DeepUQ: Assessing the Aleatoric Uncertainties from two Deep Learning Methods

Rebecca Nevin[1], Aleksandra Ćiprijanović[1,3], Brian Nord[1,2,3]
[1]Fermilab National Accelerator Laboratory, [2]Kavli Institute for Cosmological Physics, University of Chicago, [3]Department of Astronomy and Astrophysics, University of Chicago

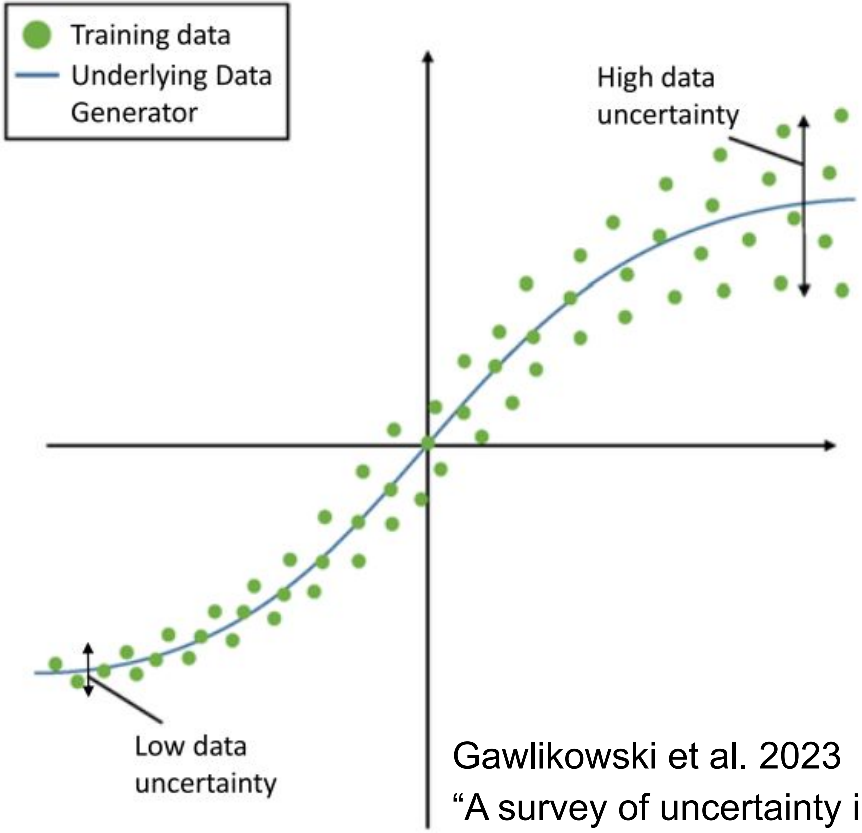
DEEP SKIES
Bringing Artificial Intelligence to Astrophysics

arXiv

paper repo

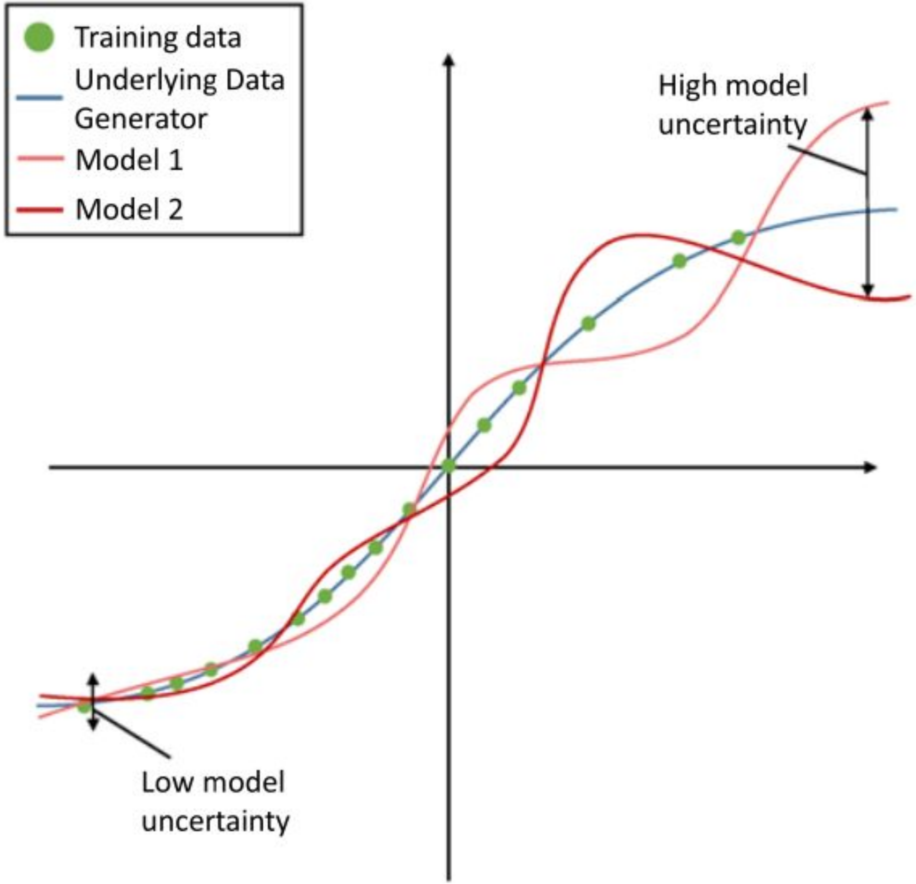
DeepUQ pypi

I focus on **aleatoric** (data) uncertainty

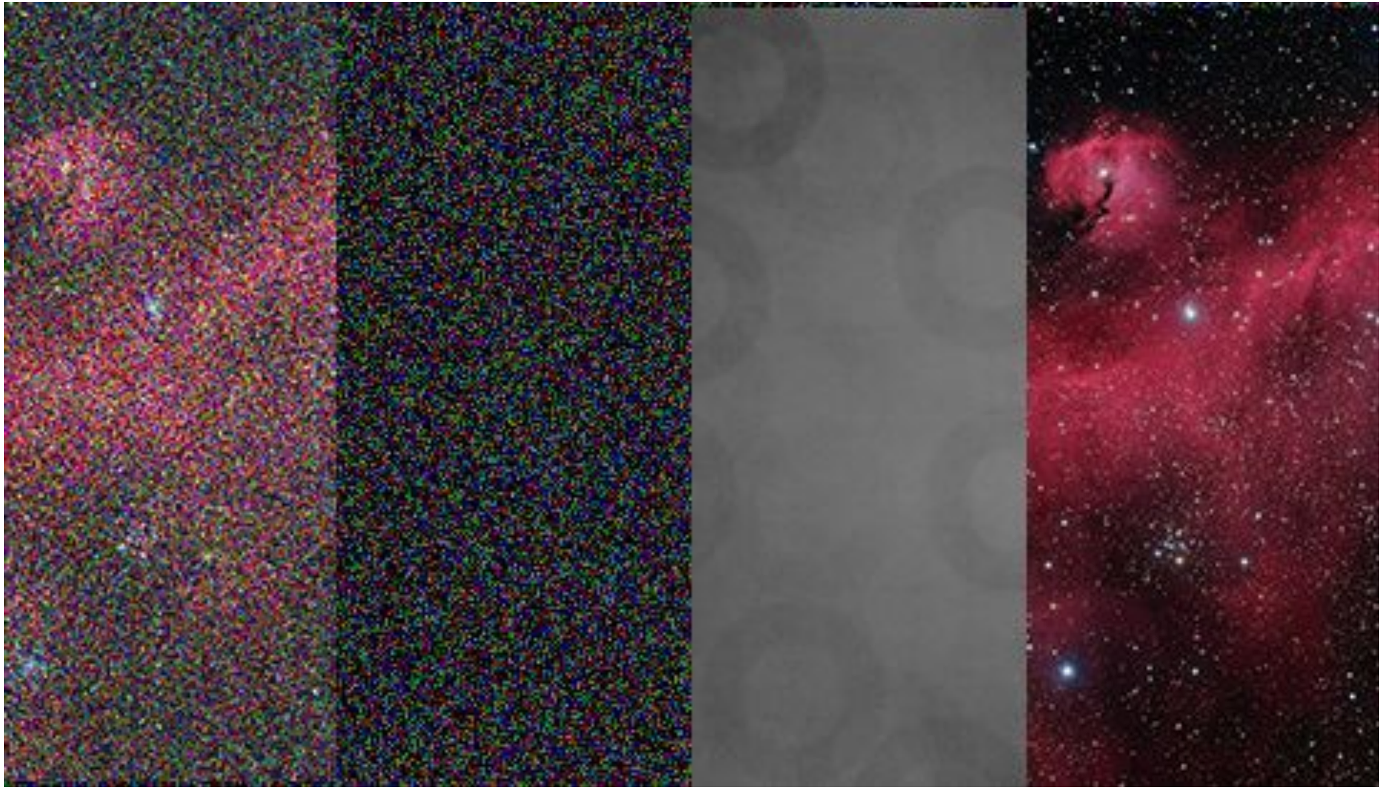


Gawlikowski et al. 2023
“A survey of uncertainty in deep neural networks”

Other work focuses on **epistemic** (model) uncertainty

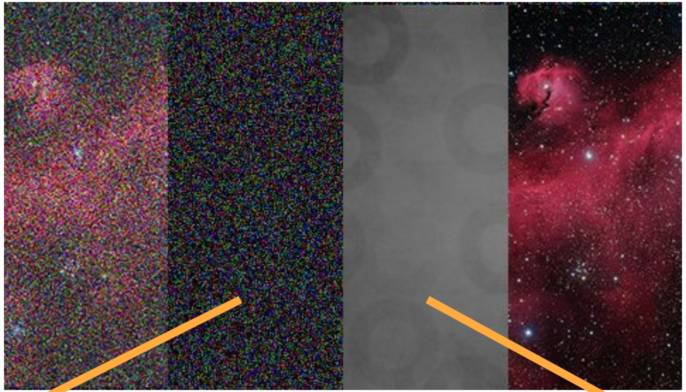


Aleatoric uncertainty is important in many physics and astrophysics applications, i.e., Poisson or Gaussian noise in astrophysics

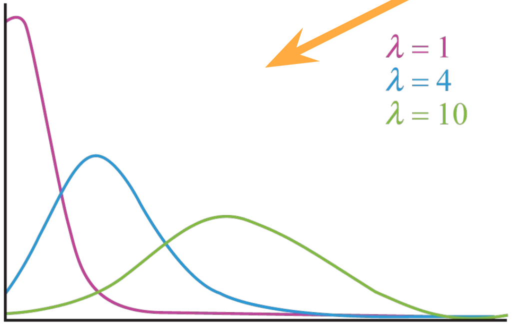


Sky and Telescope: Richard Wright

Deep learning or non deep learning methods *should* predict uncertainties that match these known distributions

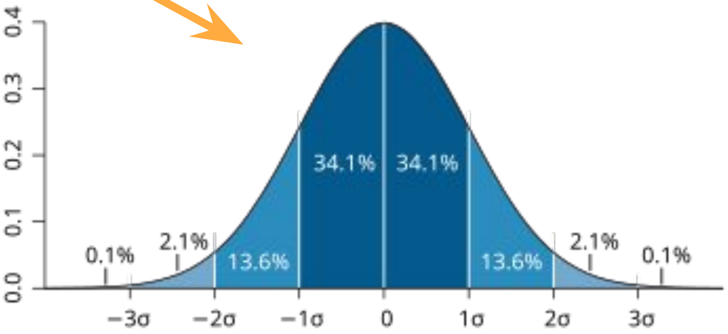


Poisson



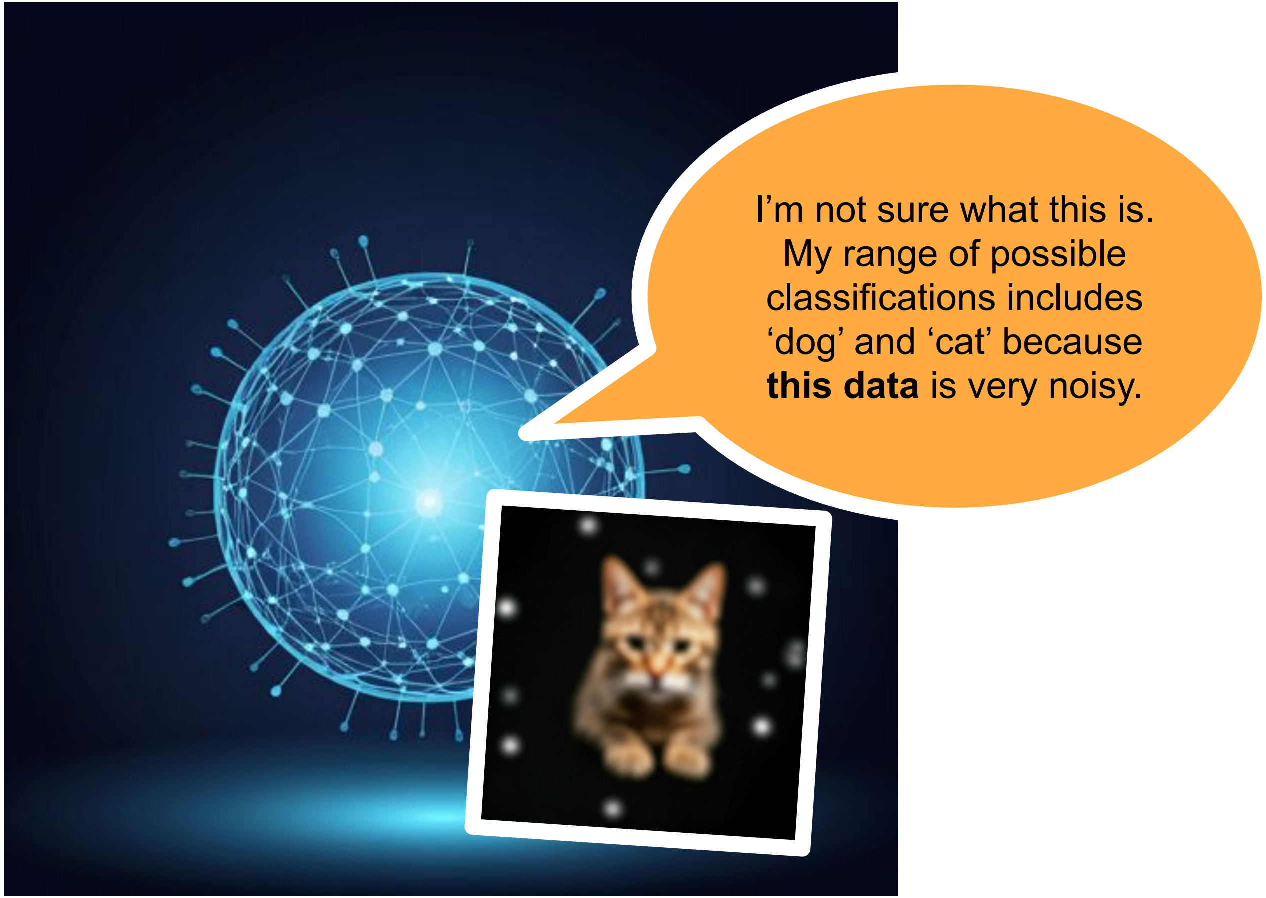
Calcworkshop.com

Gaussian / Normal

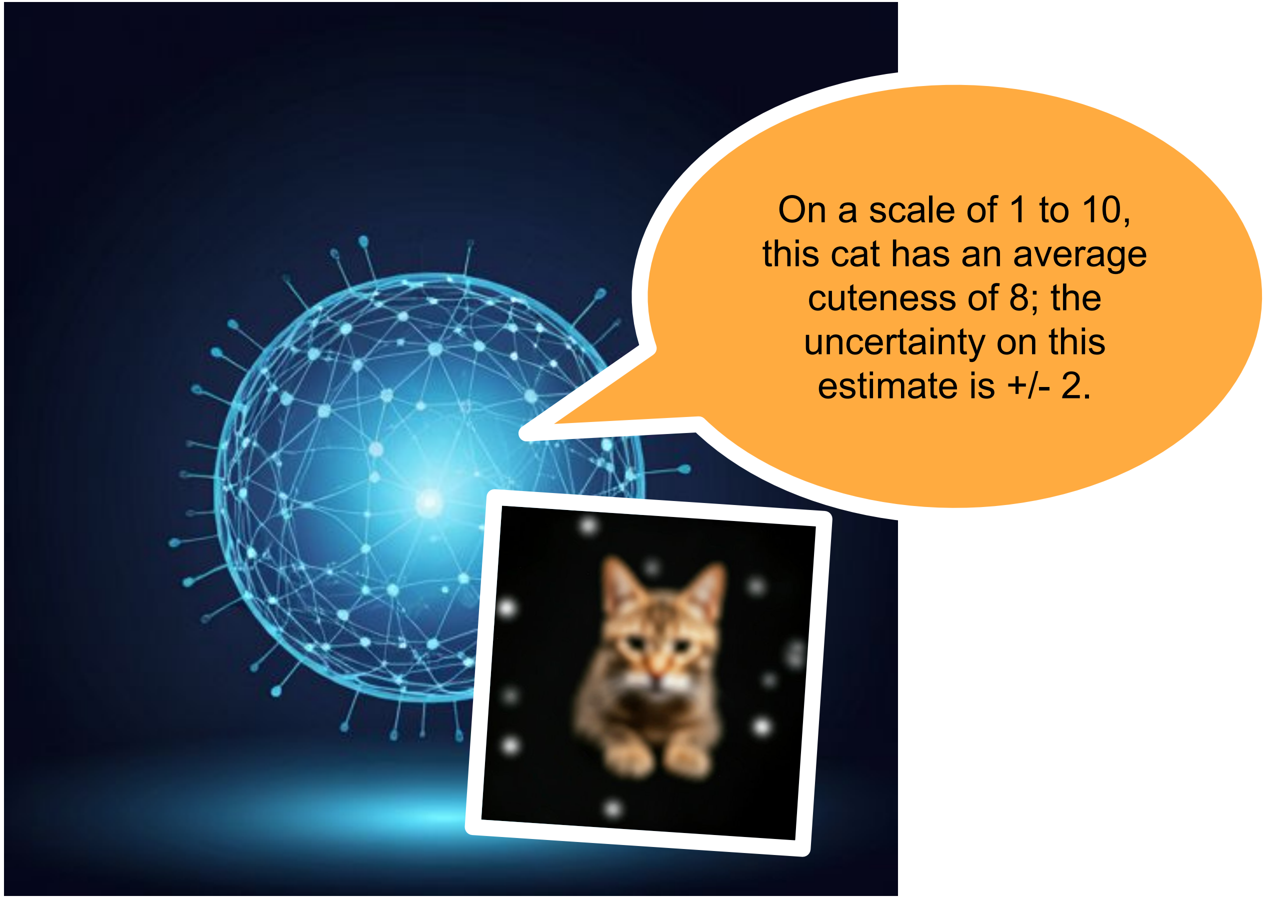


Gen AI image

Gen AI image

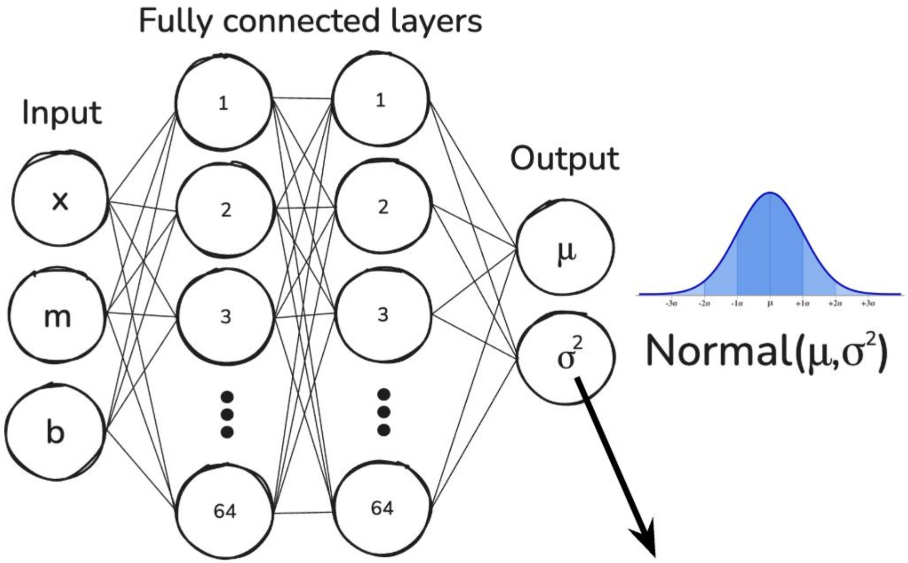

I’m not sure what this is.
My range of possible
classifications includes
‘dog’ and ‘cat’ because
this data is very noisy.

Gen AI image

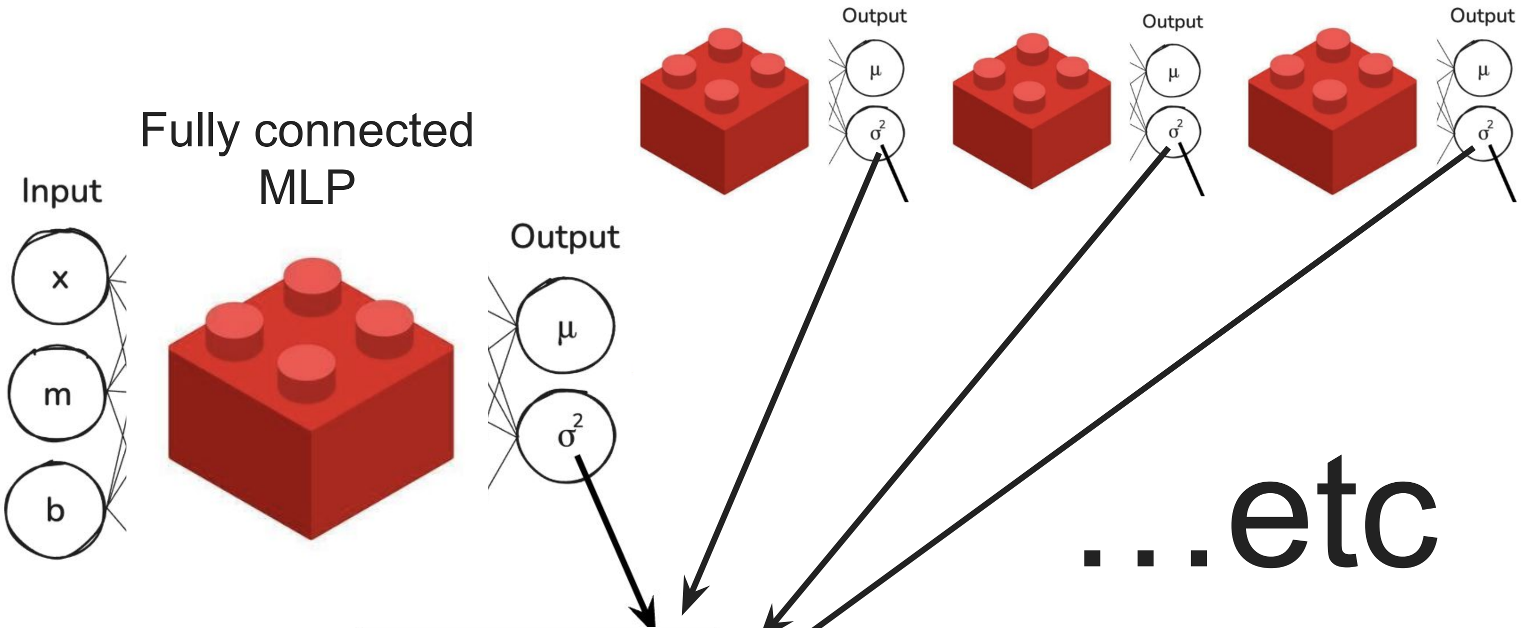

On a scale of 1 to 10, this cat has an average cuteness of 8; the uncertainty on this estimate is +/- 2.

Gen AI image

Mean variance estimation networks (MVEs) predict **aleatoric uncertainty** via their two output nodes (mean and variance)

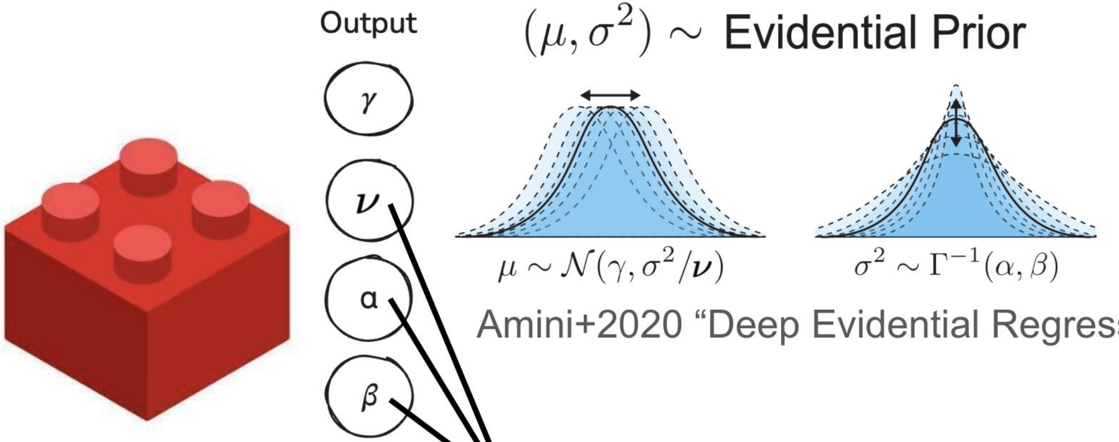


The aleatoric uncertainty for a Deep Ensemble (of many MVEs) is the average of the predicted standard deviations



$$\sigma_{\mathrm{al}} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \sigma_k^2}$$

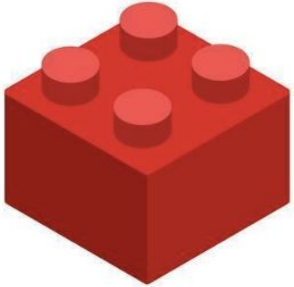
Deep Evidential Regression predicts aleatoric uncertainty using a normal-inverse-gamma loss

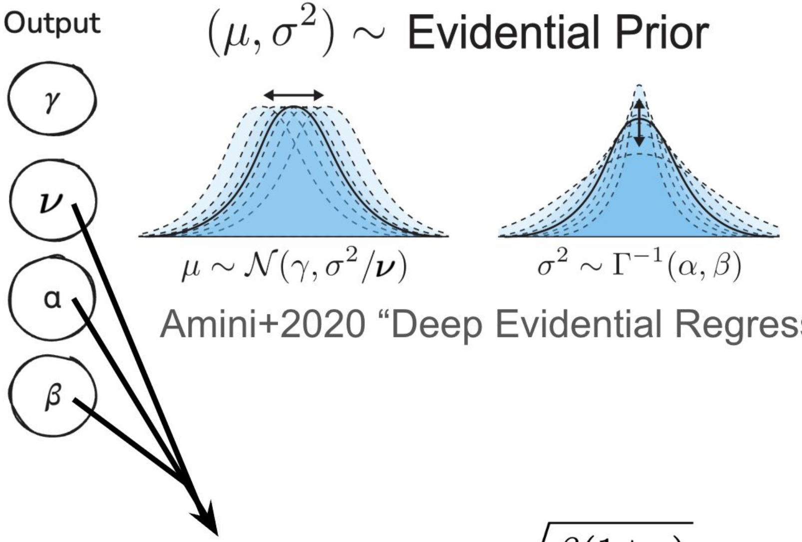


Amini+2020 “Deep Evidential Regression”

$$\sigma_{\mathrm{al}} \equiv w_{\mathrm{St}} = \sqrt{\frac{\beta(1+\nu)}{\alpha\nu}}$$

Deep Evidential Regression predicts aleatoric uncertainty using a normal-inverse-gamma loss



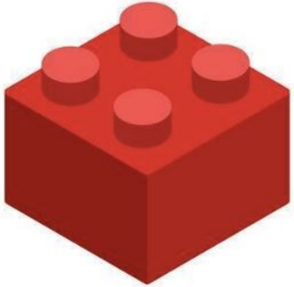


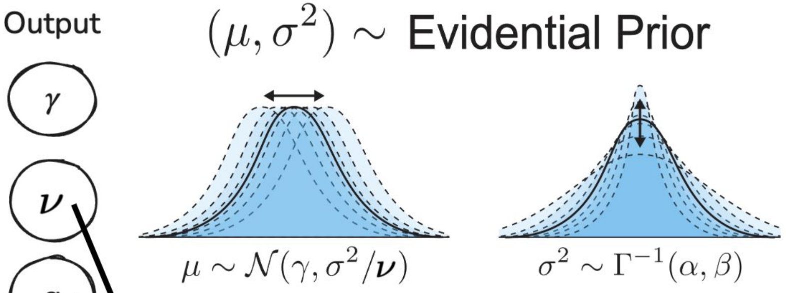
Amini+2020 “Deep Evidential Regression”

$$\sigma_{\mathrm{al}} \equiv w_{\mathrm{St}} = \sqrt{\frac{\beta(1+\nu)}{\alpha\nu}}$$

Deep Evidential Regression predicts aleatoric uncertainty using a normal-inverse-gamma loss

distribution distribution





Amini+2020 “Deep Evidential Regression”

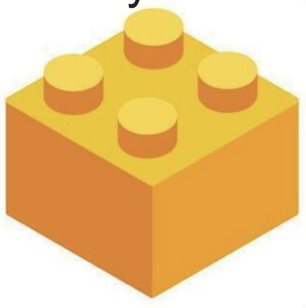
Deep Evidential Regression

$$\sigma_{\mathrm{al}} \equiv w_{\mathrm{St}} = \sqrt{\frac{\beta(1+\nu)}{\alpha\nu}}$$

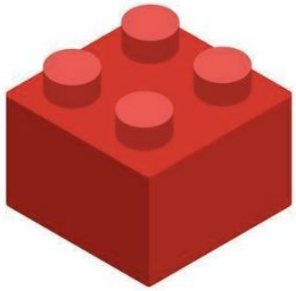
Deep Evidential Regression predicts aleatoric uncertainty using a normal-inverse-gamma loss

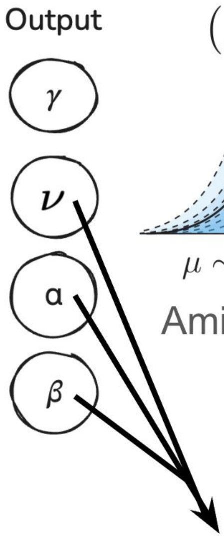
Inception gif?

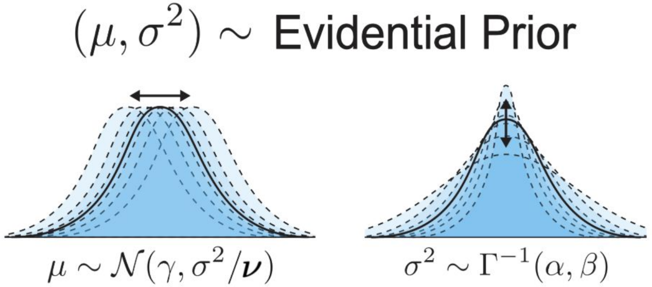
Convolutional Layers



Fully connected MLP

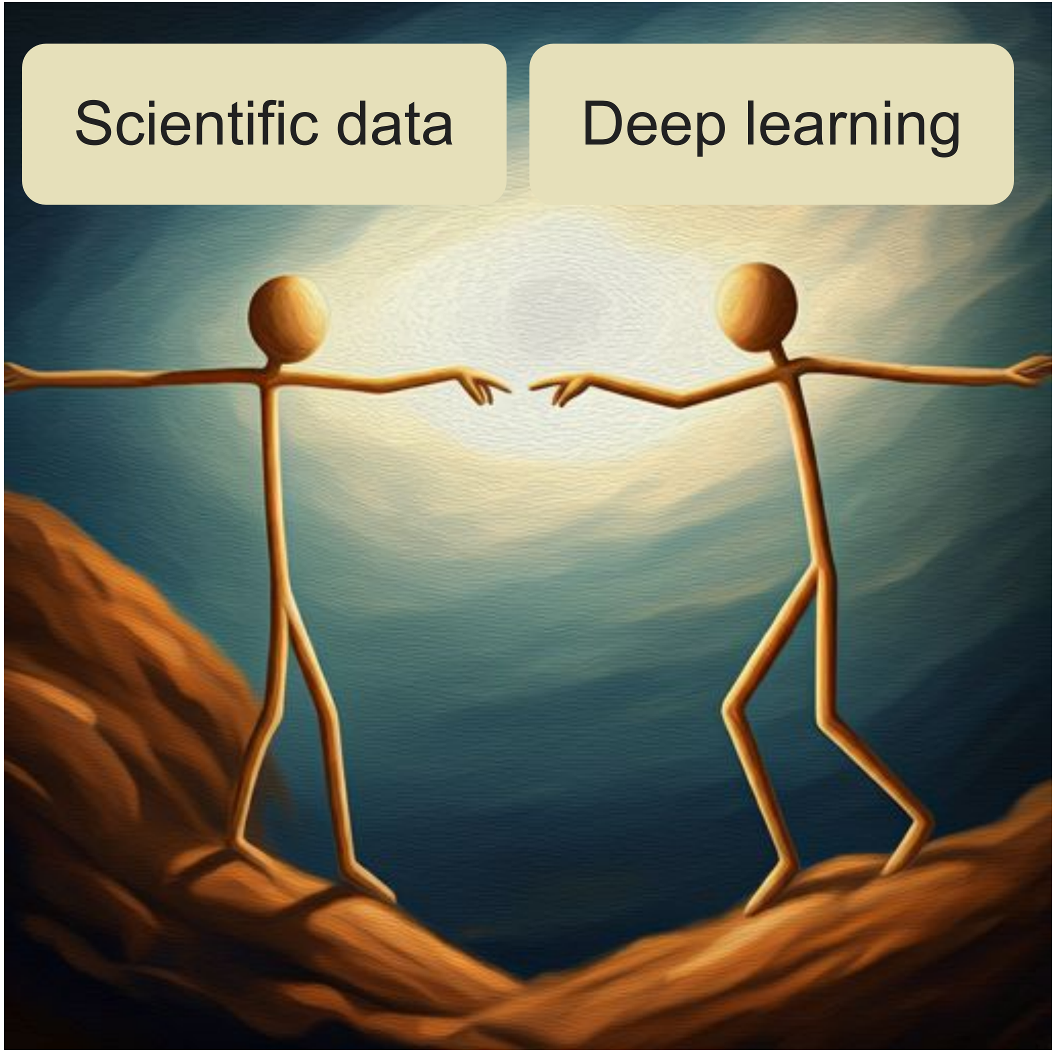






Amini+2020 “Deep Evidential Regression”

$$\sigma_{\mathrm{al}} \equiv w_{\mathrm{St}} = \sqrt{\frac{\beta(1+\nu)}{\alpha\nu}}$$



Gen AI image

Other work offers comparisons of different UQ techniques

Compare aspects of predictive uncertainty distributions (but not the exact uncertainty value):
Scalia et al. 2019 “Evaluating Scalable Uncertainty Estimation Methods for DNN-Based Molecular Property Prediction.”
Tran et al. 2019 “Methods for comparing uncertainty quantifications for material property predictions.”

A toolbox for comparing UQ methods (but not the exact uncertainty value):
Chung et al. 2021 “Uncertainty Toolbox: an Open-Source Library for Assessing, Visualizing, and Improving Uncertainty Quantification.”

Compares exact aleatoric uncertainties (but not for a variety of data types):
Caldeira & Nord 2020 “Deeply Uncertain: Comparing Methods of Uncertainty Quantification in Deep Learning Algorithms.”

Uses a variety of data types and uncertainty injection (but does not compare exact uncertainty values):
Bramlage et al. 2023. “Plausible uncertainties for human pose regression”

Uncertainty Menu

To generate the 12 total experimental datasets, there are three categories.

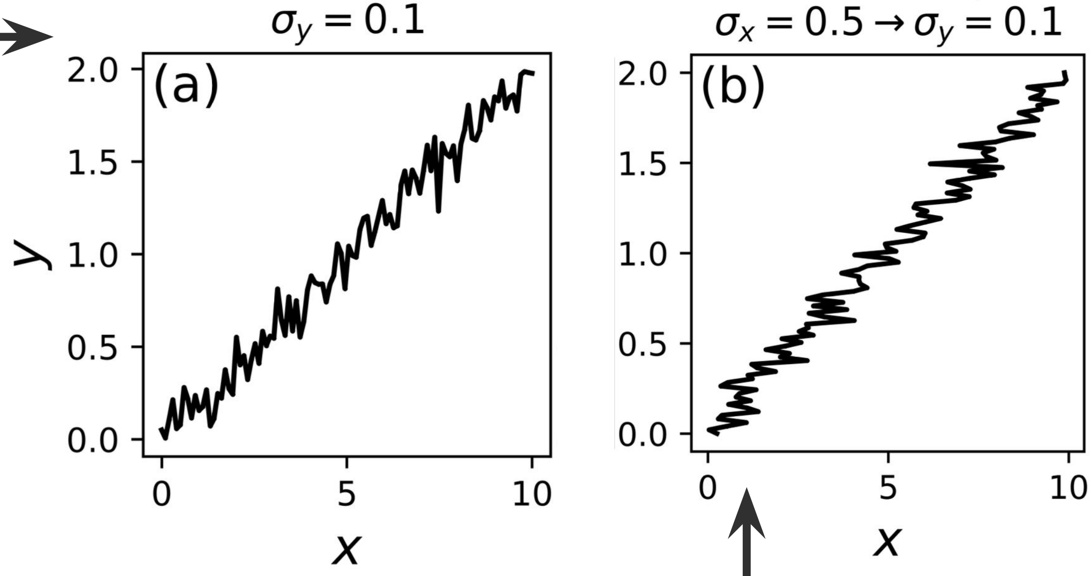
Uncertainty Menu

To generate the 12 total experimental datasets, there are three categories.

<u>Injection</u>

Output variable

Input variable



Uncertainty Menu

To generate the 12 total experimental datasets, there are three categories.

Injection	Dimensionality	Noise level
Output variable	0D: Tabular	Low ($\sigma_Y = 0.01$)
Input variable	2D: Imaging	Medium ($\sigma_Y = 0.05$)
		High ($\sigma_Y = 0.1$)

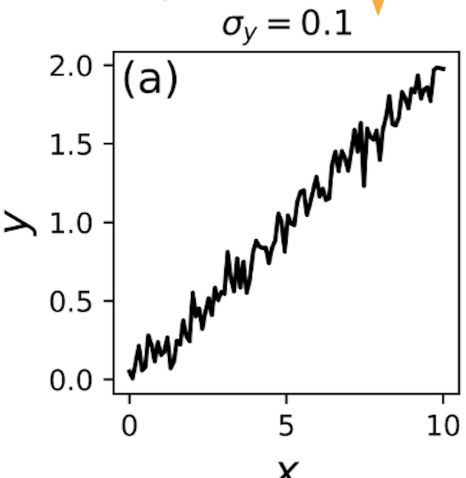
Uncertainty Menu

Below I show two options for selecting from each category.

Injection
- Output variable
- Input variable

Dimensionality
- 0D: Tabular
- 2D: Imaging

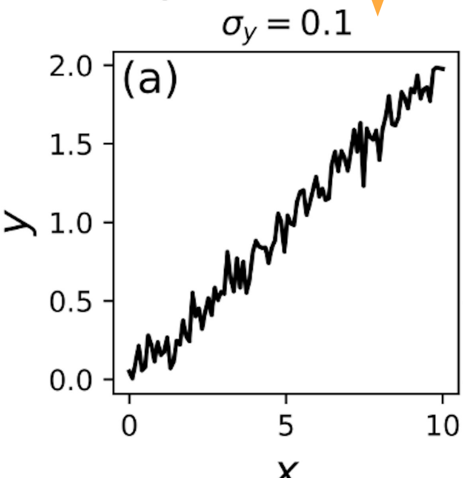
Noise level
- Low (σ_Y = 0.01)
- Medium (σ_Y = 0.05)
- High (σ_Y = 0.1)

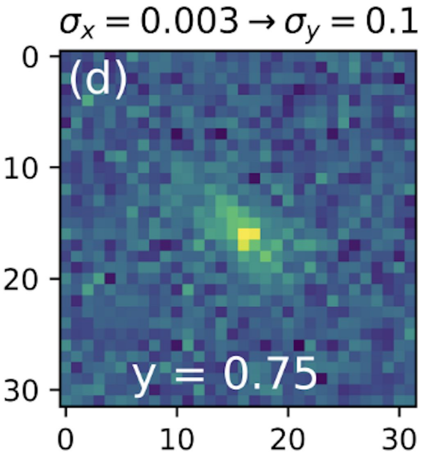


Uncertainty Menu

Below I show two options for selecting from each category.

Injection	Dimensionality	Noise level
Output variable	0D: Tabular	Low (σ_Y = 0.01)
Input variable	2D: Imaging	Medium (σ_Y = 0.05)
		High (σ_Y = 0.1)





I use `DeepBench` to generate the galaxy images.

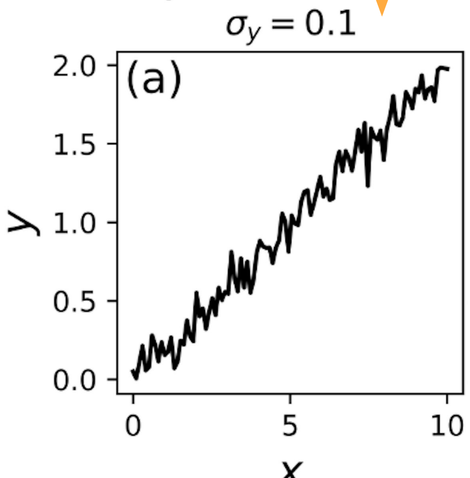
Uncertainty Menu

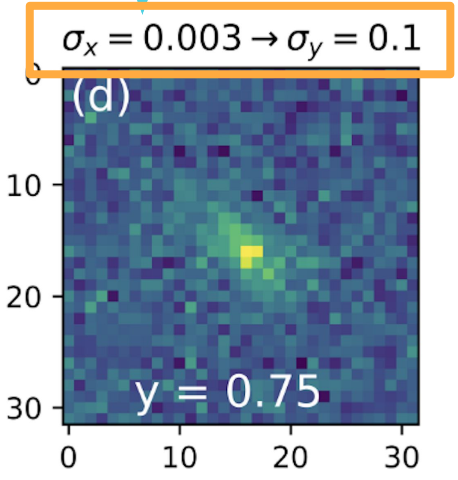
For uncertainty on the input variable, I inject the uncertainty directly on the input and propagate it to the output variable.

Injection
- Output variable
- Input variable

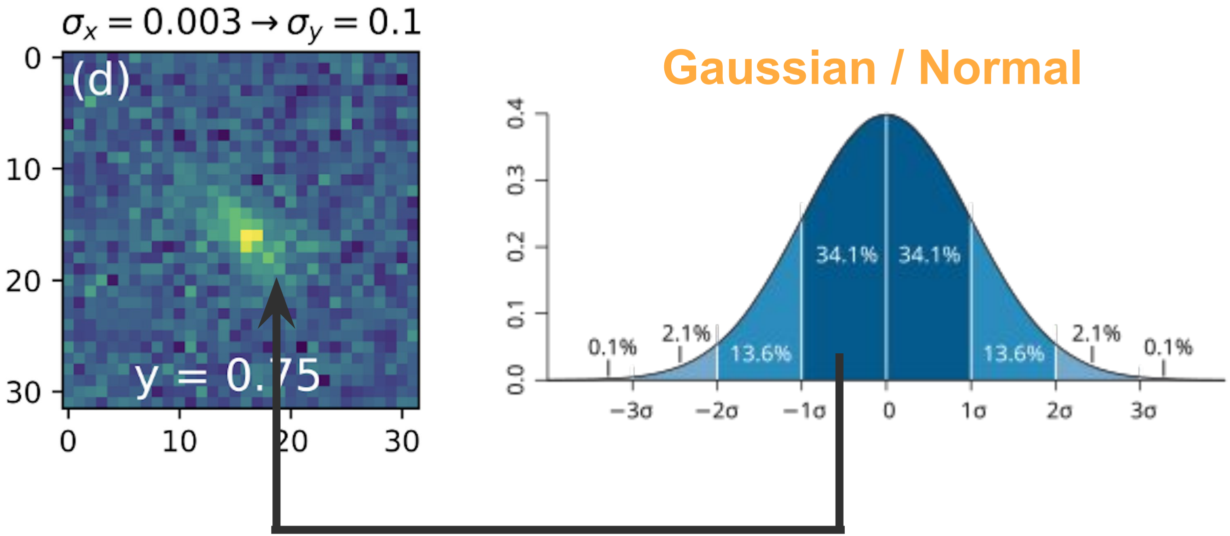
Dimensionality
- 0D: Tabular
- 2D: Imaging

Noise level
- Low (σ_Y = 0.01)
- Medium (σ_Y = 0.05)
- High (σ_Y = 0.1)





The uncertainty is injected for all data via a homoskedastic Gaussian distribution



i.e., the uncertainty is added to each pixel via a draw from a random normal with standard deviation σ_X = 0.1

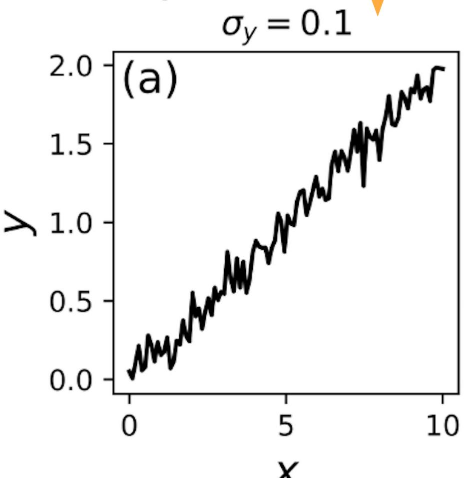
Uncertainty Menu

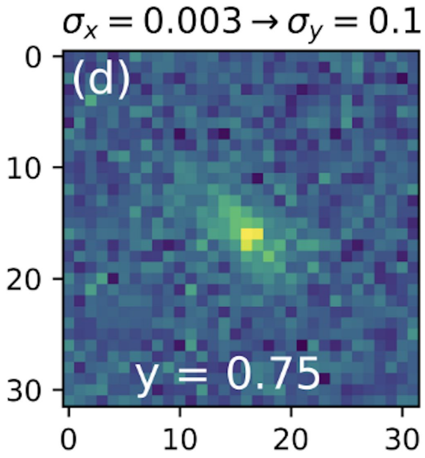
Below I show two options for selecting from each category.

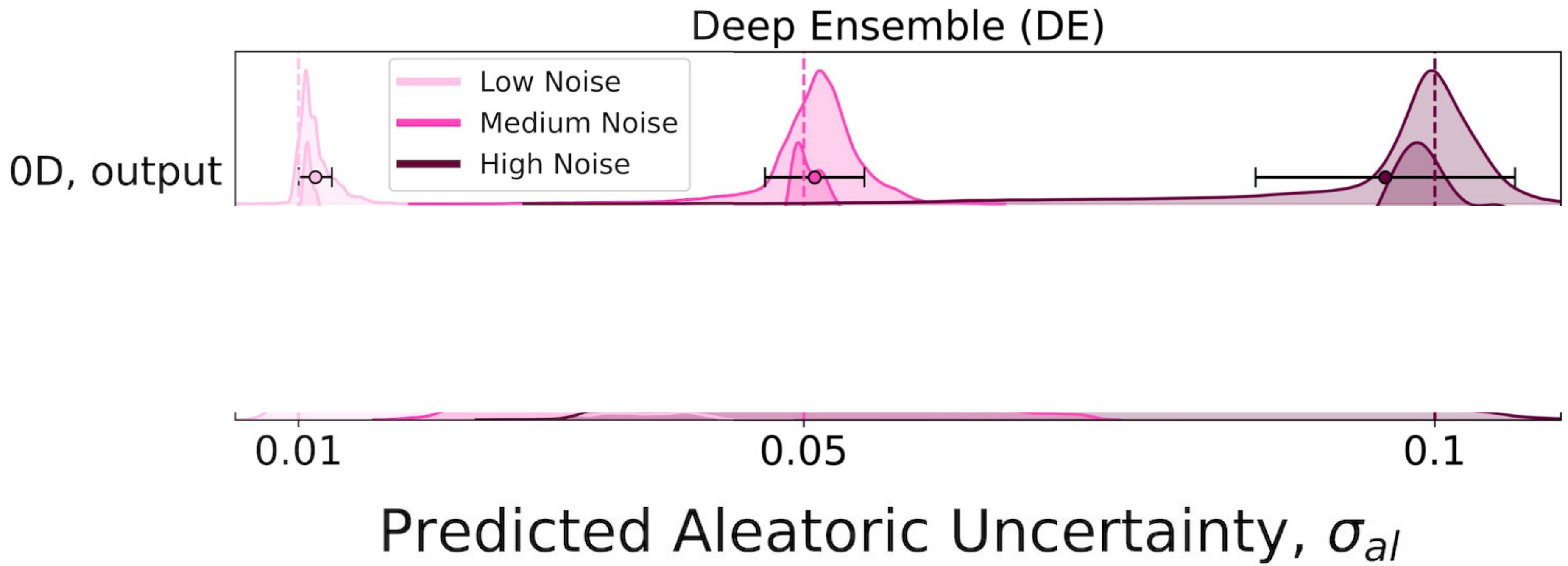
Injection
- Output variable
- Input variable

Dimensionality
- 0D: Tabular
- 2D: Imaging

Noise level
- Low (σ_Y = 0.01)
- Medium (σ_Y = 0.05)
- High (σ_Y = 0.1)

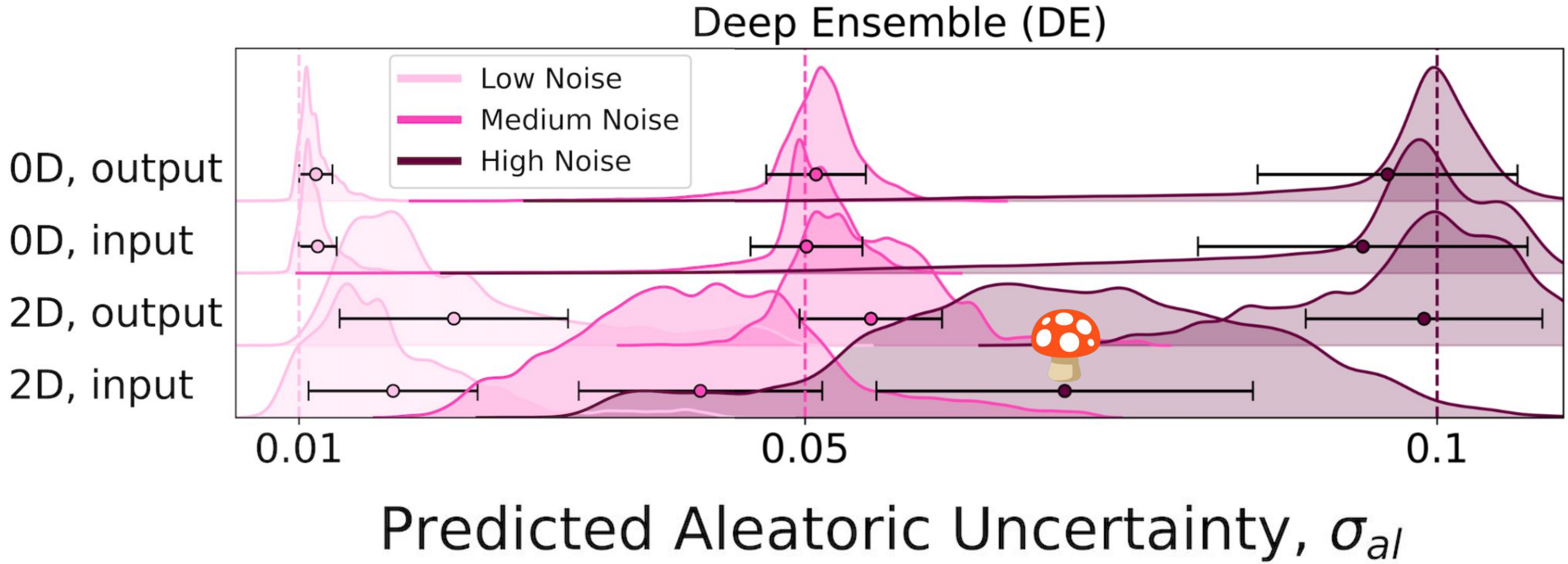




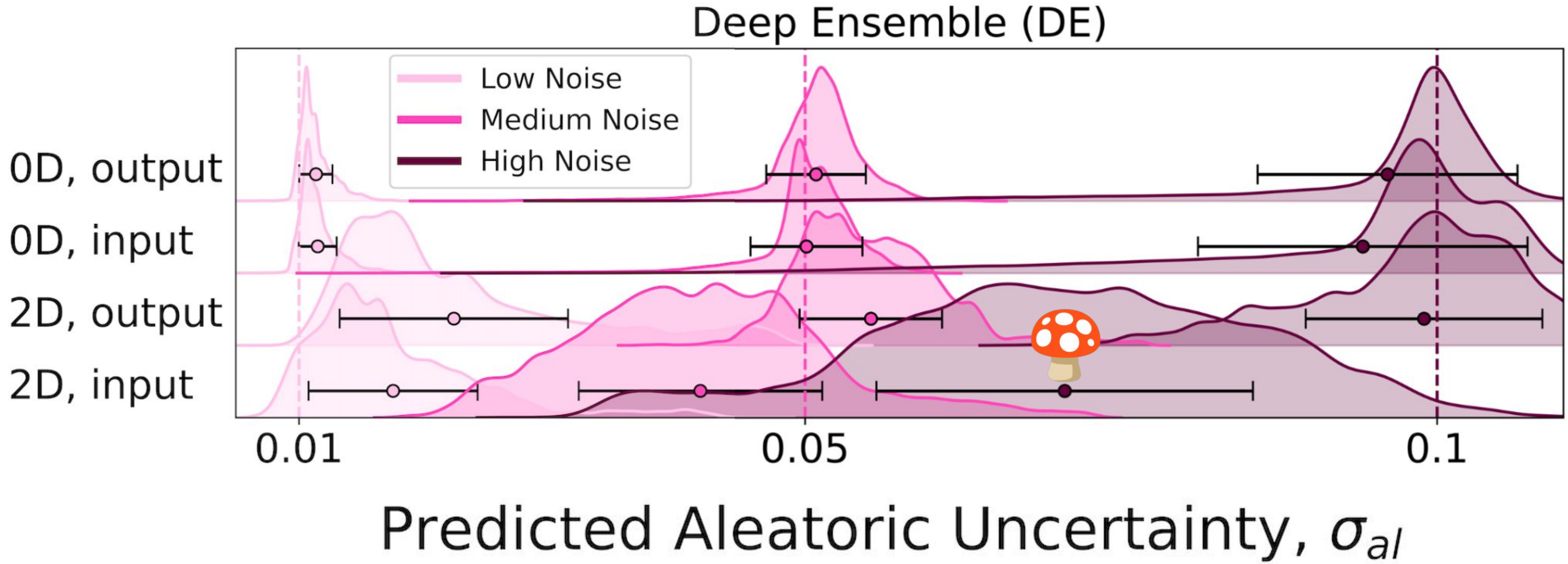


Deep Ensemble (DE)
Low Noise
Medium Noise
High Noise
0D, output
0.01
0.05
0.1
Predicted Aleatoric Uncertainty, σ_{al}

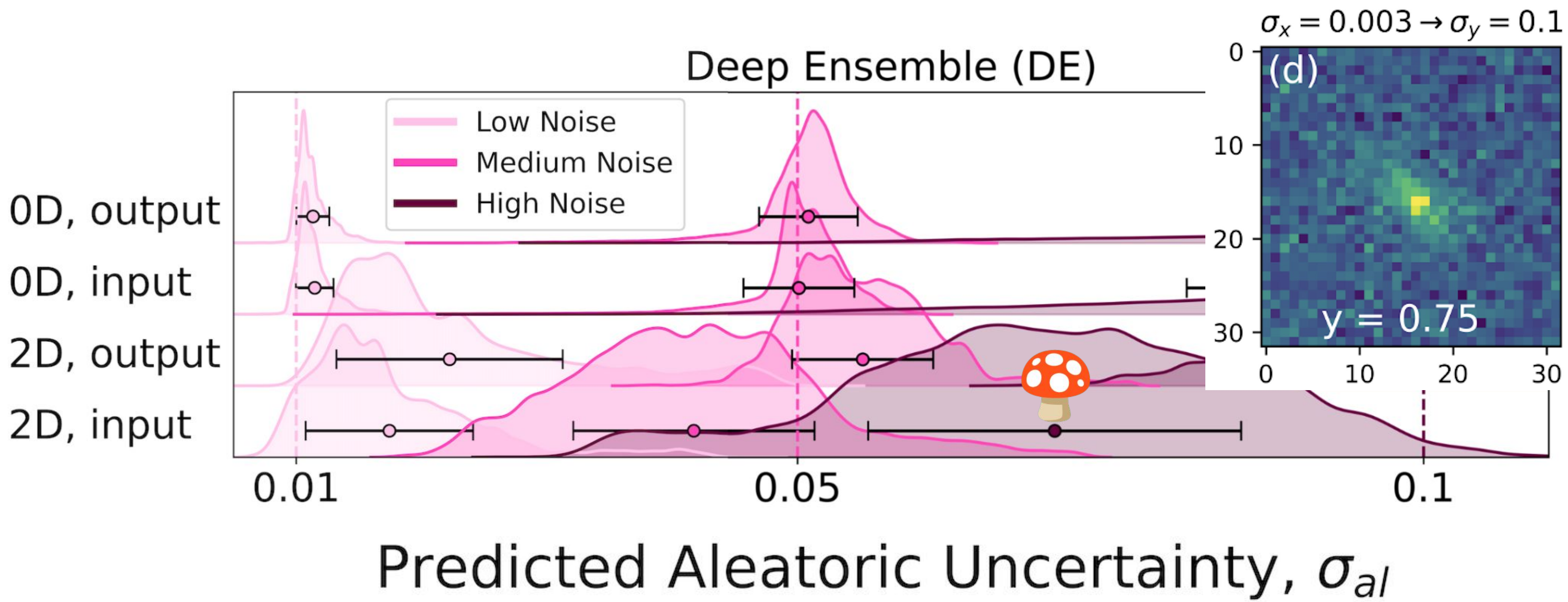
5 out of 12 experiments are miscalibrated for the Deep Ensemble



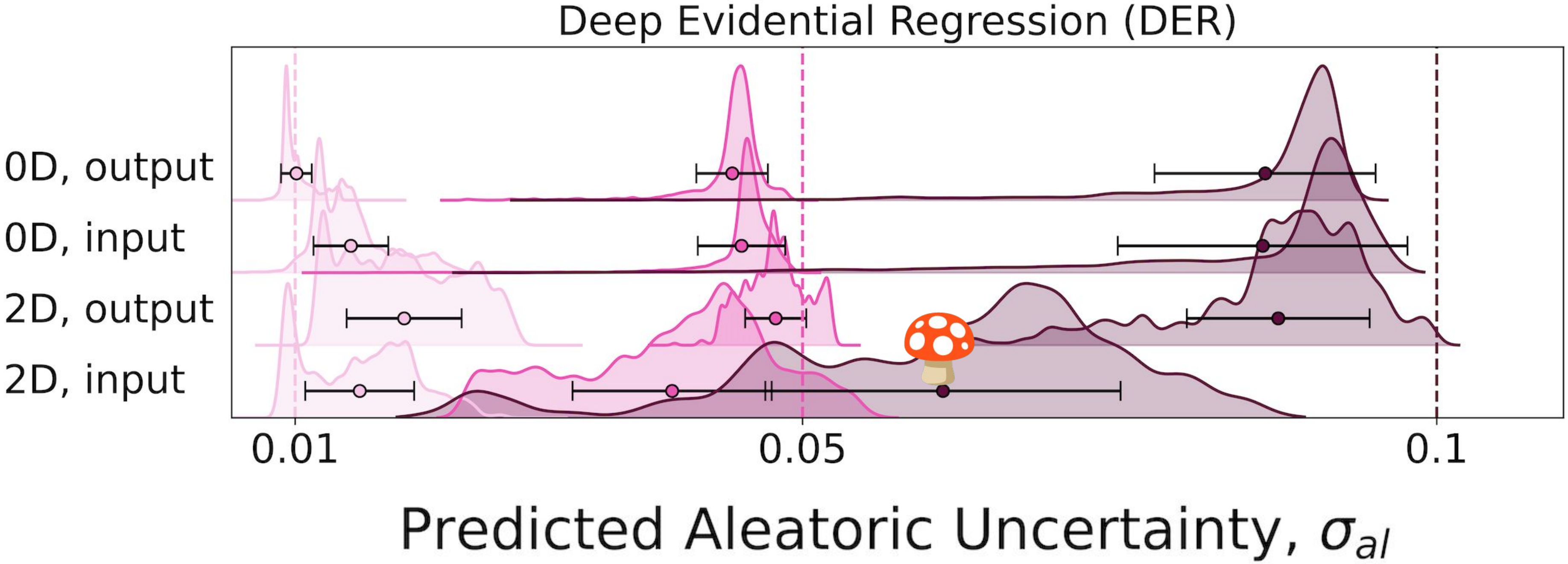
5 out of 12 experiments are miscalibrated for the Deep Ensemble



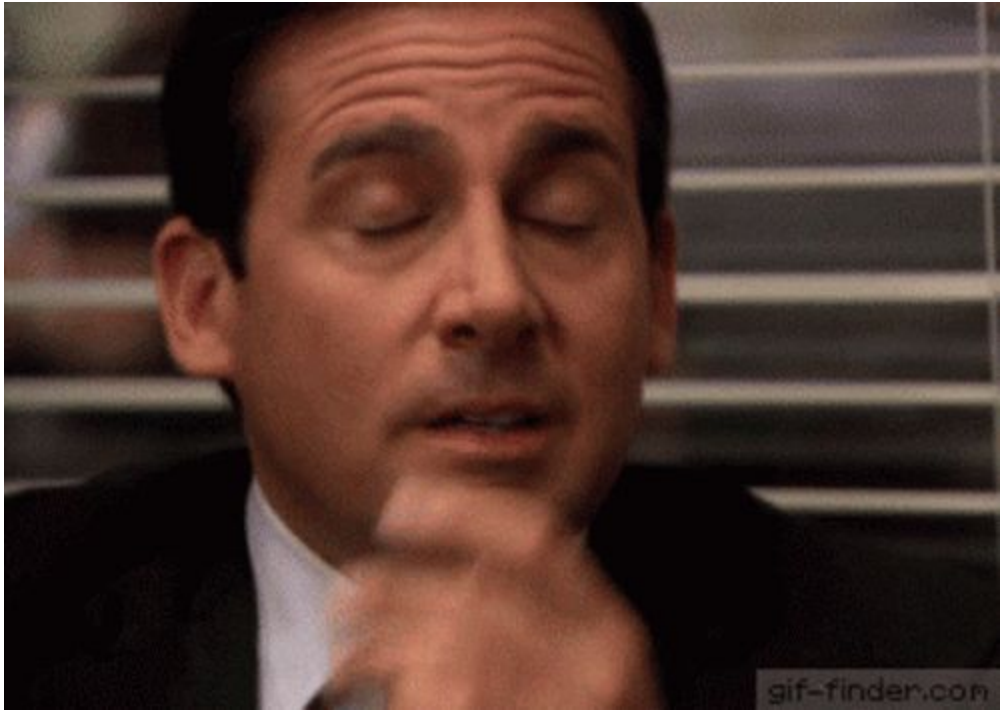
5 out of 12 experiments are miscalibrated for the Deep Ensemble



10 out of 12 experiments are miscalibrated for the Deep Evidential Regression

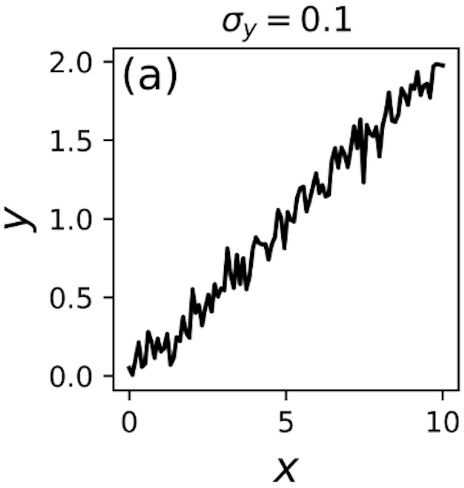


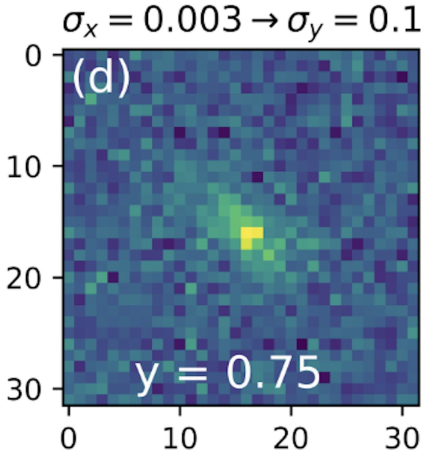
Both models are **overconfident** in most experiments; the Deep Evidential Regression is slightly worse



This problem is worse for higher dimensional (images) and higher noise data!

Caveat: All the results presented here apply only to the (simplistic) set of experiments



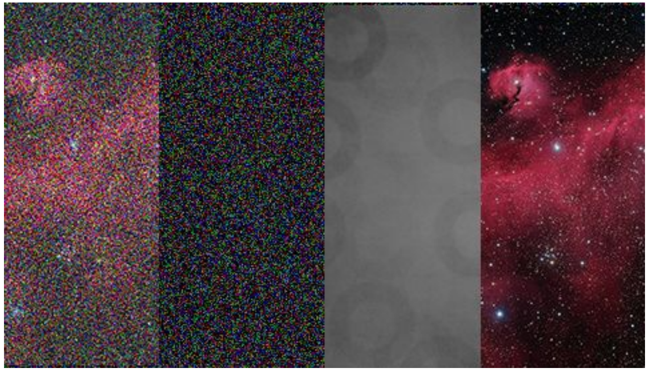


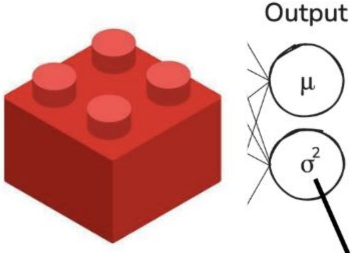
Real world data can be even messier and more uncertain

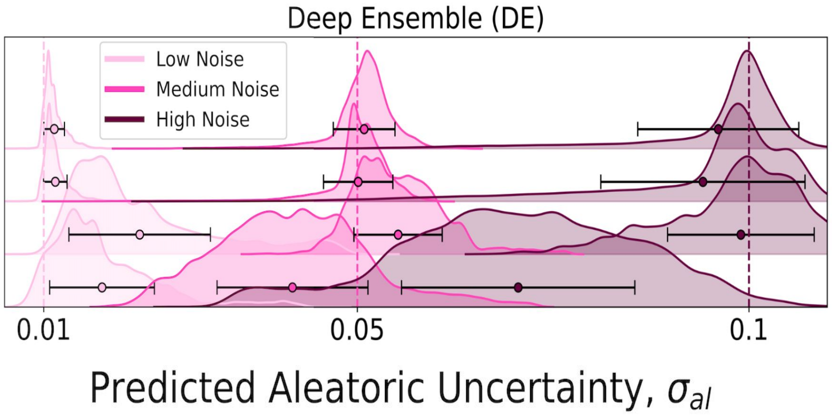
Credit: Science: NASA, ESA, CSA, Tommaso Treu (UCLA); Image Processing: Zolt G. Levay (STScI)

ATLAS Collaboration, CERN
Particle data are in tabular format

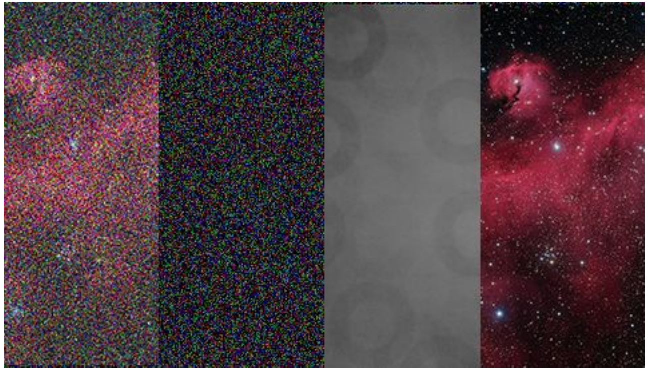
Conclusion: Scientific imaging and other datasets offer a great opportunity to test UQ methods (DE and DER); we find that they are mostly miscalibrated in aleatoric uncertainty prediction for this set of experiments.

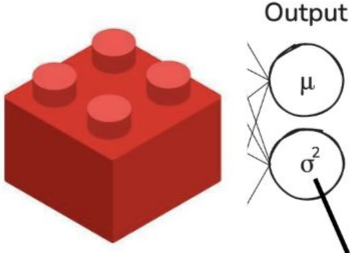


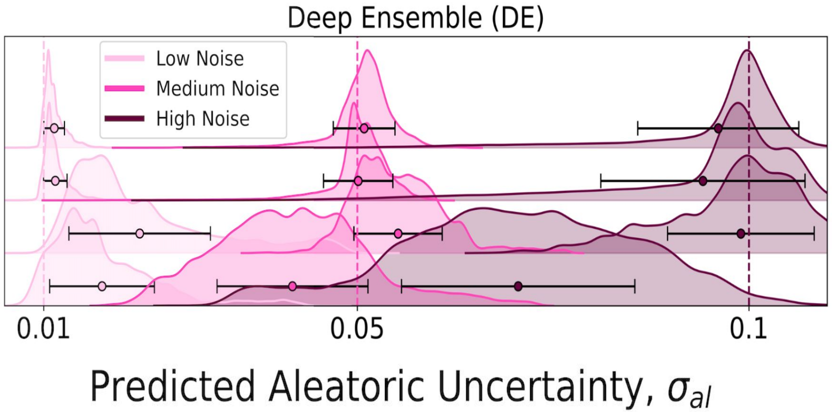




Conclusion: Scientific imaging and other datasets offer a great opportunity to test UQ methods (DE and DER); we find that they are mostly miscalibrated in aleatoric uncertainty prediction for this set of experiments.







arXiv

paper repo

DeepUQ pypi

Bonus slides

Concerns for the field of UQ:

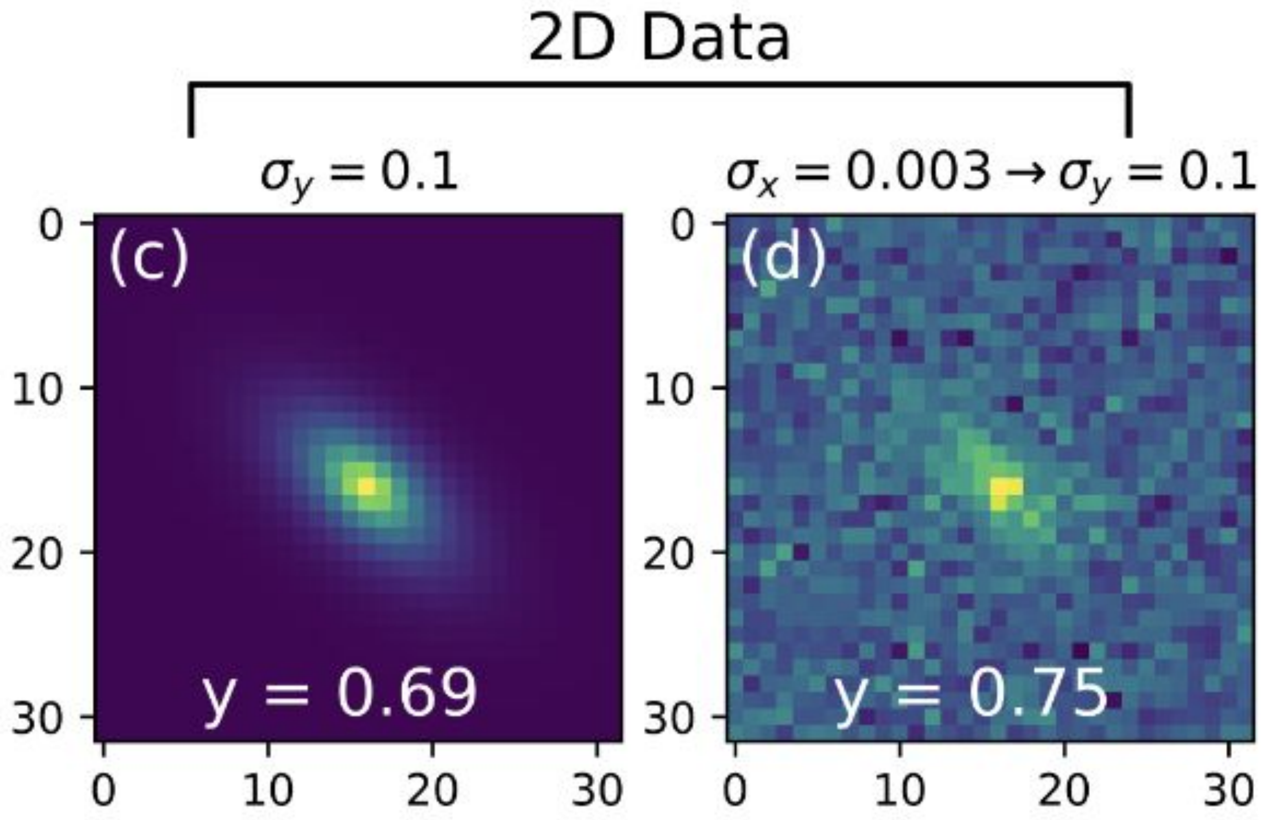
- Taxonomies are confusing/conflicting, how do we define different types of uncertainties? Aleatoric, epistemic, oh my!

- Independence of uncertainty types should be questioned (is aleatoric independent from epistemic? Is there more overlap that we're currently considering?)

- Are notions of uncertainty in physics/astronomy/science aligning with the deep learning science on uncertainty quantification? What work is to be done?

What do we need?

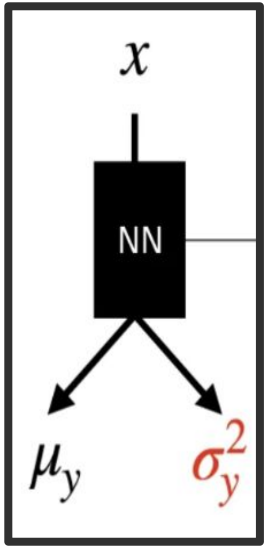
- Standardized datasets with known uncertainties to test the performance of these UQ methods
- ^More complex versions of this
- Expanding this sort of work to epistemic uncertainties

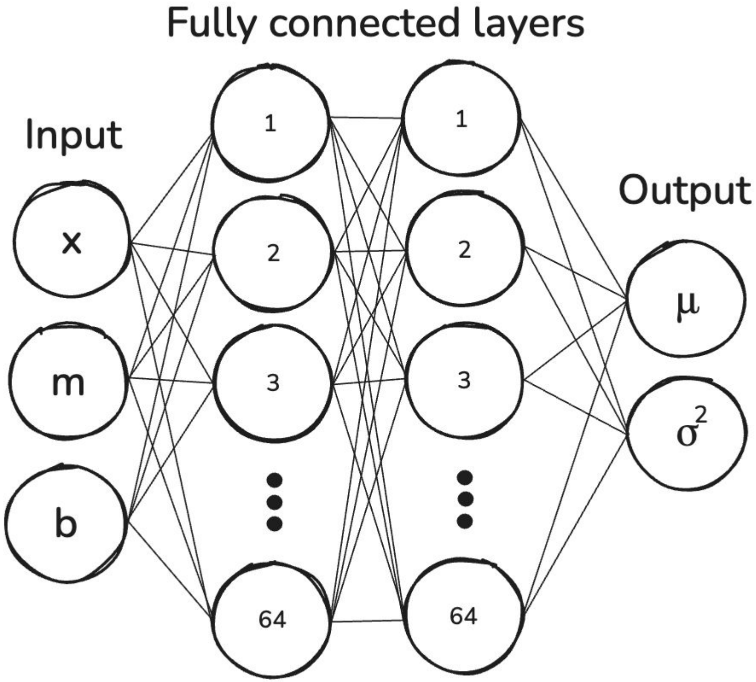


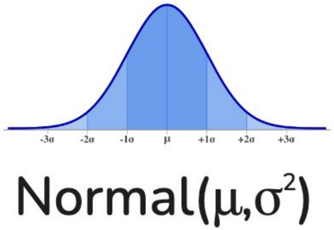
Output variable

Input variable

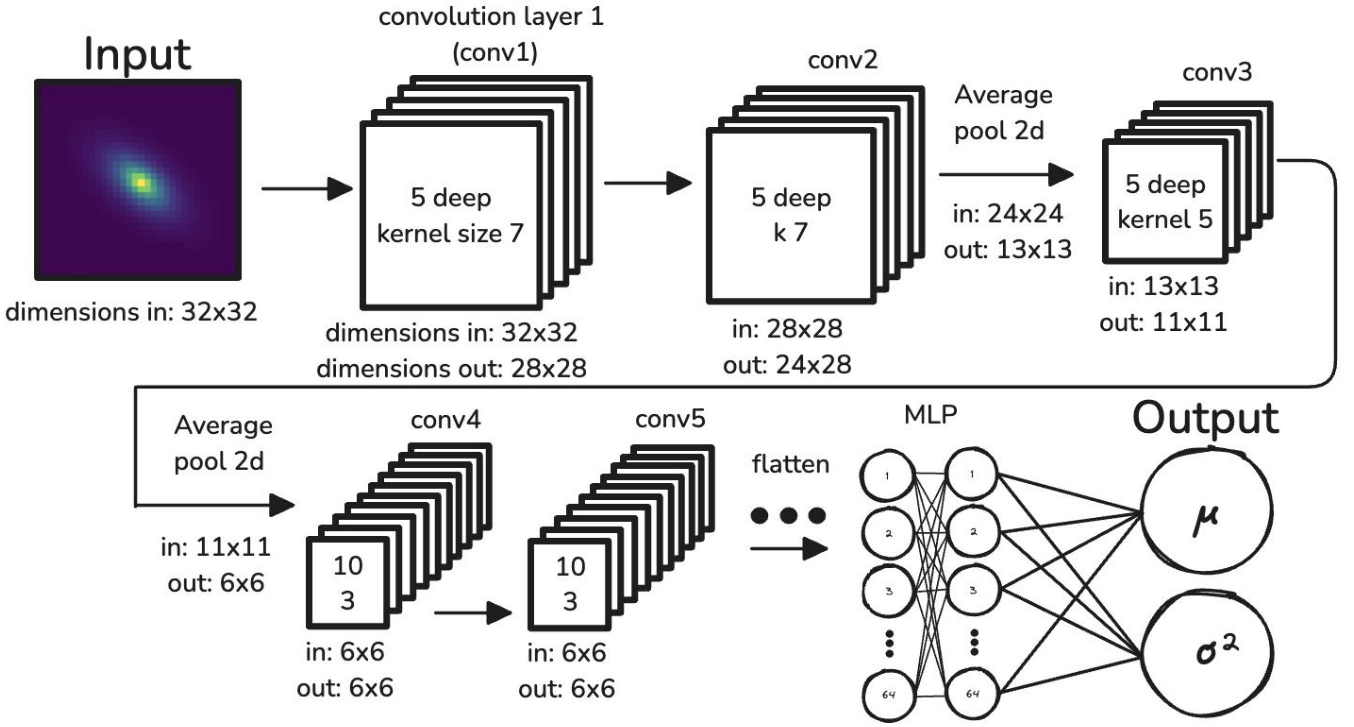
Fully connected layer architecture is simple

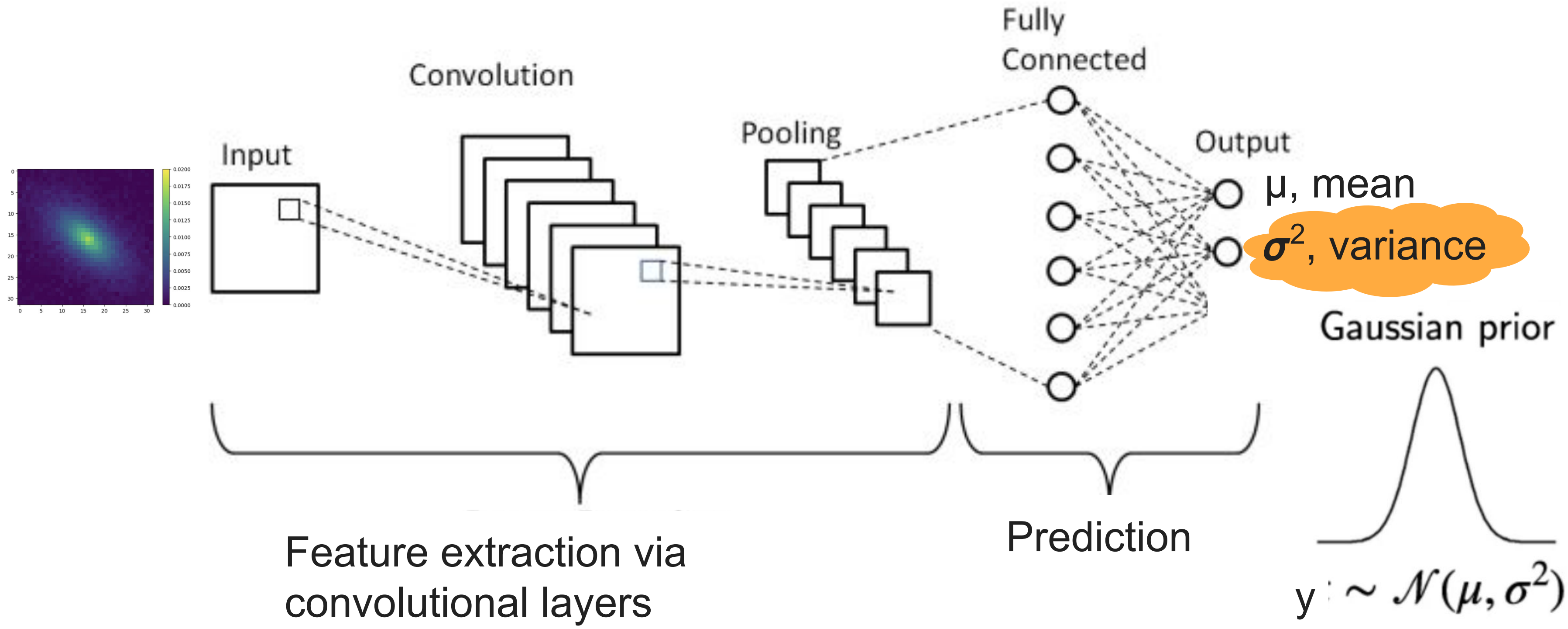






The CNN architecture adds convolutional layers on top of the existing MLP





Questions like how does the uncertainty itself behave, does it reproduce the expected profile?