Deeply Uncertain: Comparing Methods of Uncertainty Quantification in Deep Learning Algorithms

João Caldeira

ICLR Fundamental Science in the era of AI workshop

26 April 2020 [talk recorded on 14 April 2020]

Uncertainty Quantification in Deep Learning

Deep learning is used in many applications in the physical sciences.

In those sciences, we are used to having uncertainty on every measurement or prediction.

In last years, many methods put forth for uncertainty quantification:

- Bayesian Neural Networks
- Concrete Dropout
- Deep Ensembles
- ...

Epistemic and aleatoric, statistical and systematic

Uncertainty in deep learning is often divided into

aleatoric or *irreducible*: uncertainty related to corruption of input data, such as detector noise	*epistemic* or *reducible*: uncertainty stemming from an imperfect model, goes down with more data

Epistemic and aleatoric, statistical and systematic

Uncertainty in deep learning is often divided into

aleatoric or *irreducible*: uncertainty related to corruption of input data, such as detector noise	*epistemic* or *reducible*: uncertainty stemming from an imperfect model, goes down with more data

Uncertainty in physical sciences is often divided into

statistical: can be statistically determined from input data	*systematic*: not statistical

Epistemic and aleatoric, statistical and systematic

Uncertainty in deep learning is often divided into

aleatoric or *irreducible*: uncertainty related to corruption of input data, such as detector noise	*epistemic* or *reducible*: uncertainty stemming from an imperfect model, goes down with more data

Epistemic is always systematic

Statistical is always aleatoric

No statistical epistemic, but systematic aleatoric do exist

Problem summary:

1. Which uncertainty quantification method to choose
2. How to interpret the results

Problem summary:

1. Which uncertainty quantification method to choose
2. How to interpret the results

Our contribution:

Build simple sandbox with pendulum problem

From (L, T, m, θ), predict $g = 4\pi^2 L/T^2$: a problem any physics undergrad is familiar with

Setup

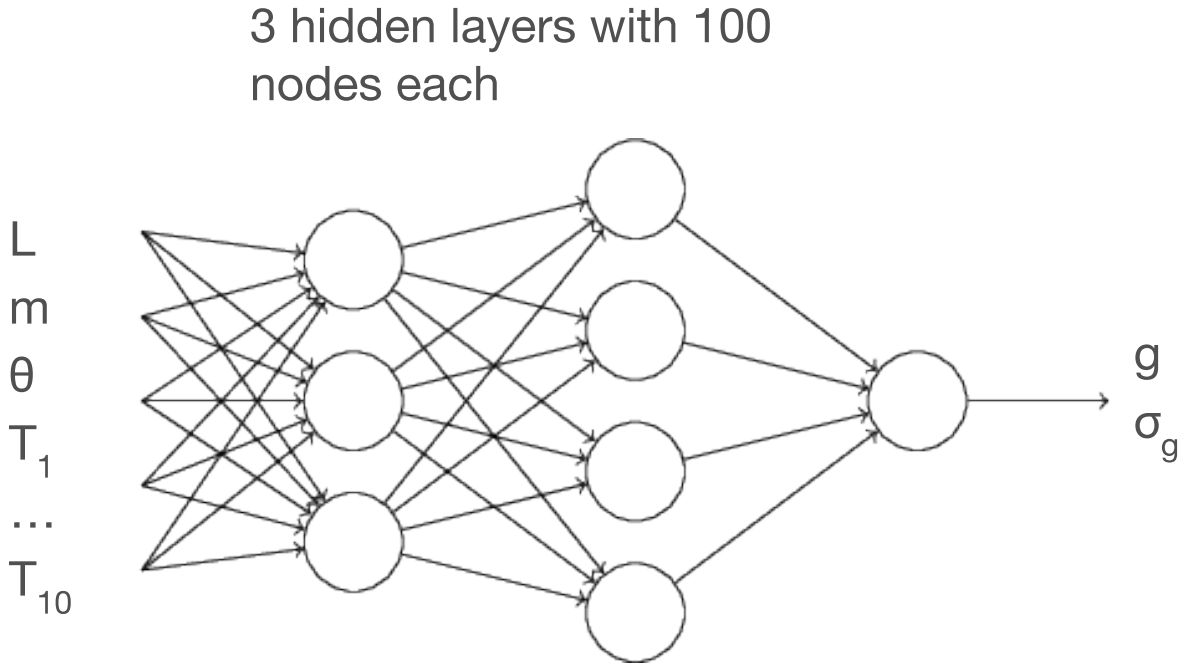


Image credit:
Michael Nielsen

Brief introduction to the UQ methods

For all these methods, optimize (g, σ_g) to maximize Gaussian log likelihood of right answer, so loss is

$$L = \log \sigma_g + \frac{1}{2}\left(\frac{g - g_{true}}{\sigma_g}\right)^2$$

σ_g provides an estimate of the aleatoric uncertainty, while the variance between different models' predictions gives epistemic uncertainty

Deep ensembles: different models

Concrete dropout: dropping different neurons

Bayesian NN: each weight is sampled from distribution

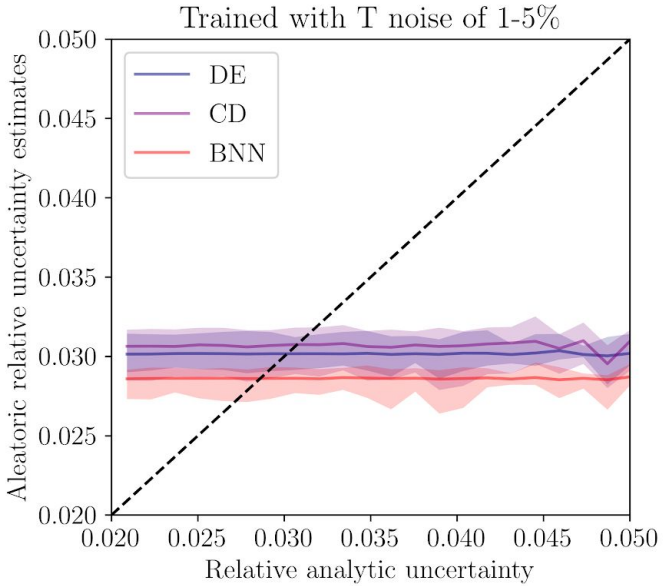
How to introduce noise

Statistical (aleatoric): add noise to the T measurements (sample them from a normal distribution)

Systematic (aleatoric): add noise to the single L measurement

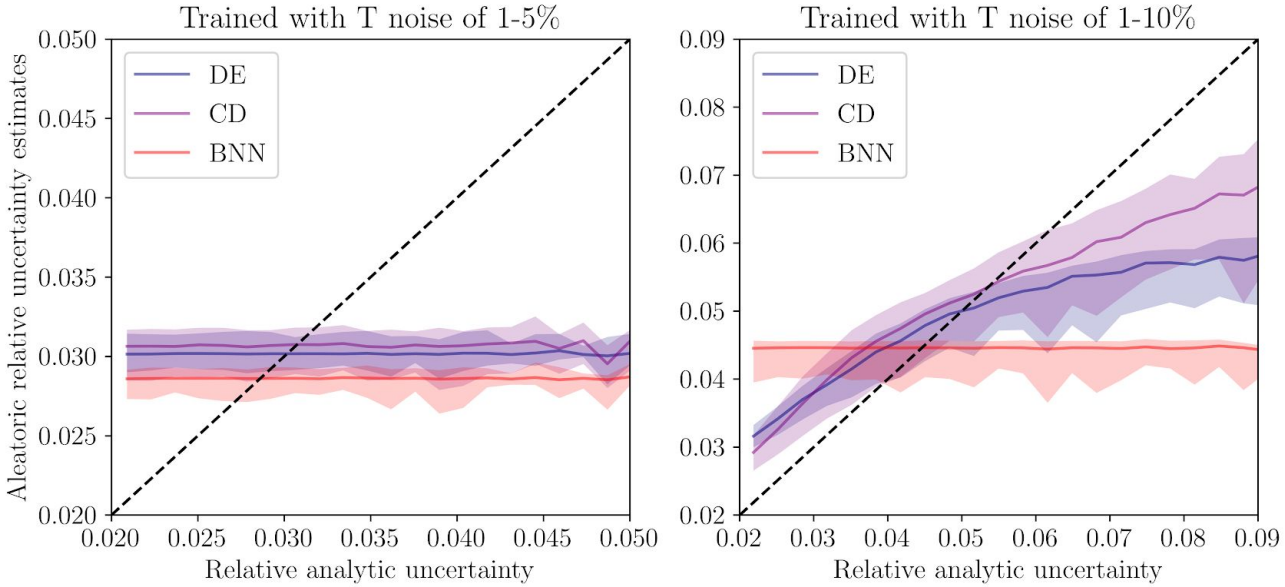
Systematic (epistemic): test in different region from training set

Results: how well is aleatoric uncertainty captured?



Plot shows 16, 50, and 84th percentiles.

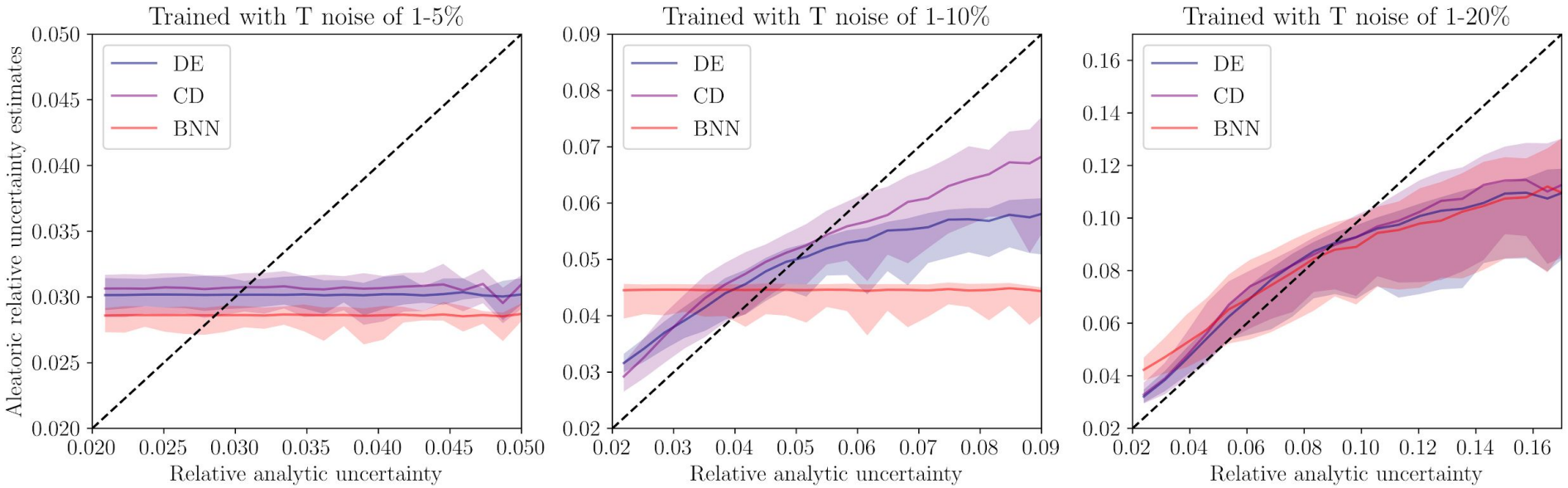
Results: how well is aleatoric uncertainty captured?



Plot shows 16, 50, and 84th percentiles.

Fermilab

Results: how well is aleatoric uncertainty captured?



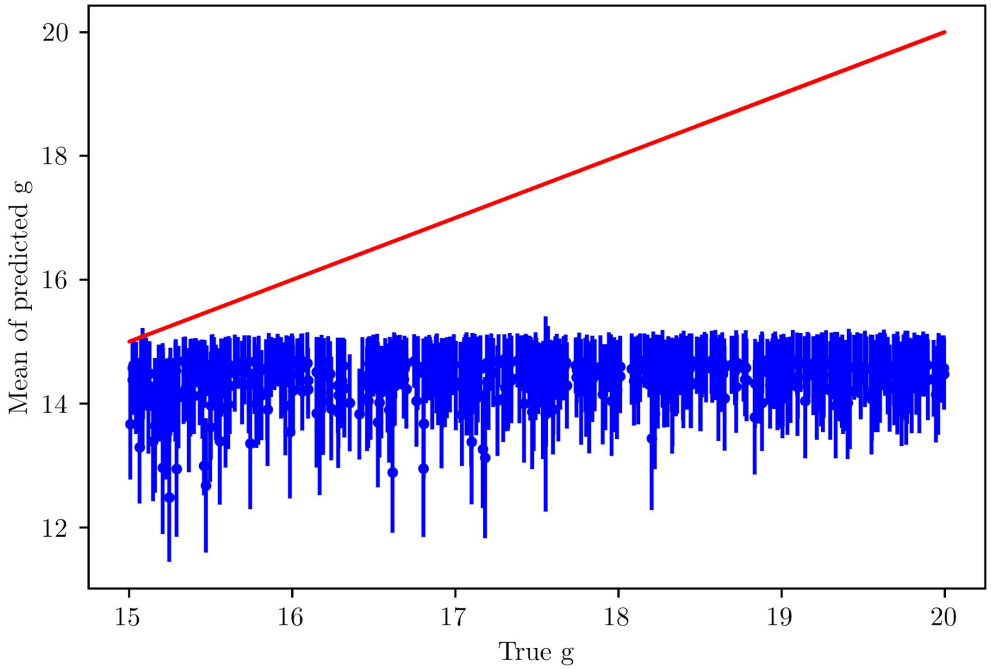
Plot shows 16, 50, and 84th percentiles.

Fermilab

Results: out of distribution uncertainties

Several ways to go out of distribution. We train on g in (5, 15) m/s^2.
Can test on g in (15, 20) m/s^2:

Terrible results for all methods!

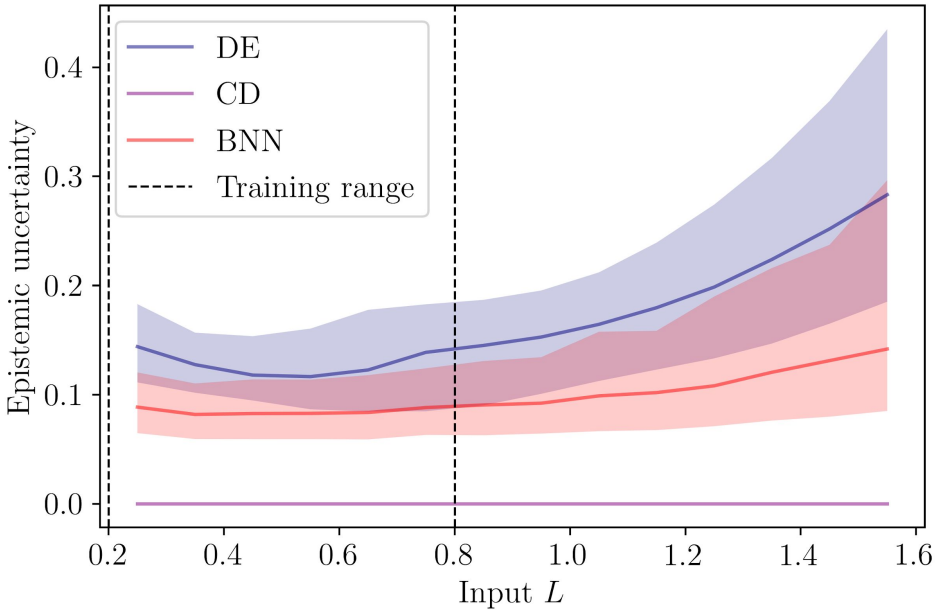


Results: out of distribution uncertainties

An easier test is to let L, T out of training distribution, keeping g in the trained range.

Concrete dropout epistemic uncertainty always very low, as dropout probabilities end up close to zero!

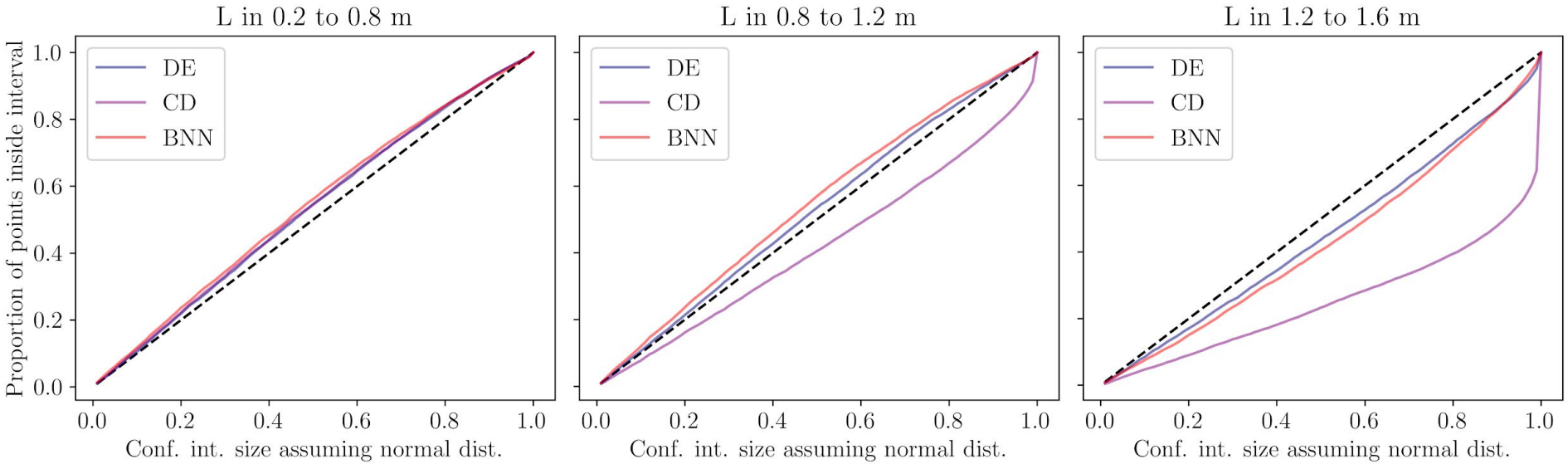
Others starting coming up when we go out-of-distribution.



Plot shows 16, 50, and 84th percentiles.

Results: calibration

Do the uncertainties accurately reflect the error?



Fermilab

Conclusions

- Aleatoric uncertainties are reasonably well-modeled *but* need to make sure to include a large range of uncertainties in the training set. Just like for predictions, but less visible! BNN needs most attention.

- Out of distribution is a hard problem still. CD in particular totally failed us, and all fail when output goes away from the training distribution.

- DE is technically simplest and seems to work well.

see paper on arXiv for more details!