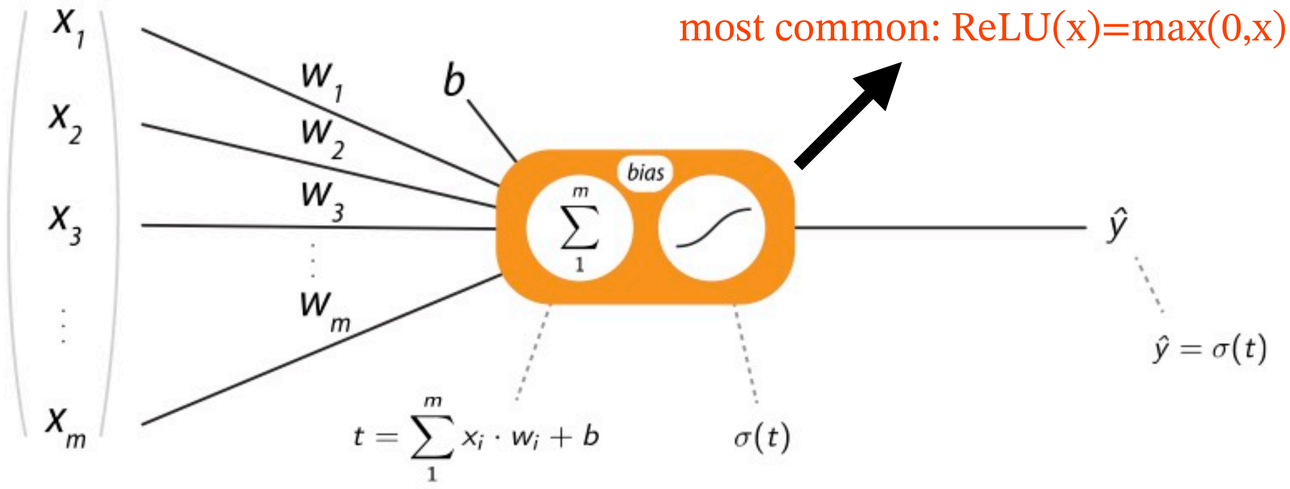


FIG. 1. Organization of a perceptron.

the theory reported here clearly demonstrates the feasibility and fruitfulness of a quantitative statistical approach to the organization of cognitive systems. By the study of systems such as the perceptron, it is hoped that those fundamental laws of organization which are common to all information handling systems, machines and men included, may eventually be understood.

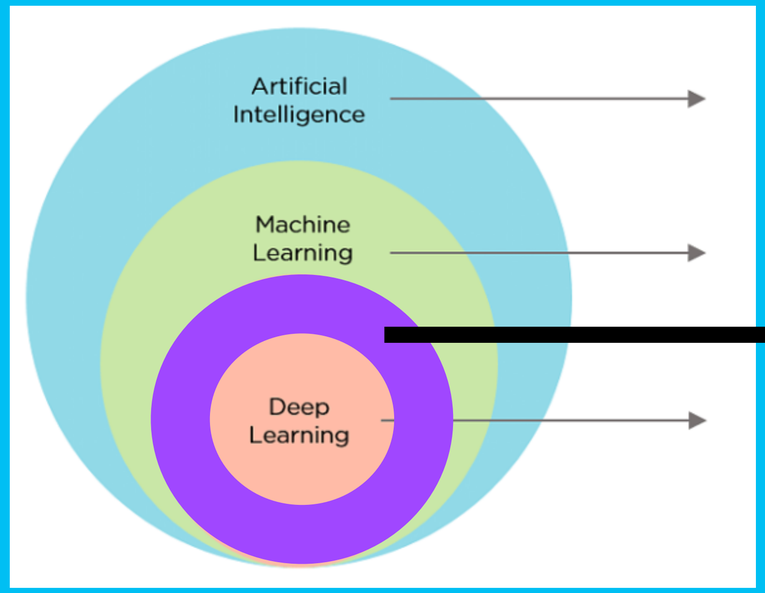
AN ARTIFICIAL NEURON





weights & bias

activation function (non linearity)

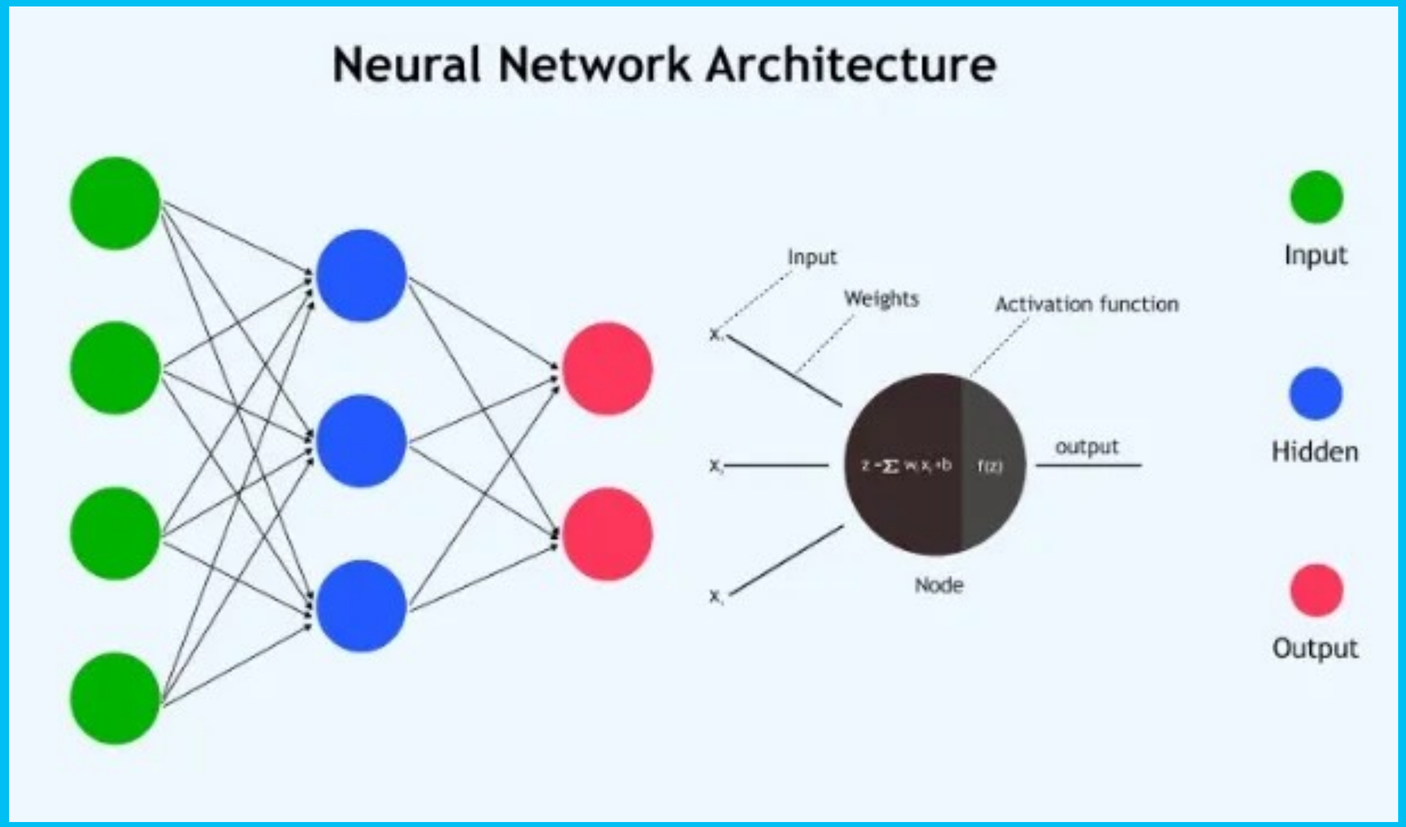


"TRADITIONAL" NEURAL NETWORKS

input pre-extracted features (magnitudes, colors, morphology, etc.)

optimize weights and biases of all the neurons to minimize a loss function

output redshift



FIRST NEURAL NETWORKS FOR PHOTO-Z

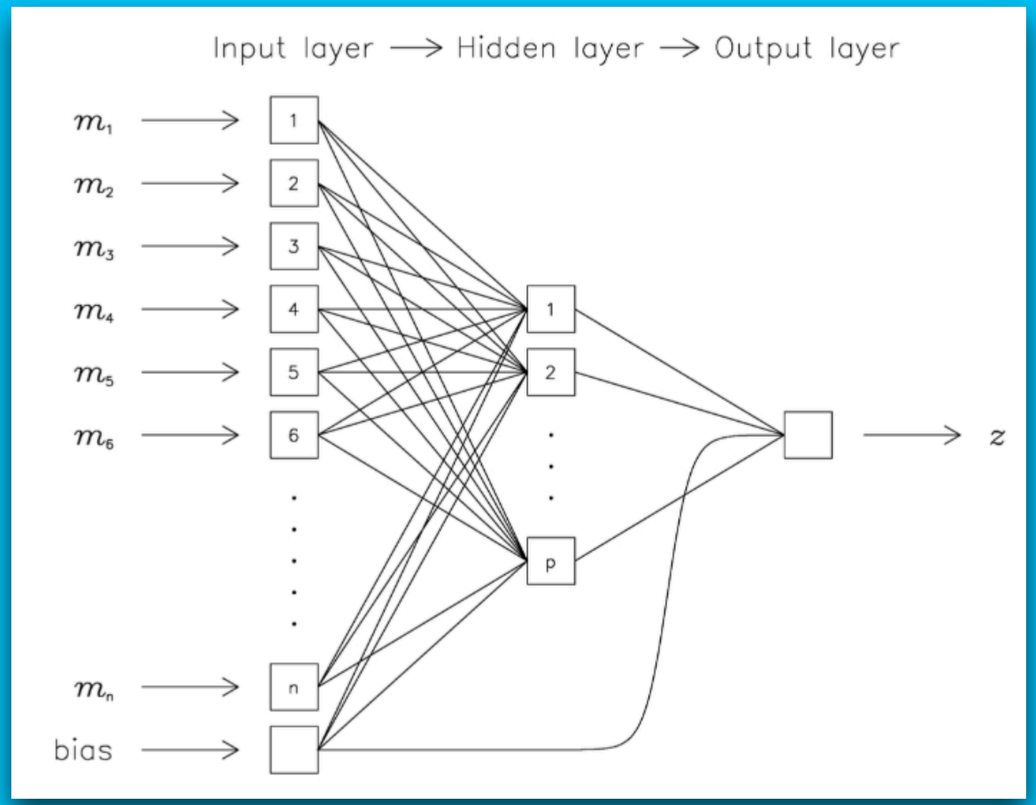
THE MULTI-LAYER PERCEPTRON (MLP)

Neural Networks for Photometric Redshifts Evaluation (Tagliaferri+2003)

Estimating photometric redshifts with artificial neural networks (Firth, Lahav & Somerville 2004)

ANNz: Estimating Photometric Redshifts Using Artificial Neural Networks (Collister & Lahav 2004)

Photometric redshifts with the Multilayer Perceptron Neural Network: Application to the HDF-S and SDSS (Vanzella+2004)



FIRST NEURAL NETWORKS FOR PHOTO-Z

SELF-ORGANIZING MAP (SOM)

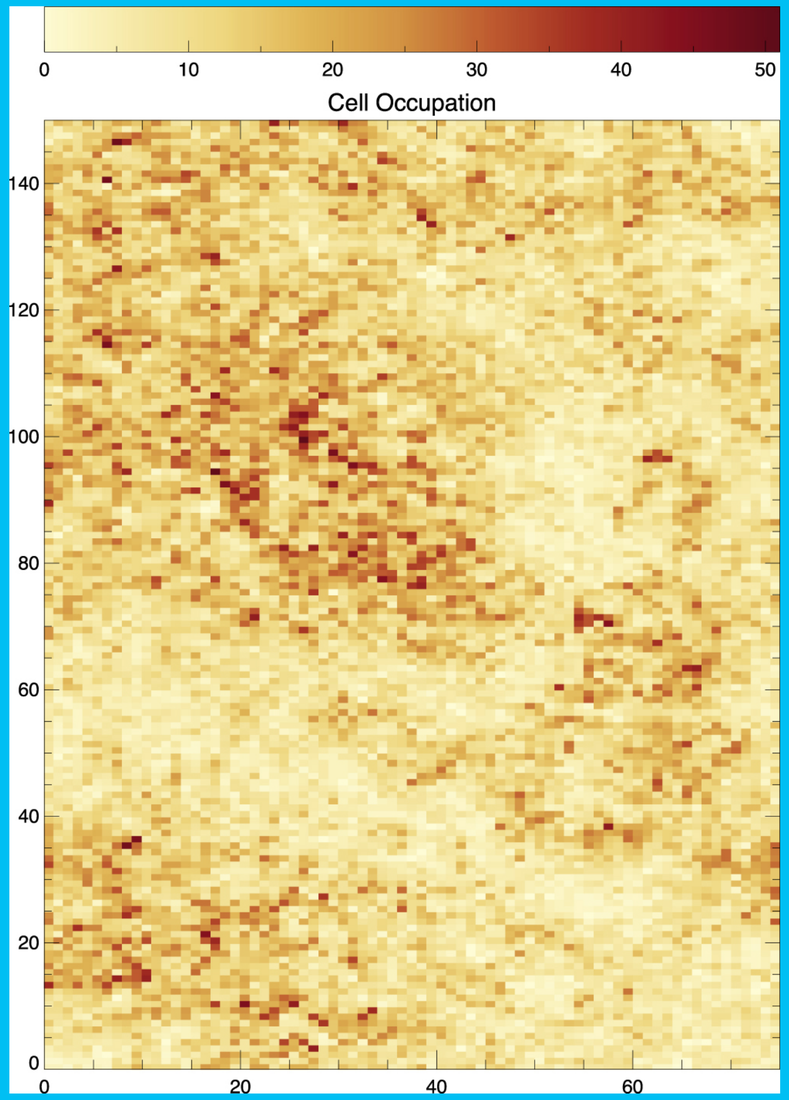
(or Kohonen maps from T. Kohonen in the 80's)

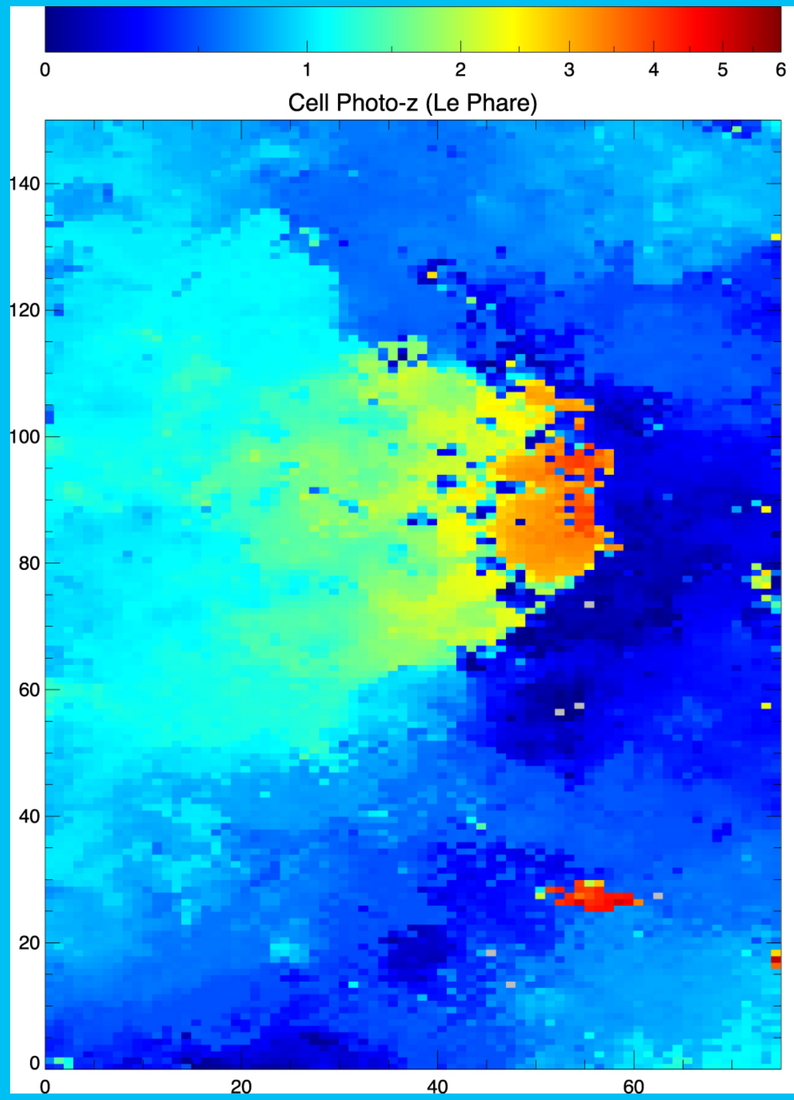
An unsupervised neural network that identifies natural groups within the data without prior knowledge;
Reduces dimensionality: project high-dimensional data onto a 2D map;
Keeps topological relationships: similar data points in the original space remain close to each other on the map.

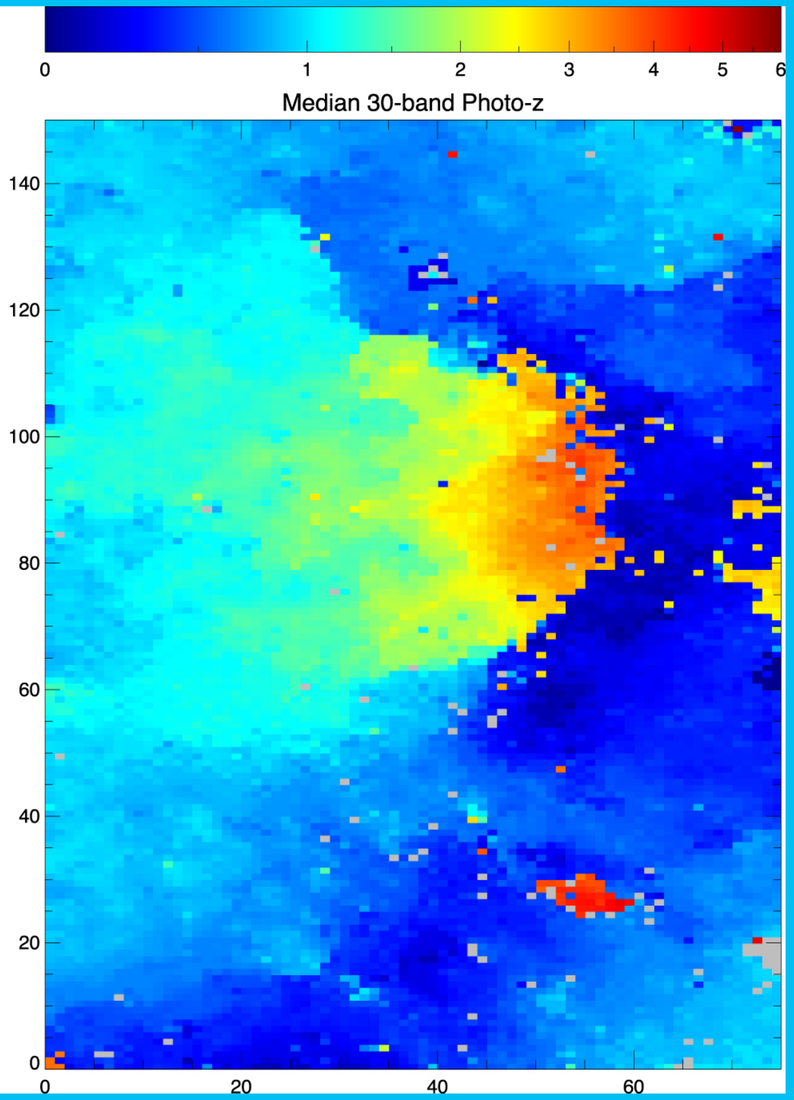
Application: calculate mean redshift within each cell from a training sample or use as data preprocessing for a more complex supervised learning model.

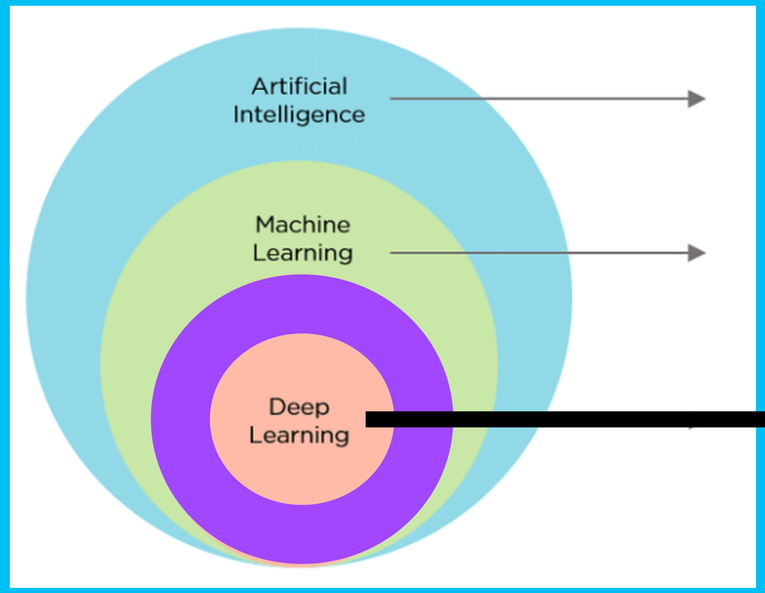
AN EXAMPLE OF SOM FOR PHOTO-Z

INPUT: COSMOS field colors (Masters+2015)









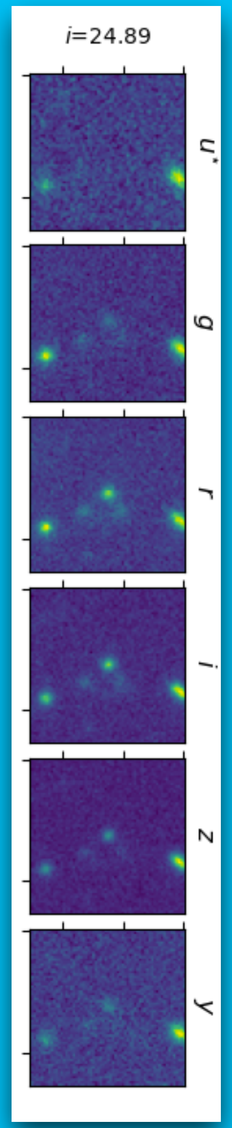
DEEP NEURAL NETWORKS

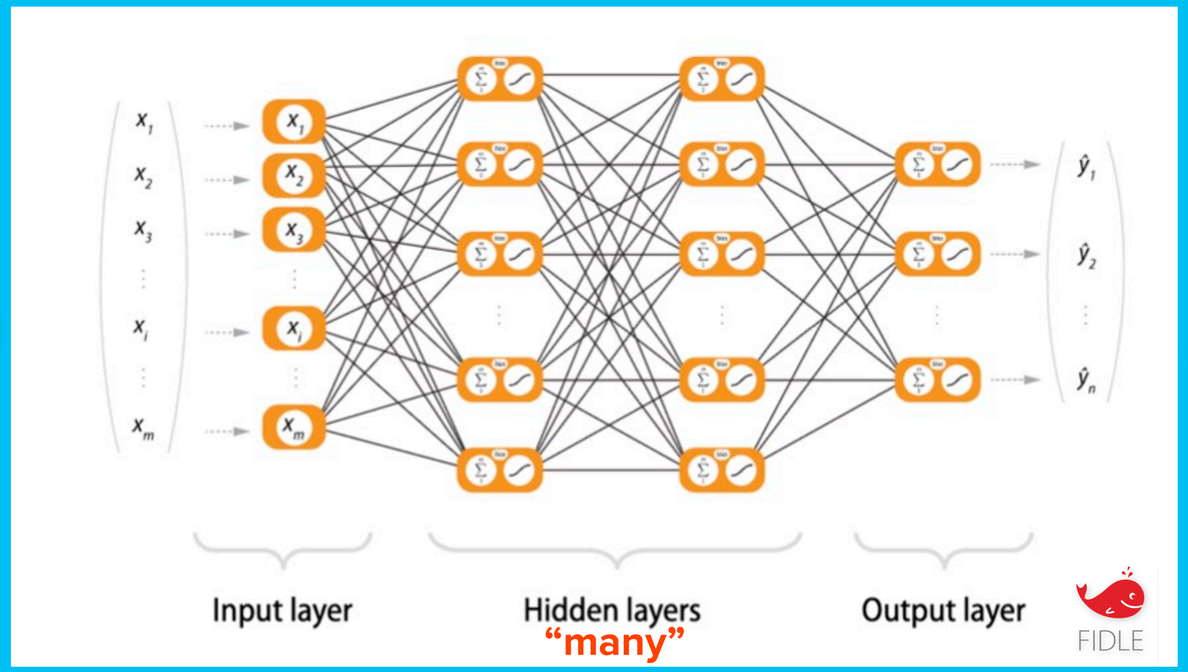
automatically learn hierarchical representations from raw data

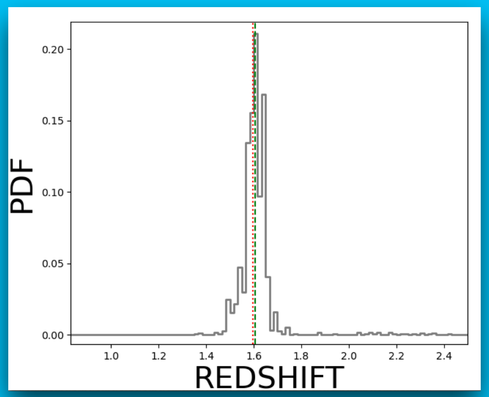
"The curse of dimensionality" (Bellman 1961) made this goal impossible for decades: If you try to learn directly from raw, high-dimensional data (e.g., a 100x100 pixel image lives in a 10,000-dimensional space), traditional ML methods fail because:

- The data are impossibly sparse in that vast space
- The number of parameters needed to model them is astronomical, leading to overfitting
- The computational cost is prohibitive

DEEP NEURAL NETWORKS







DL architectures, particularly CNNs, exploit the structure of data to defeat the curse of dimensionality

CONVOLUTIONAL NEURAL NETWORKS

The full connectivity between nodes in MLP caused the curse of dimensionality and was computationally intractable with higher-resolution images. MLPs ignore the spatial structure of images. They treat all pixels as independent features. **CNNs are variants of MLP designed to emulate the behavior of the visual cortex.** They address the limitations of MLPs by leveraging the spatial locality and hierarchical structure inherent in natural images

DEEP NEURAL NETWORK ECOSYSTEM

A Python centered world

and GPU

python™ NumPy jupyter

« Deep learning for humans »

Widely used in the implementation of practical solutions

Keras

By François Cholet (Google)
Hight level API
Part on TensorFlow since 2017
MIT licence

TensorFlow

Most used DL framework
Supported by Google
Low level API – an hard way
Apache licence

PyTorch Lightning

High-level interface for PyTorch, by William Falcon.
Lightning 2.0 is featuring a clean and stable API!!

PyTorch

From Torch library
Supported by Facebook
BSD licence

Widely used in the field of AI research

FIDLE

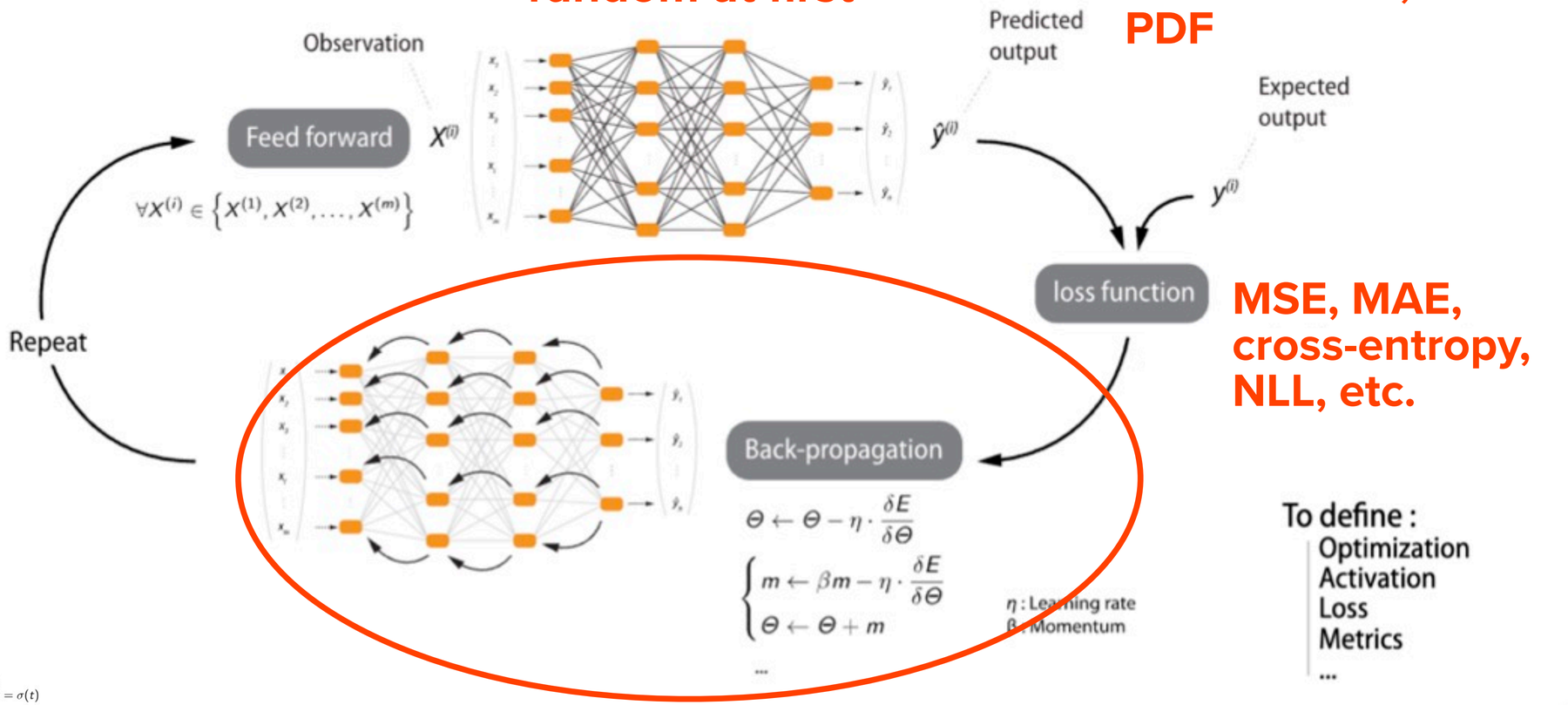
thanks to tech giants building ever-larger LLMs...

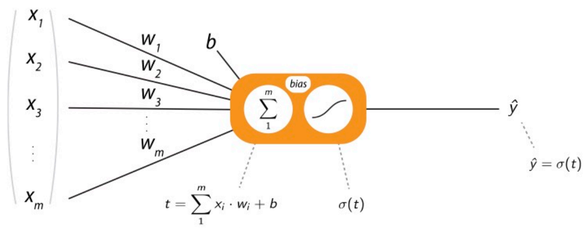
HOW DOES IT WORK?

Training process - general

lots of w and b, random at first

regression, classification, PDF



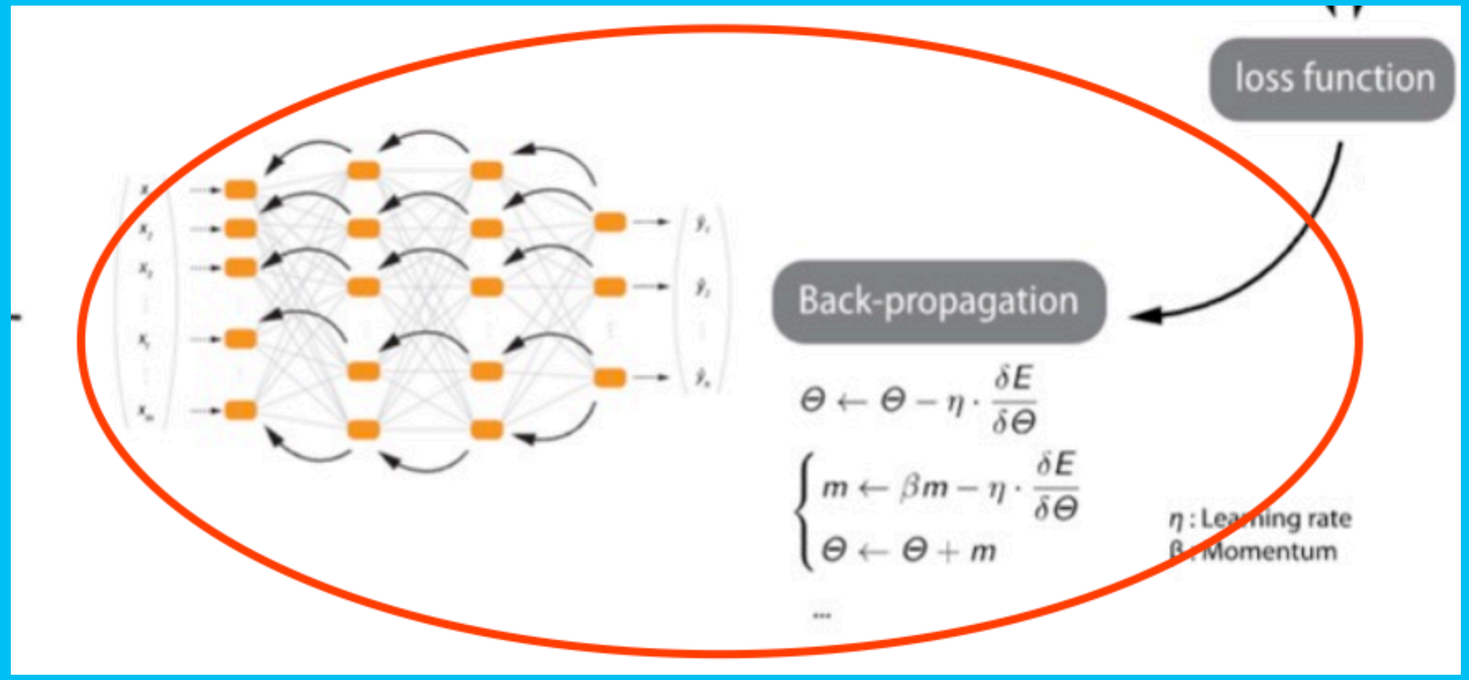


one neuron

Gradients for all w and b ← (Chain Rule) ← Loss

FIDLE

BACK-PROPAGATION



Gradients for all w and b ⟵ (Chain Rule) ⟵ Loss

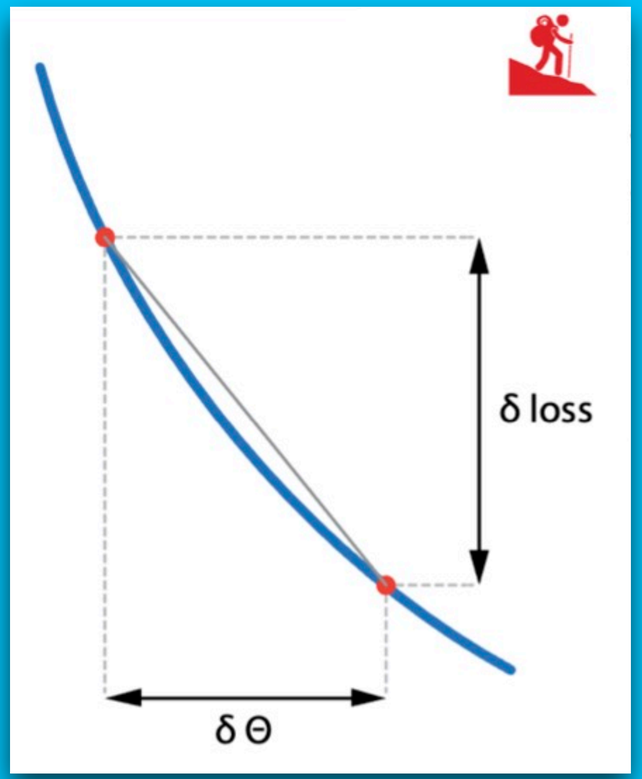
The network is just a massive composition of functions :

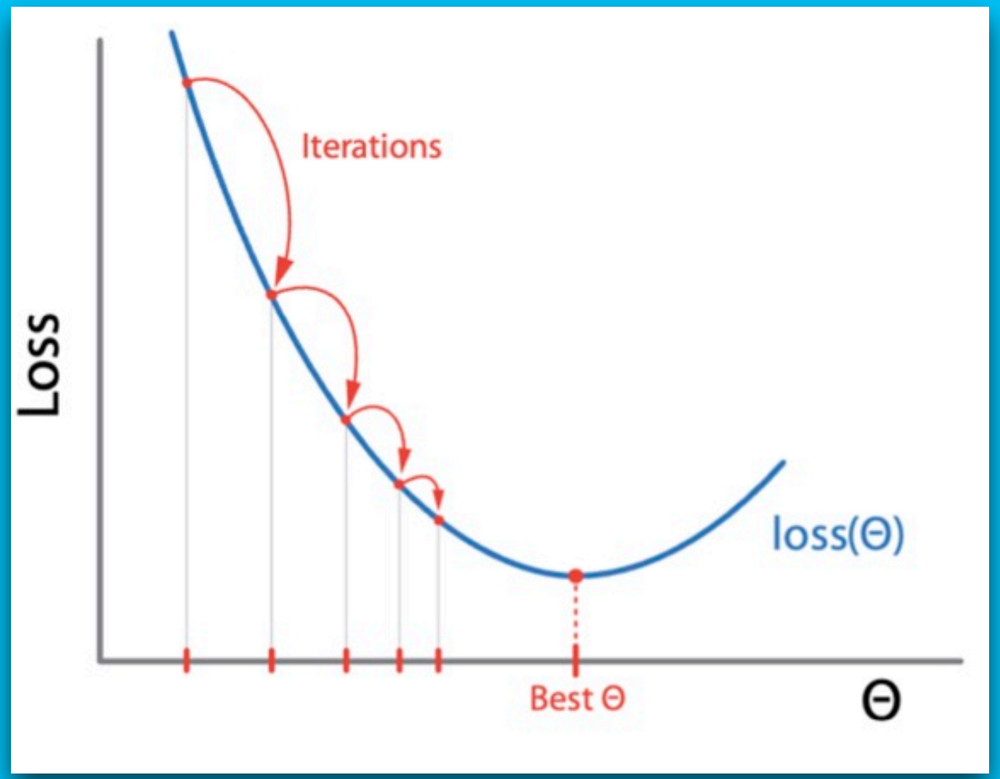
output = f(g(h(...(input)...)))

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$$

The gradient for one layer can be computed using the gradient from the layer ahead of it. The chain rule from calculus allows you to calculate the derivative of the final loss with respect to any weight by multiplying the derivatives along the path that connects them. Extremely efficient!

GRADIENT DESCENT

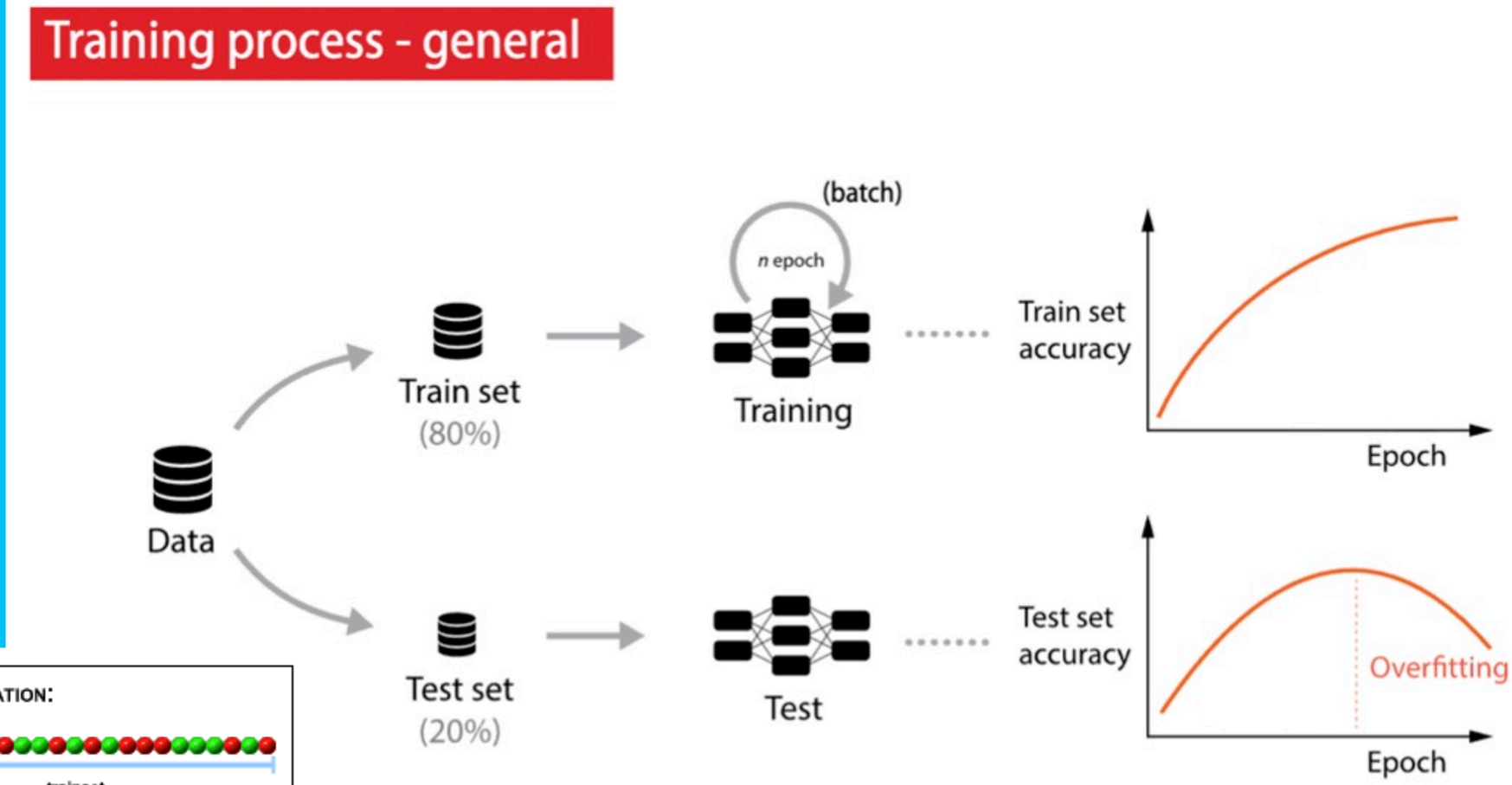


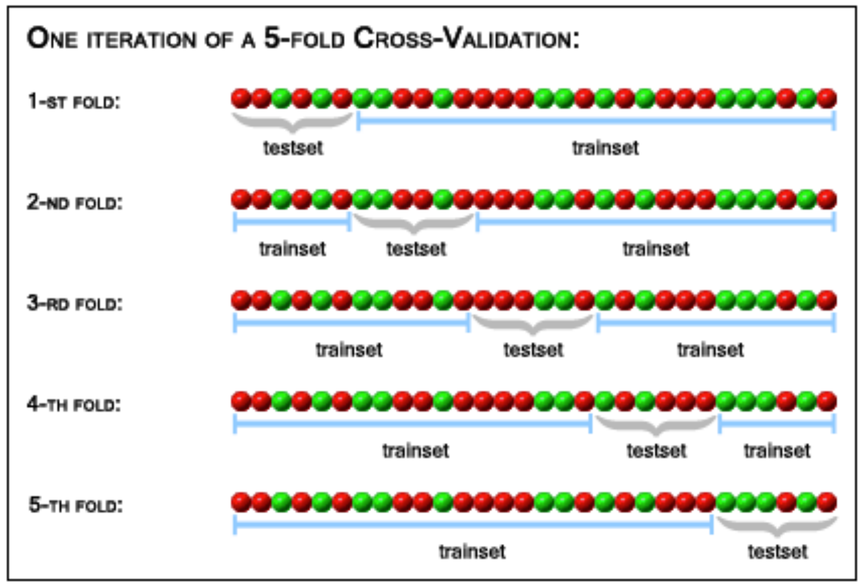


Gradients + Learning Rate → updated w and b

New w = Old w - (Learning Rate * Gradient of w)

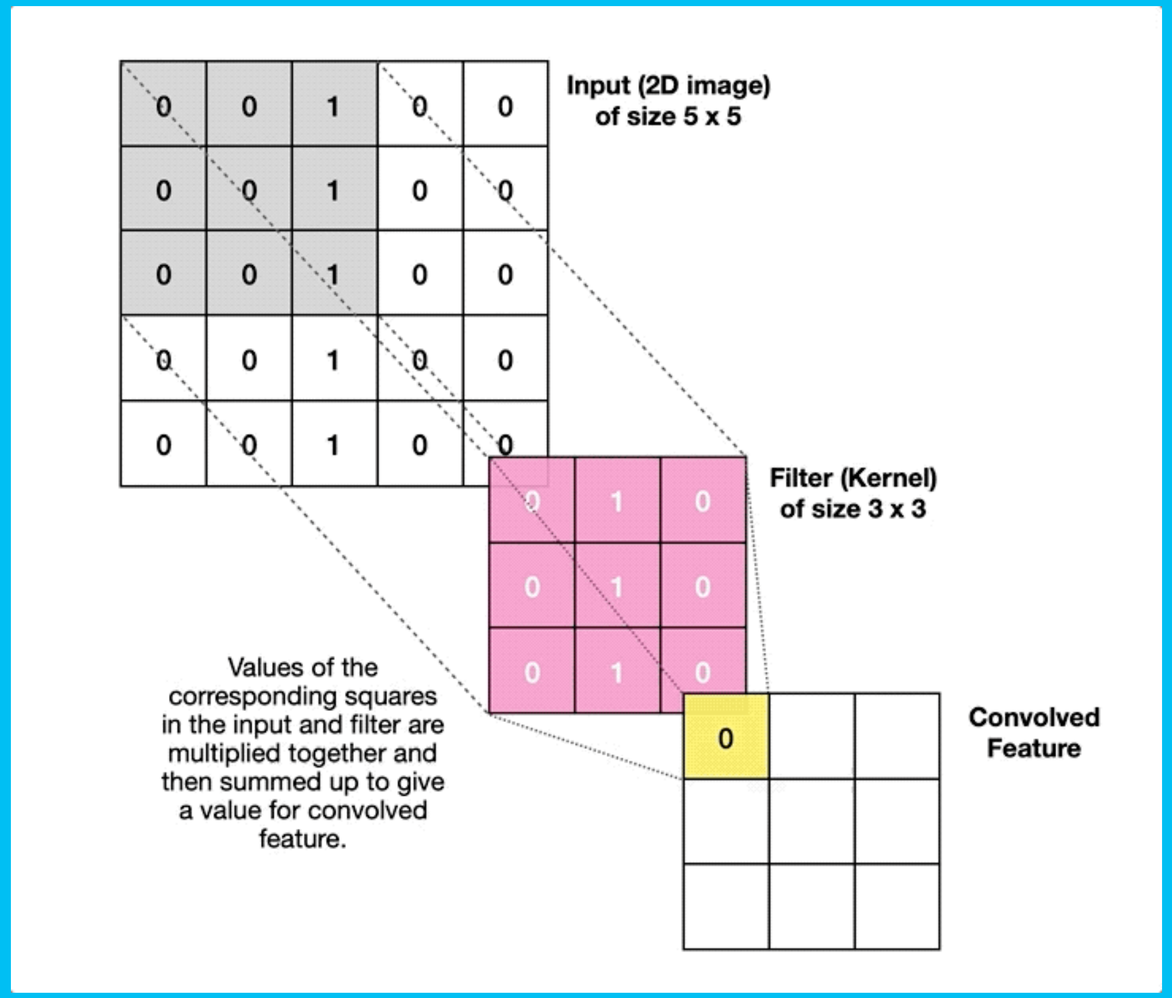
CROSS-VALIDATION PROTOCOL



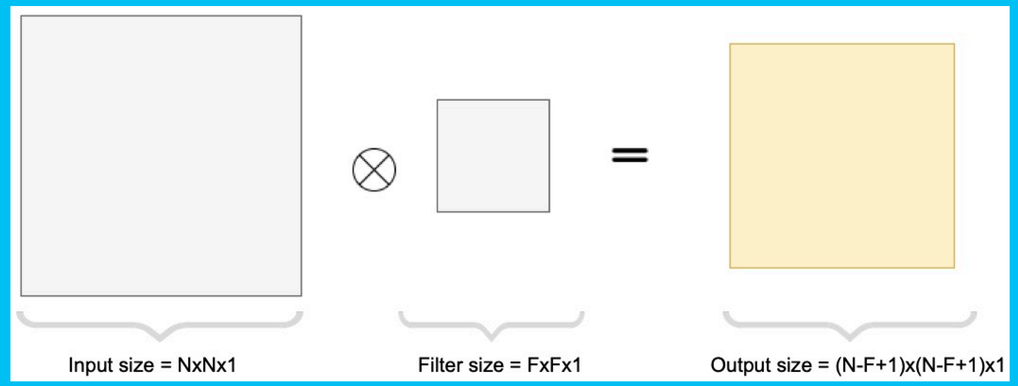


FIDLE

BACK TO CNN



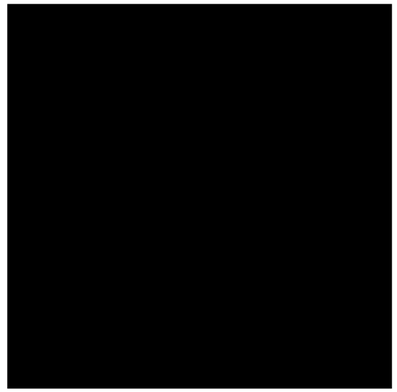
A CONVOLUTION



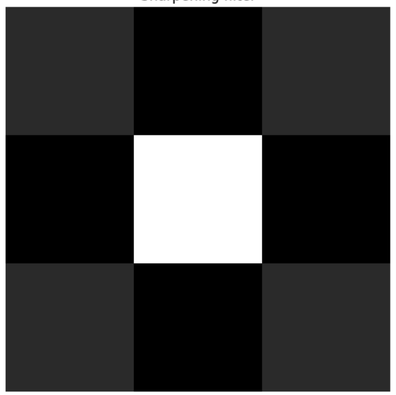
KERNELS ARE DETECTORS

Original Image

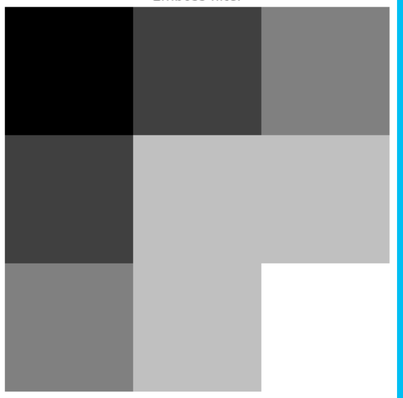
Blur filter



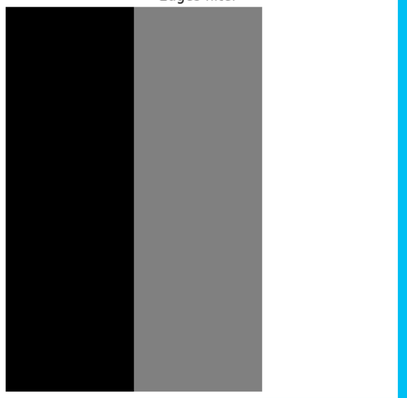
Sharpening filter



Emboss filter



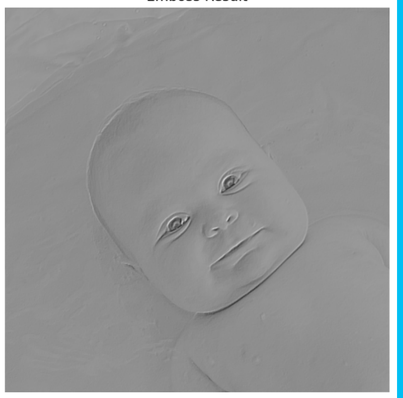
Edges filter



Blur Result

Sharpening Result

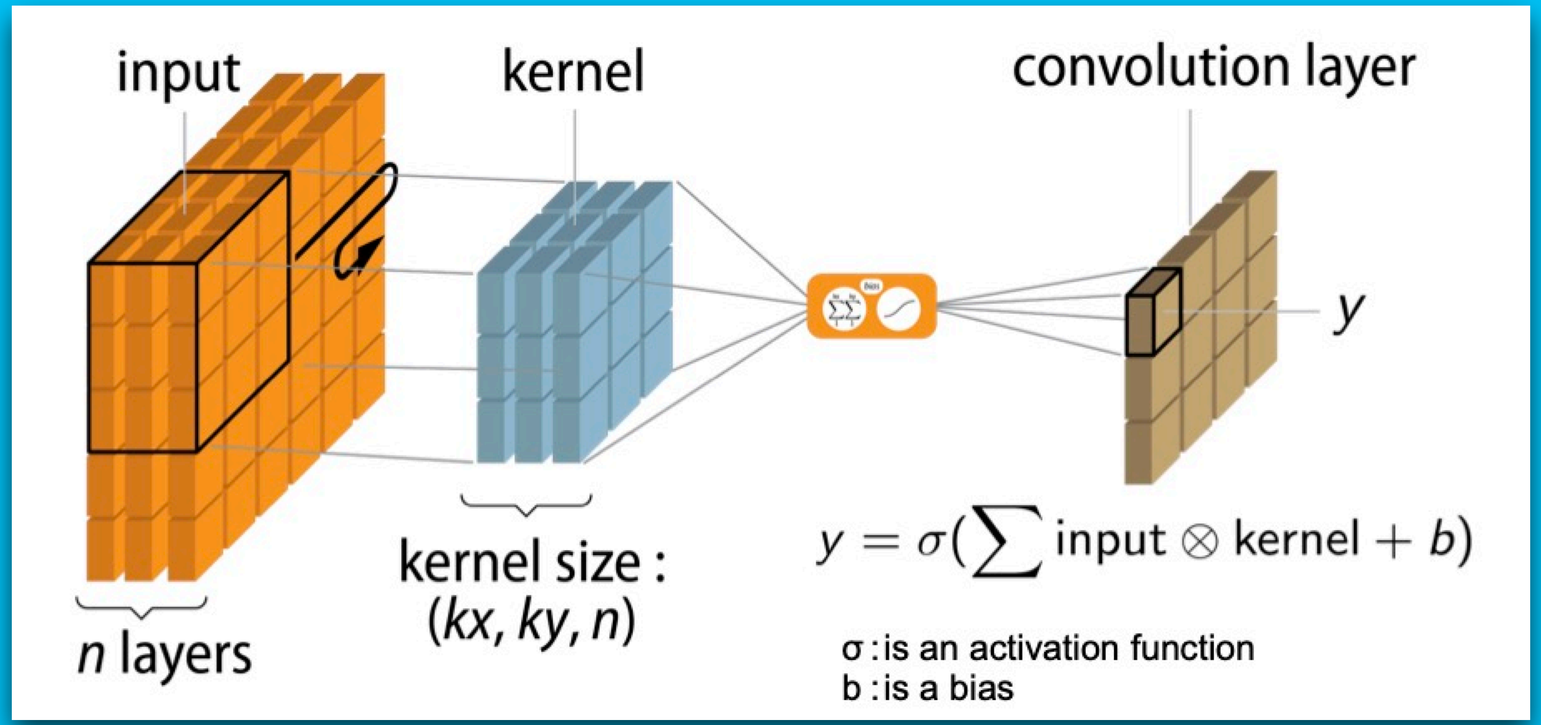
Emboss Result

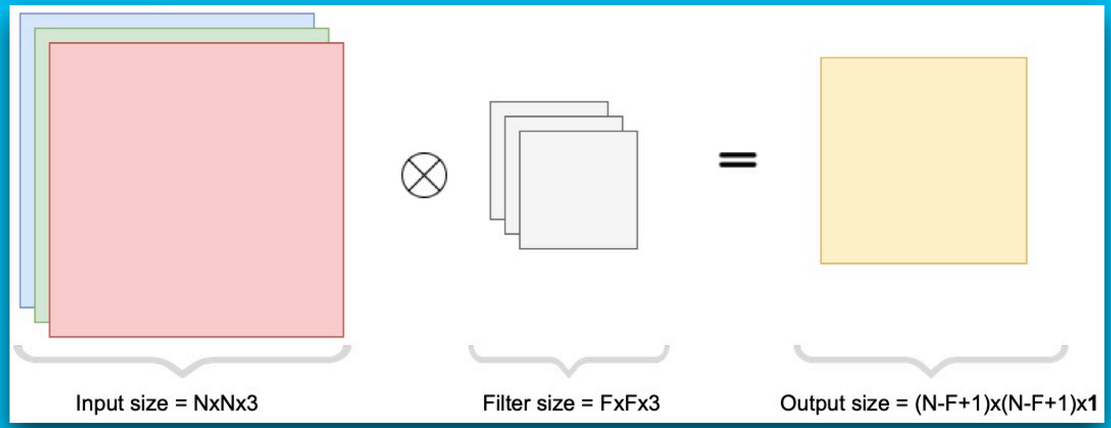


Edges Result

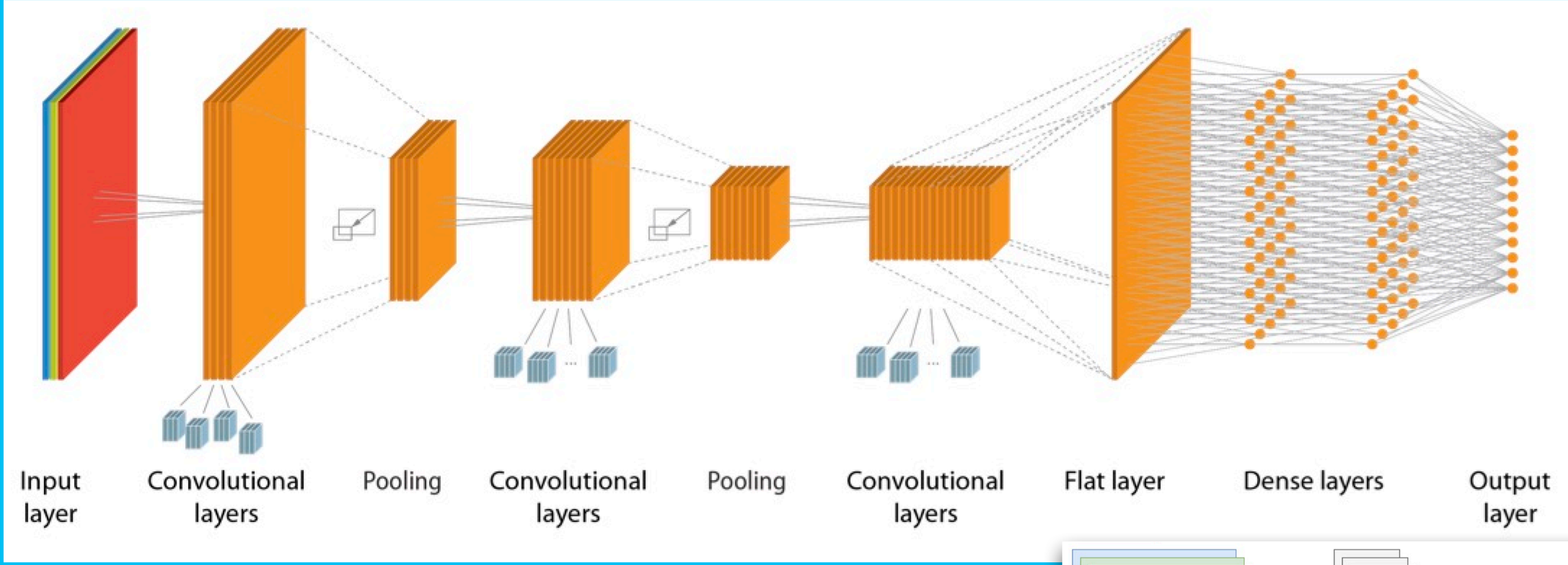
CONVOLUTIONAL LAYER

The n input layers are the galaxy images in n different filters

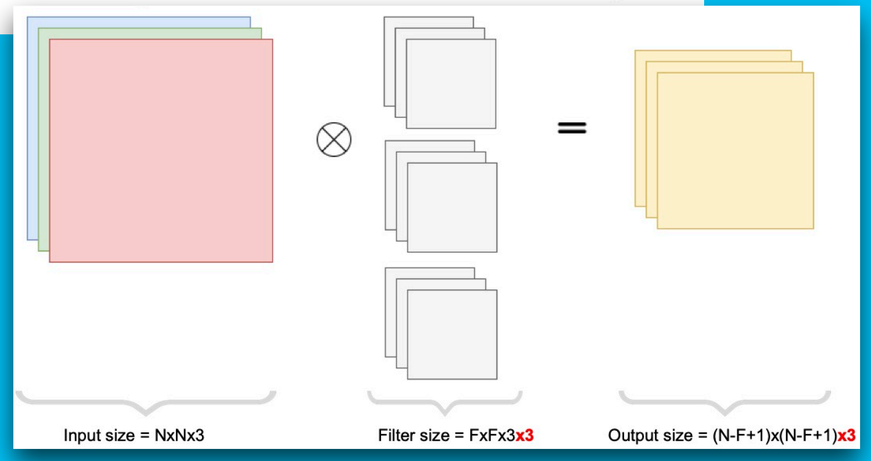




CONVOLUTIONAL NEURAL NETWORK



the kernels are the weights to be adjusted (millions)



CONVOLUTIONAL NEURAL NETWORKS

- Local connectivity: kernels focus on small regions, capturing spatial hierarchies
- Parameter sharing: the same kernel is applied across the entire image, drastically reducing the number of parameters compared to fully connected networks like MLP
- Translation invariance: pooling makes the network robust to small shifts in object position
- Hierarchical feature learning: automatically learns features at multiple scales (edges → textures → objects)

CNN FOR PHOTO-Z

First demo with images + colors: Hoyle 2016, D'Isanto & Polsterer 2018

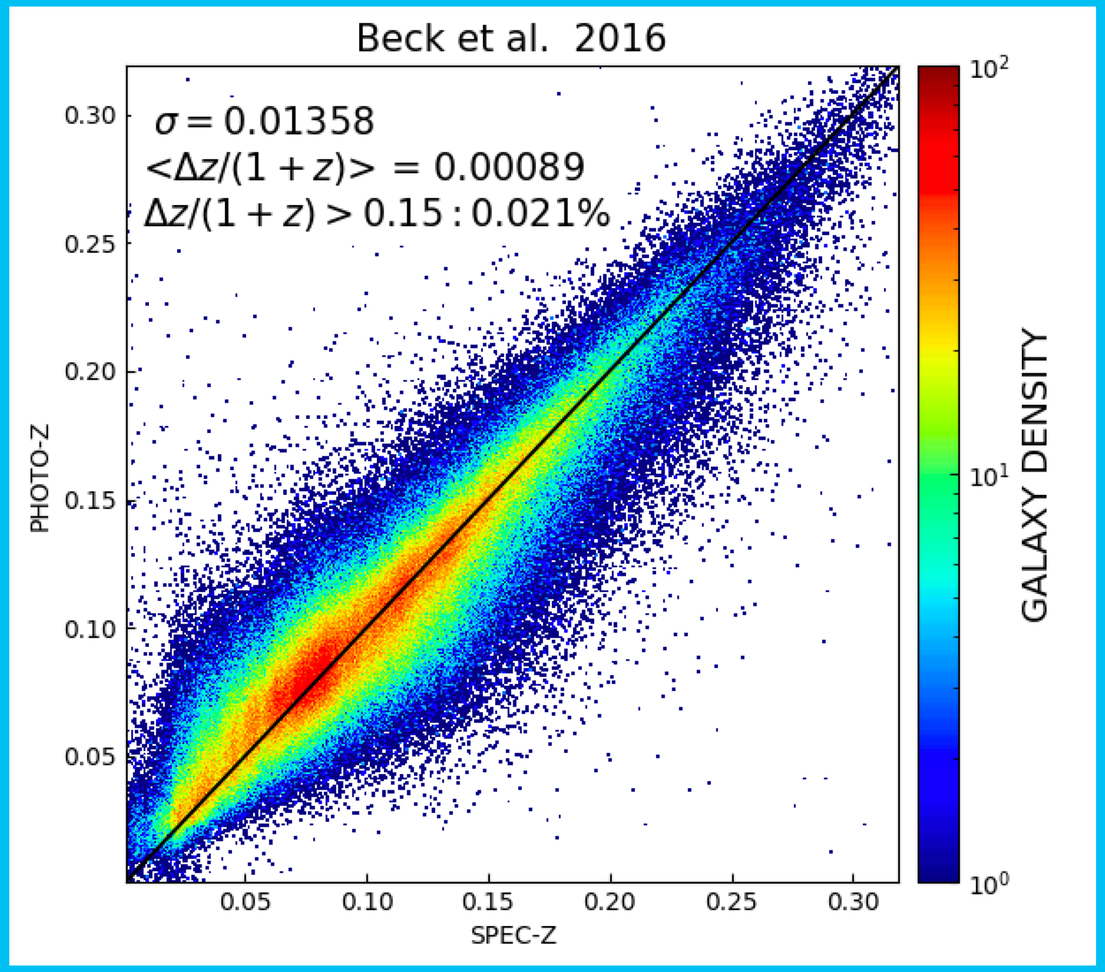
Proof of concept with images only: Pasquet+2019

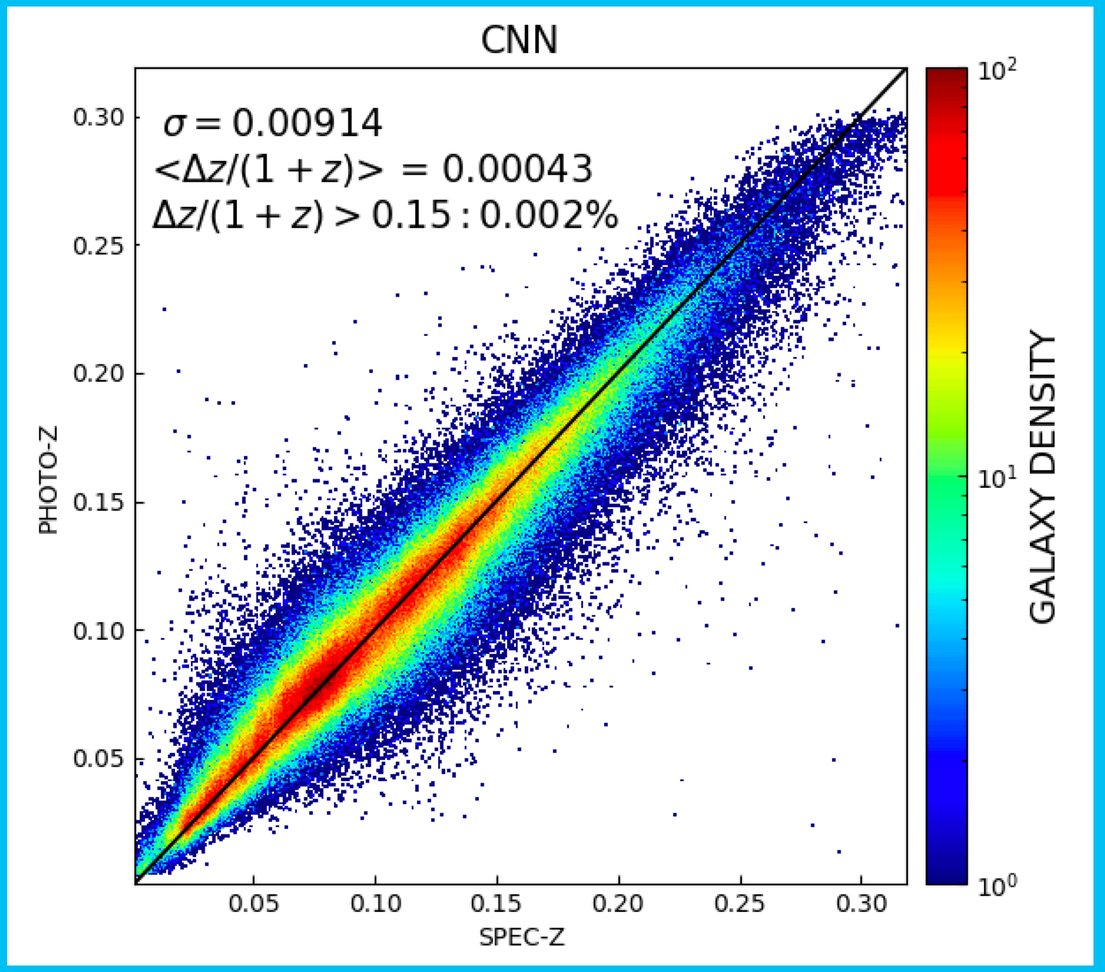
Many attempts at improvements: Dey+2021, Hayat+2021, Ait-Ouahmed+2024, et al.

CNN+MLP: Menou+2019, Henghes+2021, Henghes+2022, Yao+2023, Zhang+2024, Roster+2024, Wei+2025, et al.

Multi-modal* CNN "Mantis Shrimp": Engel+2025, fusing GALEX (UV), PanSTARRS (optical), and WISE (infrared).

*multi-modal: use data of different types (e.g. images and spectra)



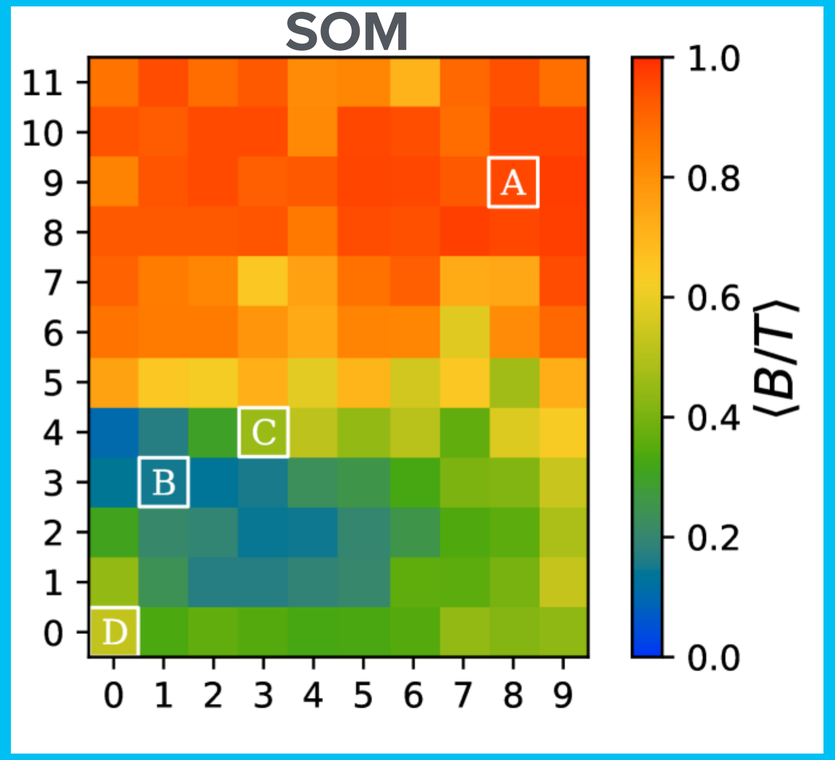


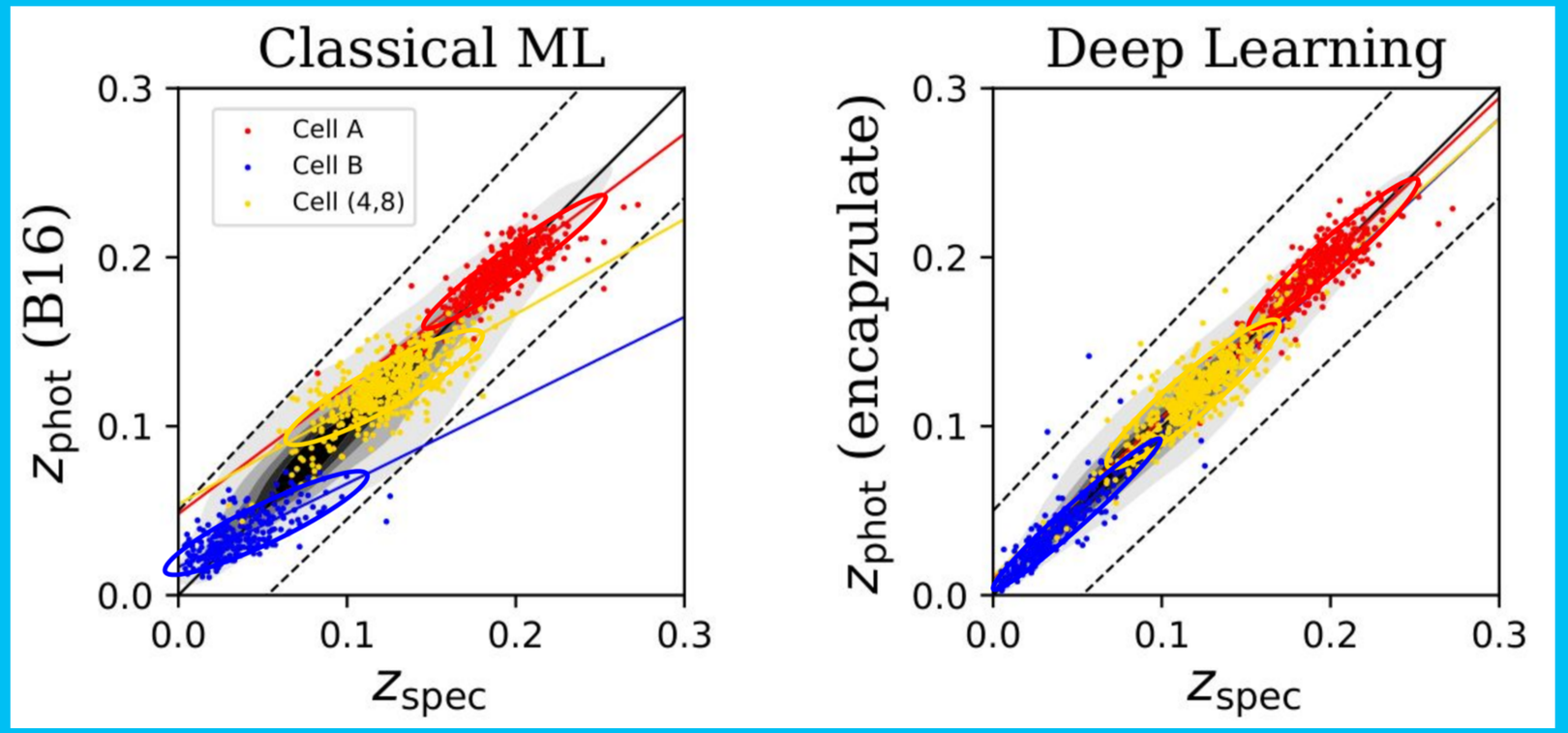
kNN, Beck+2016

CNN, Pasquet+2019

WHY DL IS BETTER THAN ML

Moran+2025





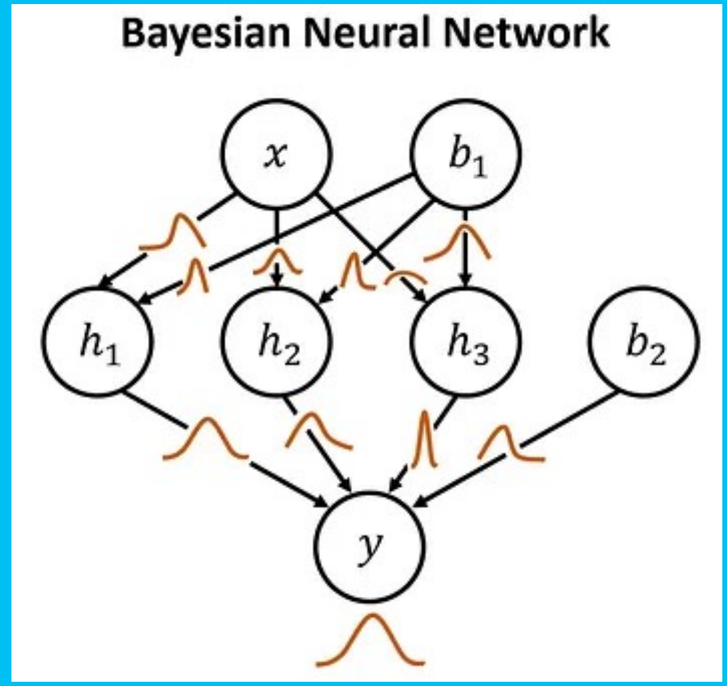
Classical ML shows significant **color-dependent attenuation bias** (photo-z systematically biased towards the cell's mean spec-z), particularly for bluer or less bulge-dominated galaxies. This suggests that the photometry models are less accurate for these and/or that some pixels are more informative about redshift than their relative flux implies. **DL weighs redshift information from individual pixels more optimally than integrated photometry.**

PDF OUTPUT

Uncertainty quantification is critical for downstream applications
(e.g., weak lensing, galaxy evolution studies)

BAYESIAN CNNS ➡ **Treat the weights as random variables**

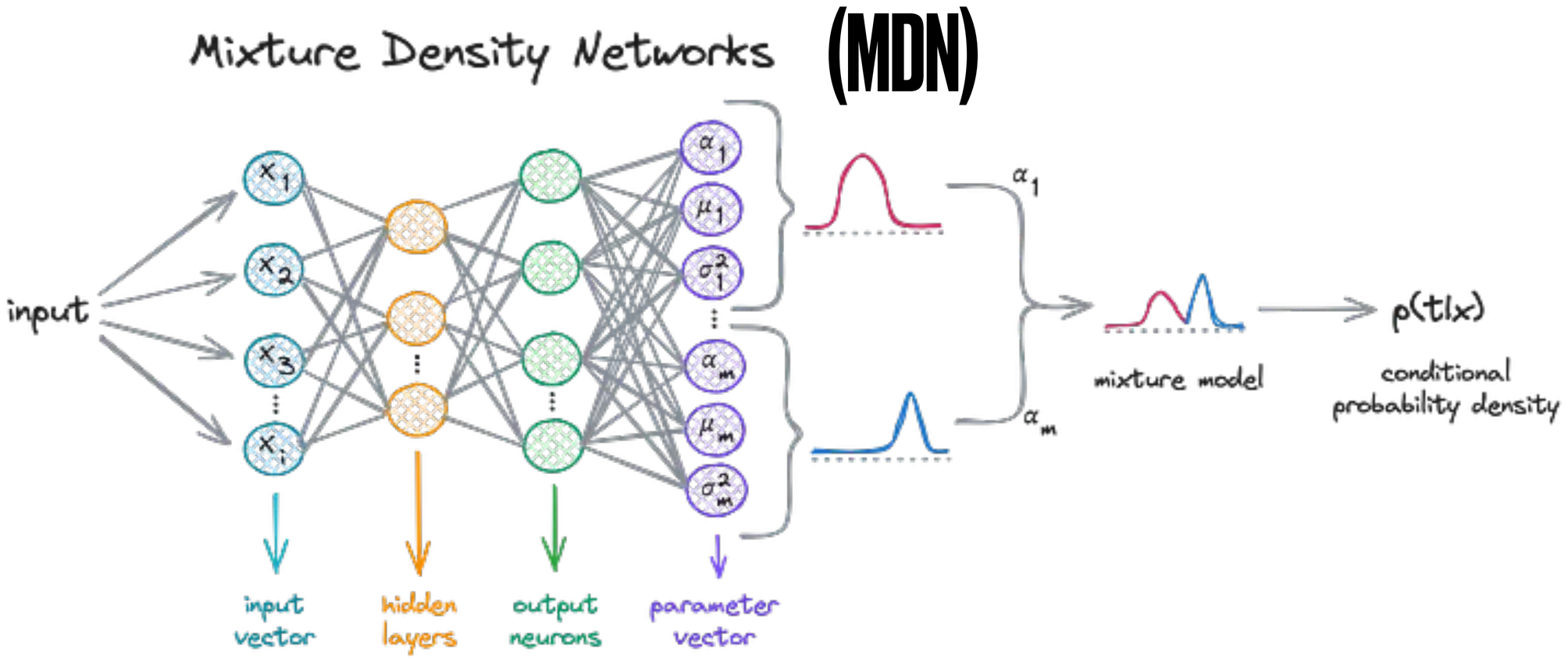
Training: learn the posterior distribution of the weights using **variational inference** (approximates the posterior with a variational distribution, typically a Gaussian) or MC Dropout (applied during both training and inference)
Inference: for a given galaxy, perform multiple forward passes by sampling weights from their posterior distributions. The PDF is the resulting distribution of predictions.



Limitations

- Computational cost
- May not scale well to very large datasets or high-dimensional inputs (e.g. full images)
- Requires careful tuning of the prior and variational distributions
- VI and MC Dropout are approximations: may not capture the true posterior perfectly

ALTERNATIVELY attach a MIXTURE DENSITY NETWORK at the end of the network



Training is done by maximizing the likelihood or minimizing the negative log-likelihood

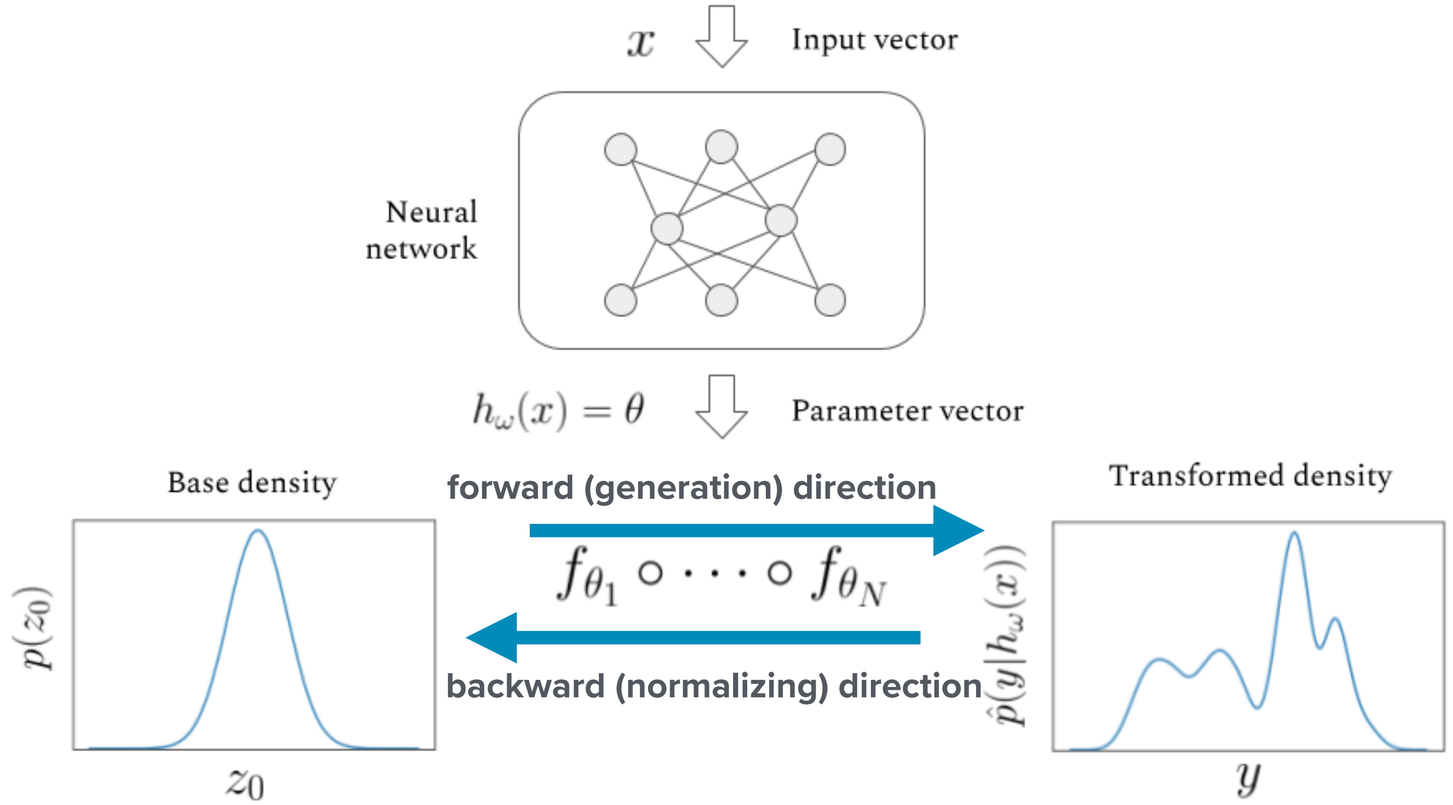
The network generates the weights, means, and variances for each component of the mixture

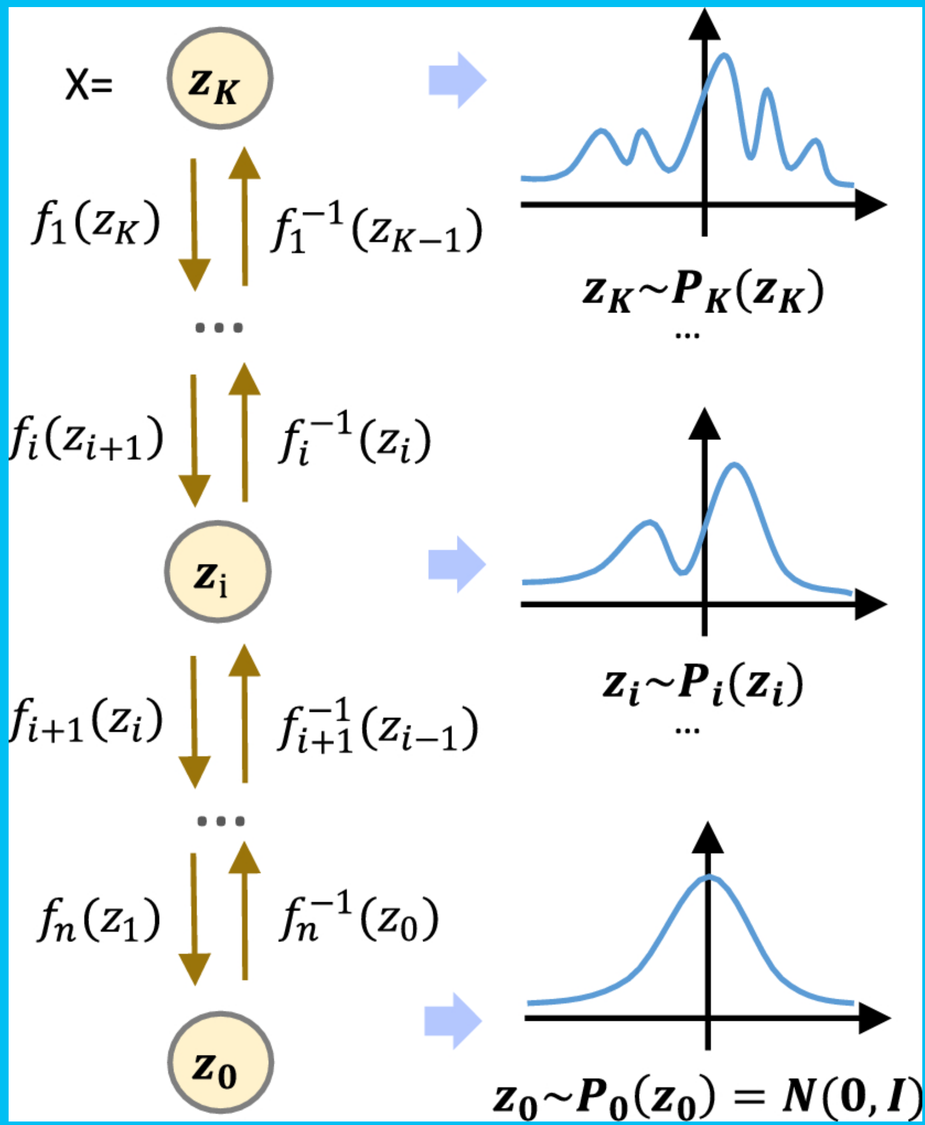
(e.g. Ansari+2021)

ALTERNATIVELY

attach a **NORMALIZING FLOW** at the end of the network

Ren+2025 (ML)





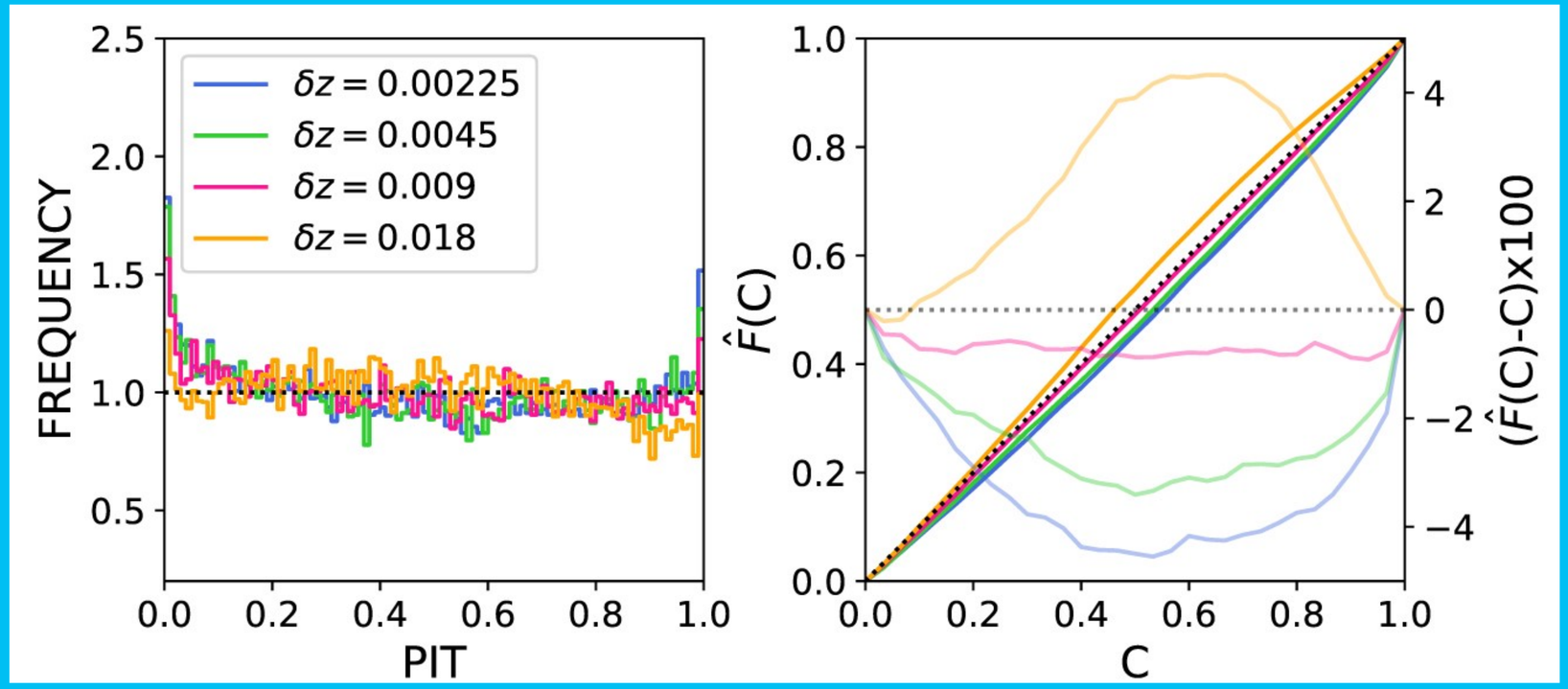
each transformation function should be easily invertible and its Jacobian determinant must be easy to compute.

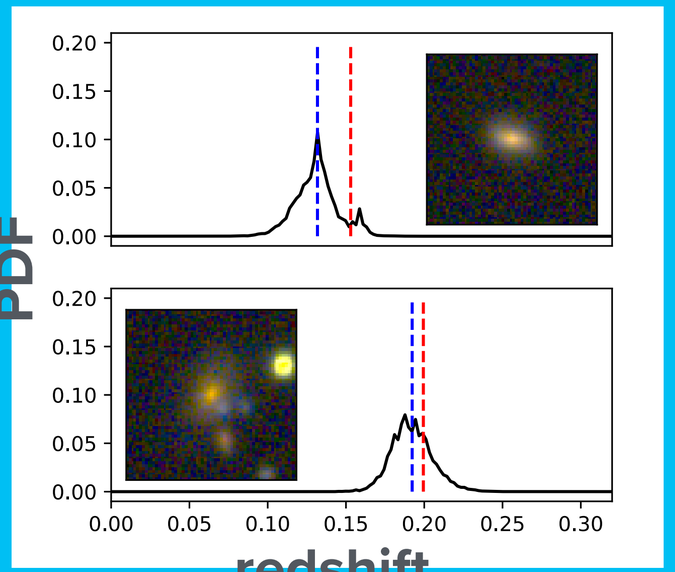
ALTERNATIVELY

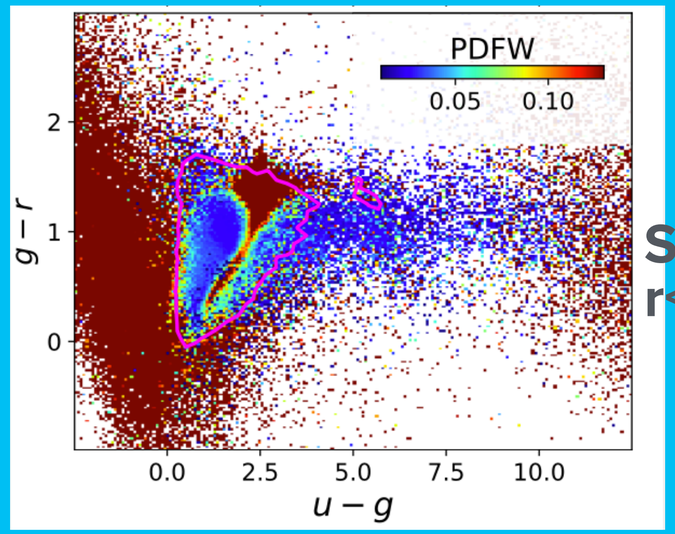
OUTPUT A CLASSIFICATION INSTEAD OF A REGRESSION

(Pasquet+2019, Treyer+2024, and more)

PDF(z_i) = i δz, where $\delta z = z_{max}/N_{bin}$







SDSS
r<17.8

pass PDF tests, can be multi-modal, detect poor predictions, e.g. predictions outside the parameter space of the training sample (stars, galaxies with dubious colors, etc.)

Re predictions outside the parameter space of the training sample:

it’s not just about the physical parameter space of the galaxies in training sample

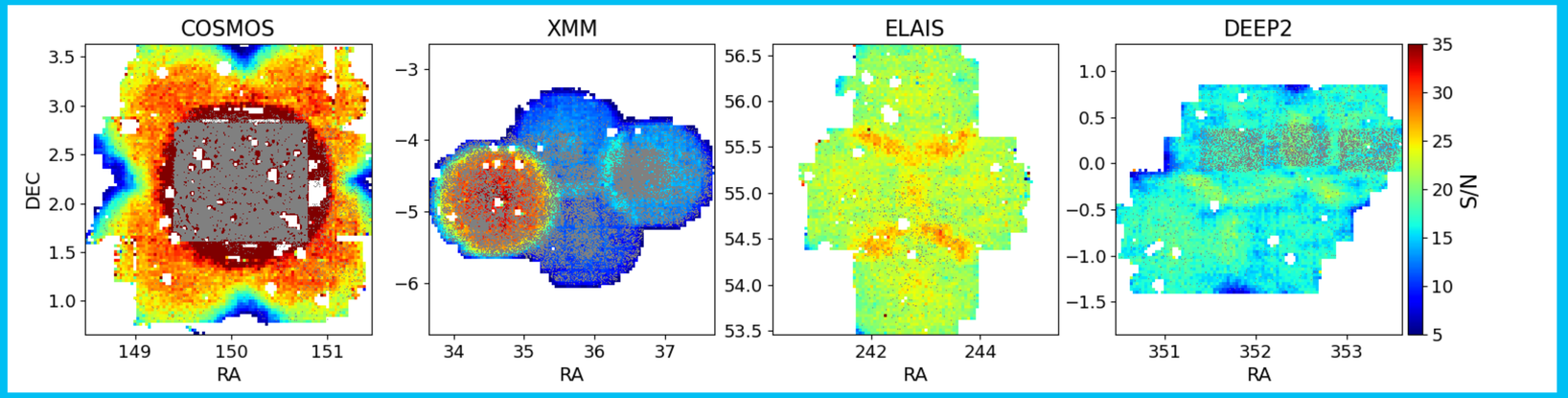
THE DOMAIN ADAPTATION PROBLEM

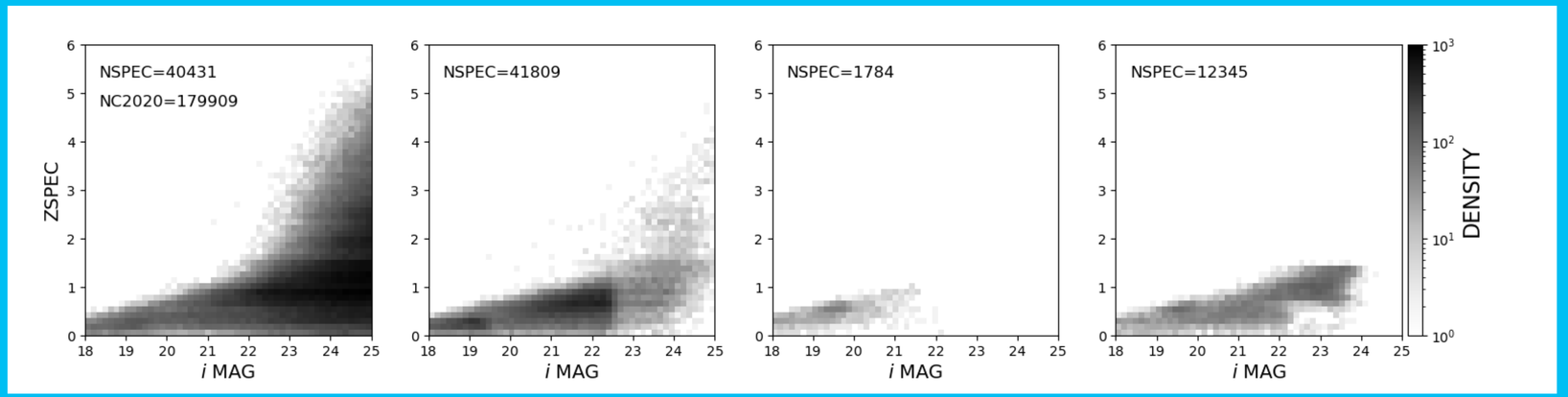
A model trained on one survey (or on one survey field) performs poorly on another due to different observing conditions

AN EXAMPLE OF DOMAIN ADAPTATION PROBLEM

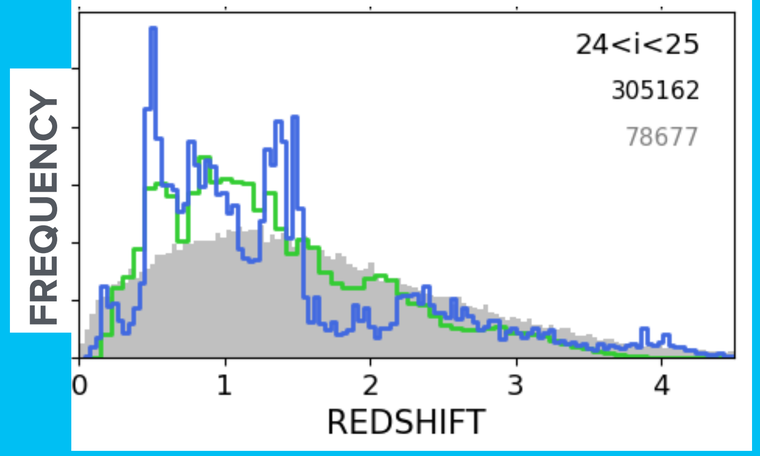
(Treyer+2025)

HSC-CLAUDS SURVEY

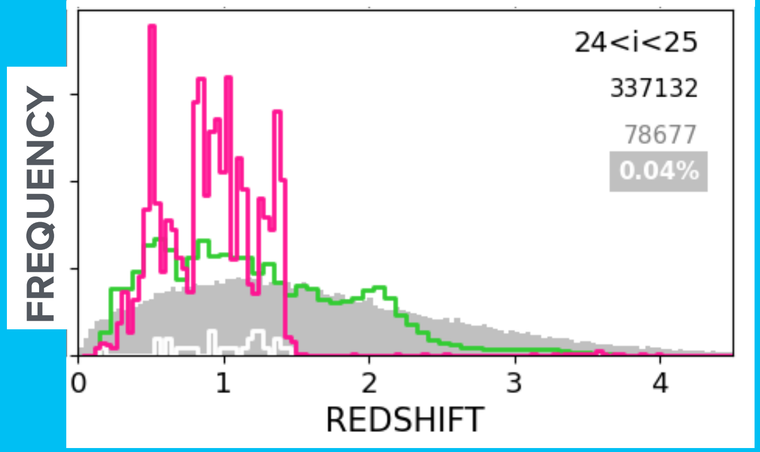




DEEP2 trained on COSMO UDF



DEEP2 trained on COSMO UDF + DEEP2



ztrain
zphot SED fitting
zcnn

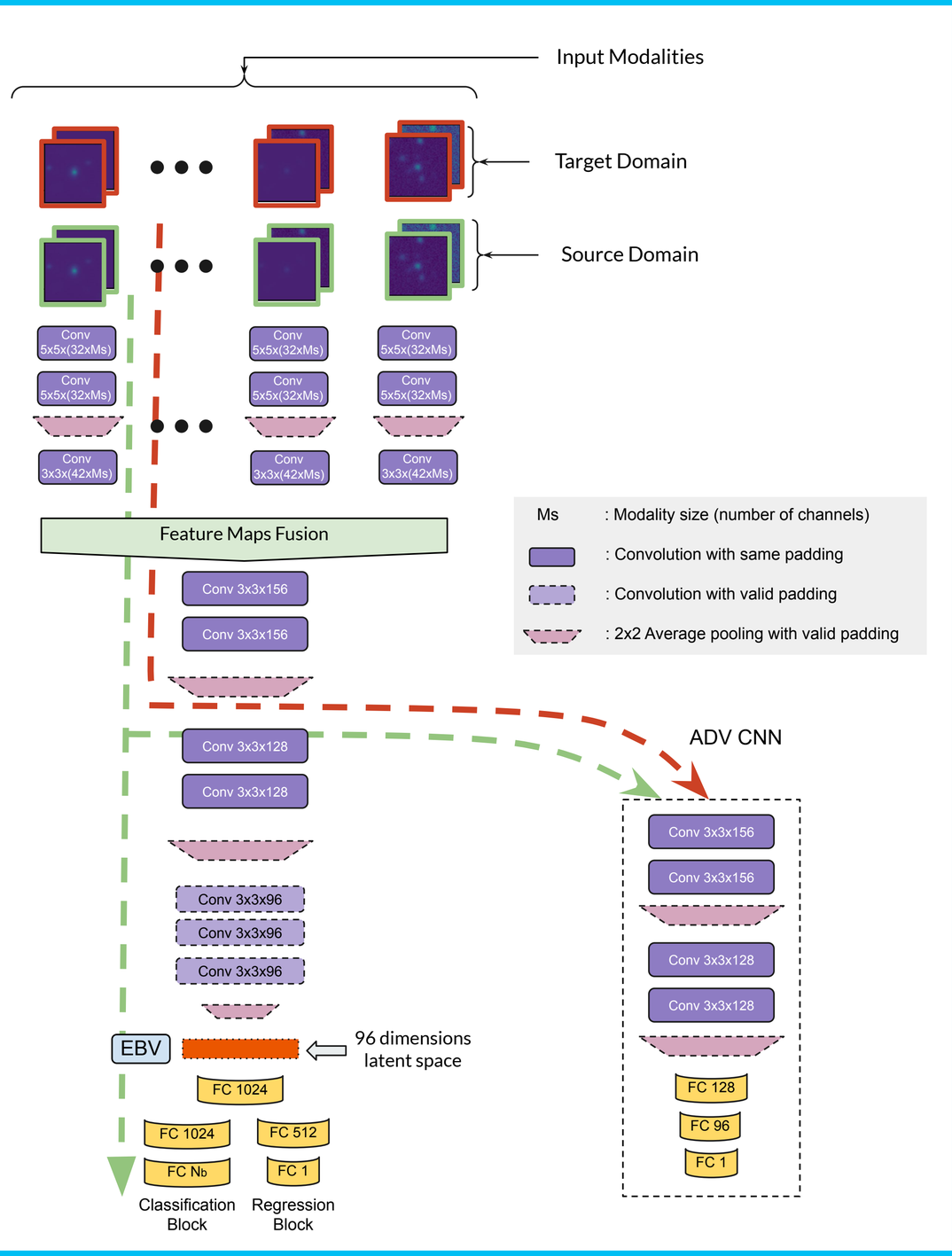
CNN using *ugrizy* images

AN EXAMPLE OF SOLUTION TO THE DOMAIN ADAPTATION PROBLEM

An adversarial module (ADV) receives feature maps from a layer of the main network. Its objective is to identify their field of origin - source or target. The main network is trained, besides estimating redshift, to produce feature maps that will fool the ADV.

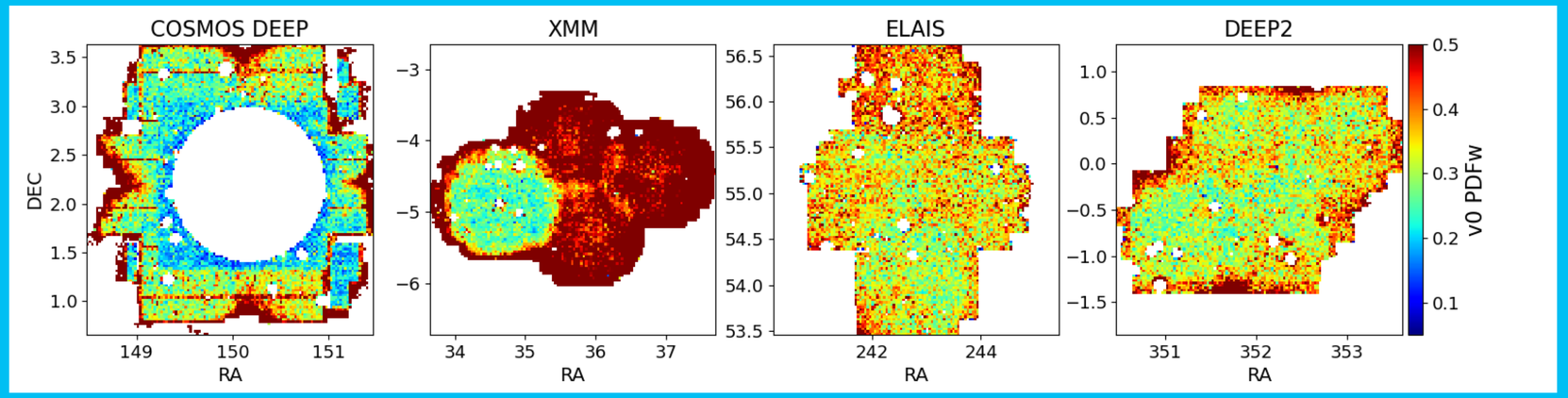
A minimax game: the layers of the network that supply the feature map input to the ADV are trained with the inverse of the ADV's loss (the confusion loss) → they aim to generate feature maps that maximize the ADV's error while enabling accurate redshift predictions.

The ADV and redshift estimation objectives are optimized simultaneously at each iteration, using the same learning rate and the same cross-entropy loss function.

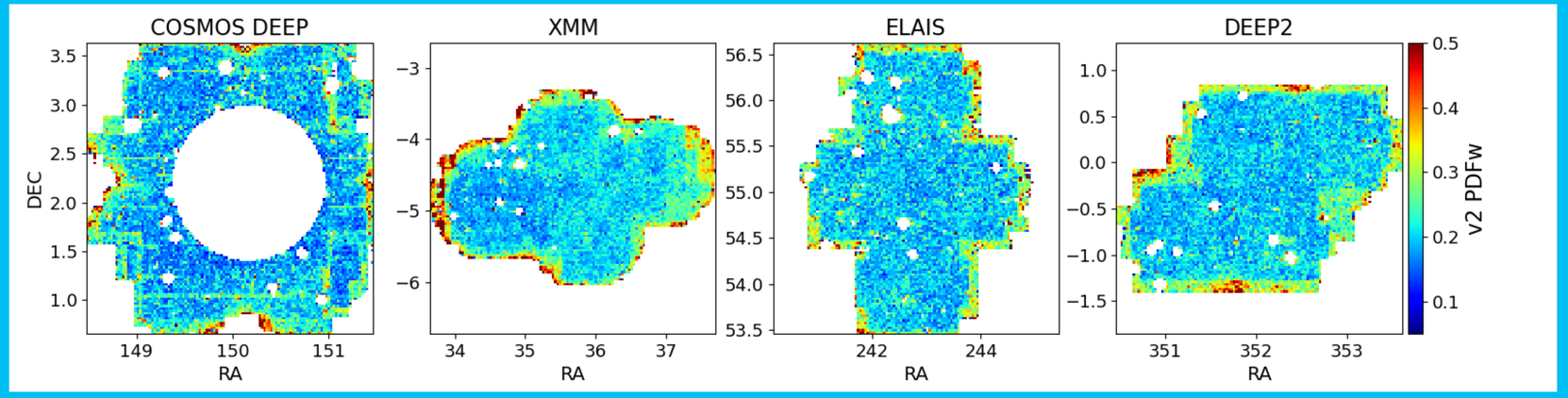


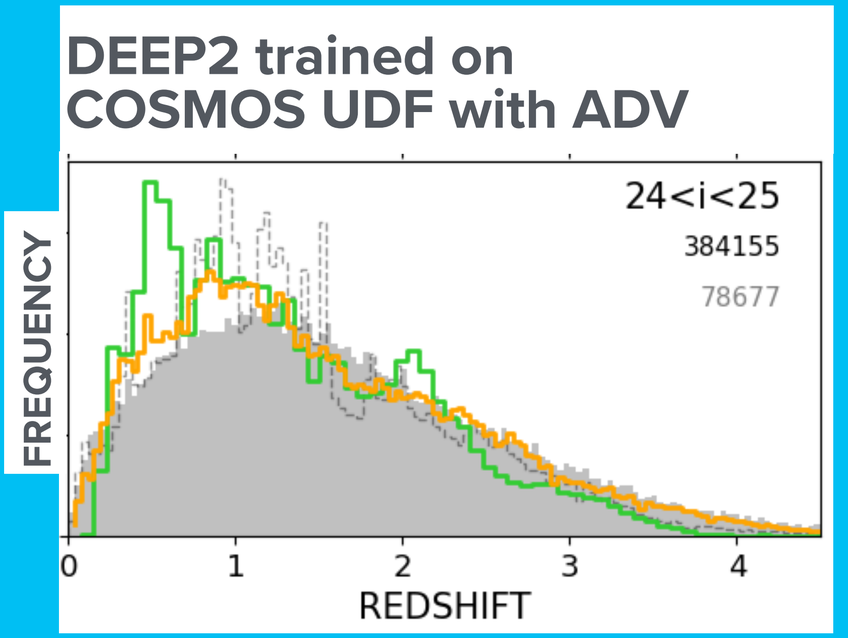
AN EXAMPLE OF SOLUTION TO THE DOMAIN ADAPTATION PROBLEM

BEFORE



AFTER





BEYOND CNN: TRANSFORMERS

CNNs use fixed-size kernels, limiting their ability to capture global context.

Attention mechanisms are a revolutionary DL concept that allows models to focus on specific parts of the input data and to capture dependencies between all elements (mimic human behavior while reading a sentence or looking at an image).

Transformers use self-attention to assign weights to different parts of the input, dynamically adjust these weights based on the context (backbone of language models).

Visual Transformers (e.g., ViT) apply self-attention to images by splitting images into patches, like words in a sentence.

Photo-z estimation: compute attention scores to determine which features or combinations of features are most important for predicting the redshift

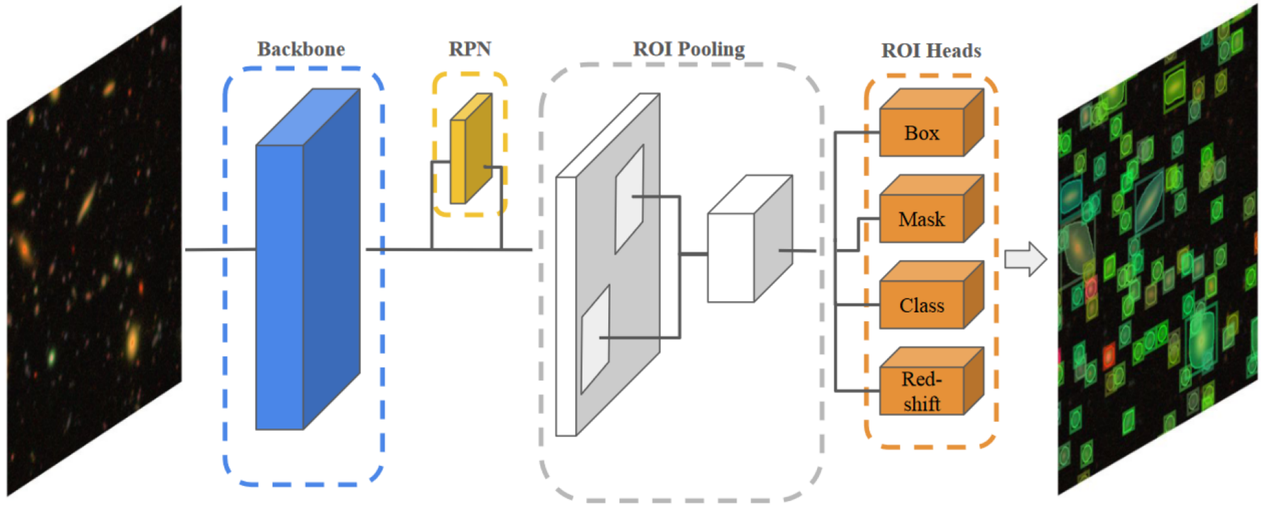
Output: A weighted sum of the features, which are then used to predict the redshift.

APPLICATIONS

DeepDISC

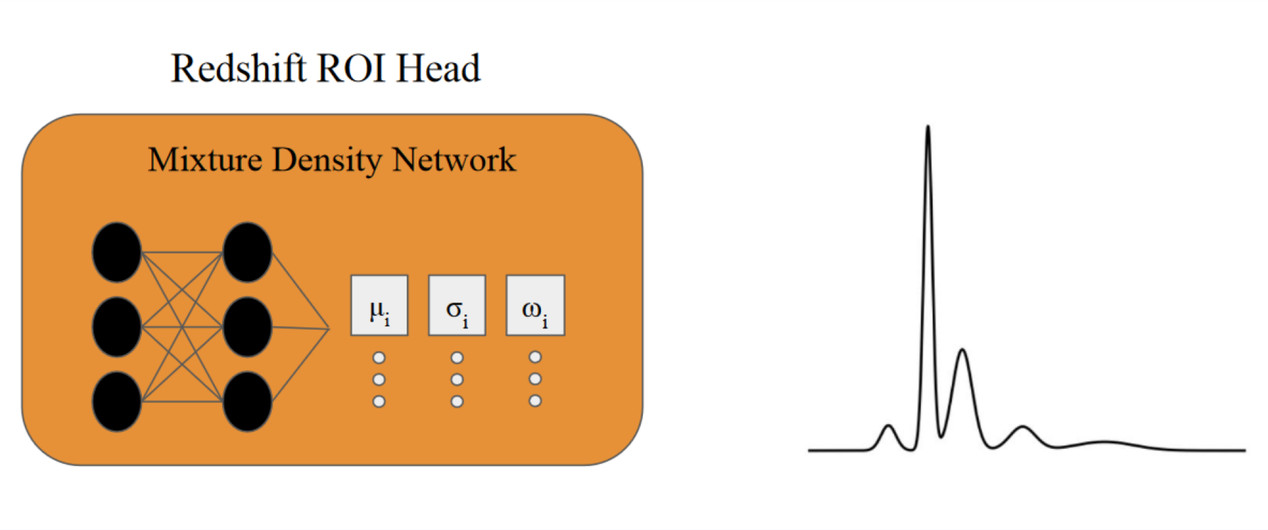
Merz+2023

Deep Detection, Instance Segmentation, Classification



Use resources from DETECTRON2 (library of next-gen object detection and segmentation models compiled by Facebook AI Research).
one main limitation: a deblended ground truth must be provided during training in order to detect and segment deblended objects.

DeepDISC photo-z: attaches a MDN to output redshift PDFs, outperforming standalone CNN approaches on simulated LSST images (Merz+2025).



APPLICATIONS

astroCLIP
(Parker+2024)

A self-supervised, transformer-based foundation model* that simultaneously embeds galaxy images and spectra into a shared, physically meaningful latent space.

1. Single-modal pretraining using SSL technique
2. The pre-trained image and spectrum encoders are aligned into a unified latent space via a contrastive learning objective similar to CLIP (Contrastive Language-Image Pre-training)

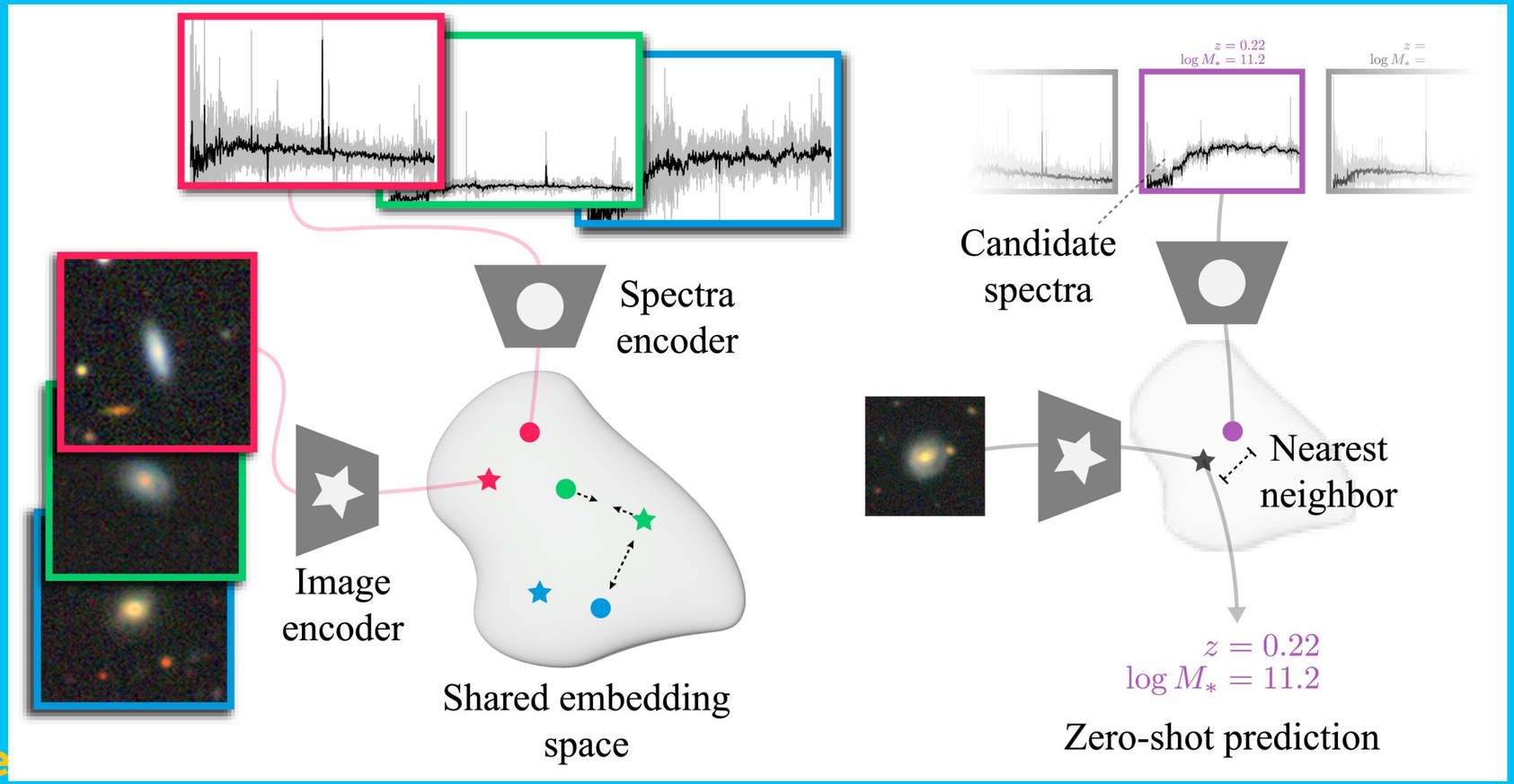


Photo-z estimation: performed by zero-shot k-NN or few-shot MLP regression directly on the embeddings, showing competitive or superior results to supervised baselines

*foundation model = large, general-purpose network pre-trained on vast and diverse datasets (e.g. images, spectra, catalog data) and then fine-tuned or adapted for more specific tasks → a major shift compared to single-purpose supervised models

CONCLUSIONS

Estimating photo-z with AI methods is a booming field, motivated by ongoing and forthcoming XL missions (*Euclid*, *Nancy Grace Roman Space Telescope*, *Vera C. Rubin Observatory)*, and piggybacking on the *architectures, methods, and software infrastructure* that have been/are being developed and scaled primarily for the language model revolution.

The future?

- methods that reduce reliance on large, perfect spec-z training sets;
- methods that explain the black box: like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations), both model-agnostic methods designed to provide interpretable measure of feature influence

Learn from https://www.fidle.cnrs.fr/