Uncertainty Quantification in Deep Learning

Danijela Horak, Head of AI Research BBC

Aug, 2025.
Serbia

AGENDA:

Part I
Theory: motivation and getting the basic understanding of the problem
Practice: training an MLP classifier for MNIST and error analysis
Part II
Theory: Formal definition of uncertainty, Callibration , Deep Ensambles, MC dropout,
Practice: deep ensambles implementation, error analysis, MC dropout
Part II
Theory: different types of uncertainty, idealized uncertainty, OODs,
Practice: error analysis on all three,

The back story: detection of fully AI generated images

Partial Manipulation

Models fit for use in the news room:

1. Accuracy is paramount
2. One can never achieve perfect accuracy
3. How to deal with ambiguity
4. How to deal with the cases of models being confidently wrong
5. How to explain the concept 70% prob/conf this image is manipulated and what signal does that provide for the users?

We need models that know when they do not know.

Models fit for use in the news room:

1. Accuracy is paramount
2. One can never achieve perfect accuracy
3. How to deal with ambiguity
4. How to deal with the cases of models being confidently wrong
5. How to explain the concept 70% prob/conf this image is manipulated and what signal does that provide for the users?

We need models that know when they do not know.

Option 1 – explainability
Option 2 – uncertainty quantification

OPTION 1: **EXPLAINABILITY**

	Ground Truth: True	Ground Truth: False
Predicted: True		
Predicted: False		

OPTION 1: **EXPLAINABILITY**

	Ground Truth: True	Ground Truth: False
Predicted: True		
Predicted: False		

OPTION 1: **EXPLAINABILITY**

Explanations must be:

1. Humanly understandable
2. Faithful (accurately represents the internal logic of the model)

Problems

1. features are non-semantic for many learning tasks and most of the time
2. Features are polysemantic (A single MLP neuron lights up for several distinct patterns.)
3. But even if the explanation agrees with your intuition, the model may be wrong and you may be wrong

OPTION 1: **EXPLAINABILITY – ARE HUMAN UNDERSTANDABLE EXPLANATIONS ACHIEVABLE WITH DNNS**

Questioning Representational Optimism in Deep Learning: The Fractured Entangled Representation Hypothesis

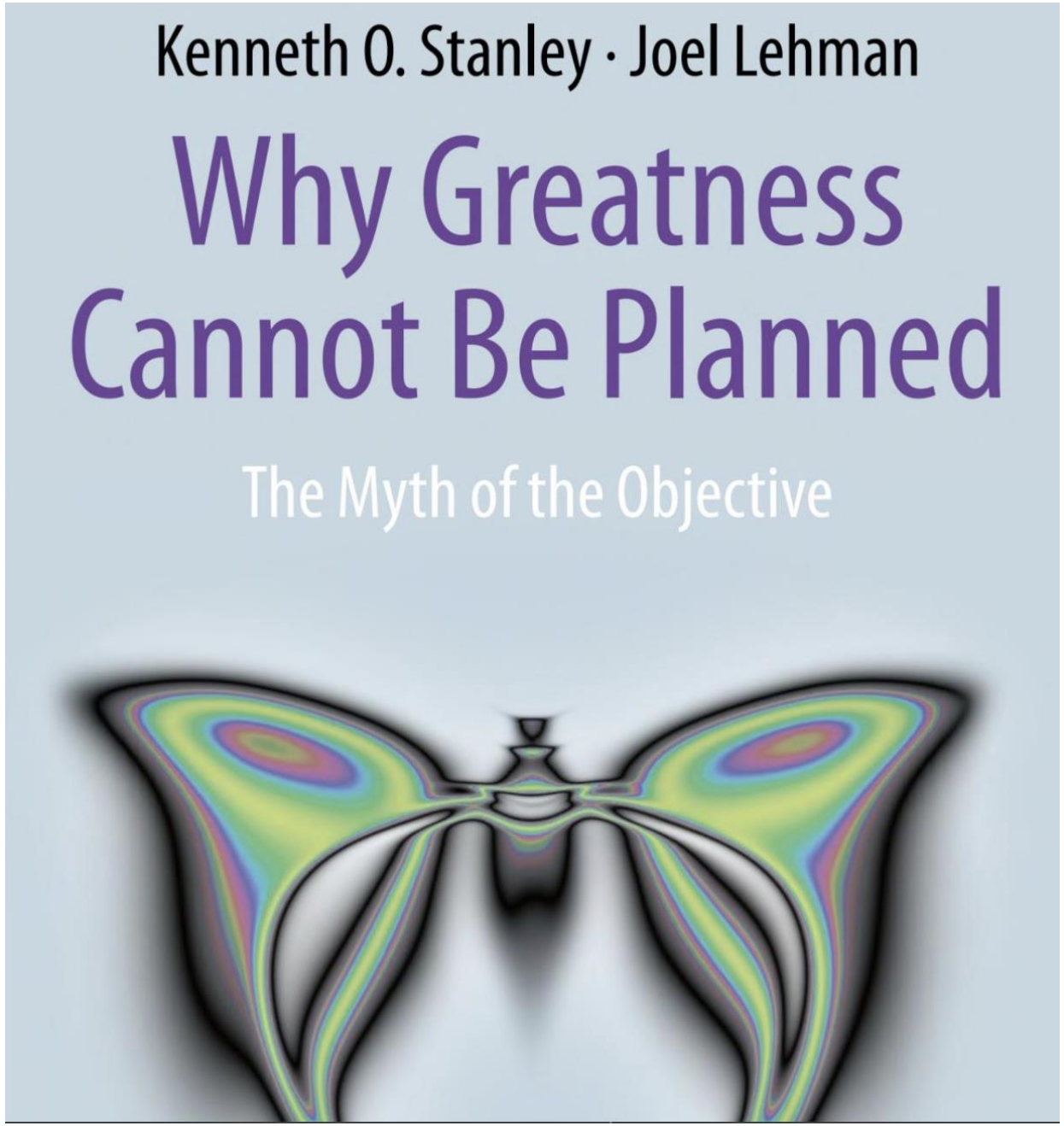
Akarsh Kumar[1], Jeff Clune[2,3], Joel Lehman[4], Kenneth O. Stanley[5]

[1]MIT, [2]University of British Columbia, [3]Vector Institute, [4]University of Oxford, [5]Lila Sciences

NNs trained via SGD vs NNs trained through an open ended search

Fractured entangled representation vs unified factored representation

https://arxiv.org/pdf/2505.11581


Kenneth O. Stanley · Joel Lehman
Why Greatness Cannot Be Planned
The Myth of the Objective

OPTION 1: **EXPLAINABILITY – ARE HUMAN UNDERSTANDABLE EXPLANATIONS ACHIEVABLE WITH DNNS**

Questioning Representational Optimism in Deep Learning: The Fractured Entangled Representation Hypothesis

Akarsh Kumar[1], Jeff Clune[2,3], Joel Lehman[4], Kenneth O. Stanley[5]

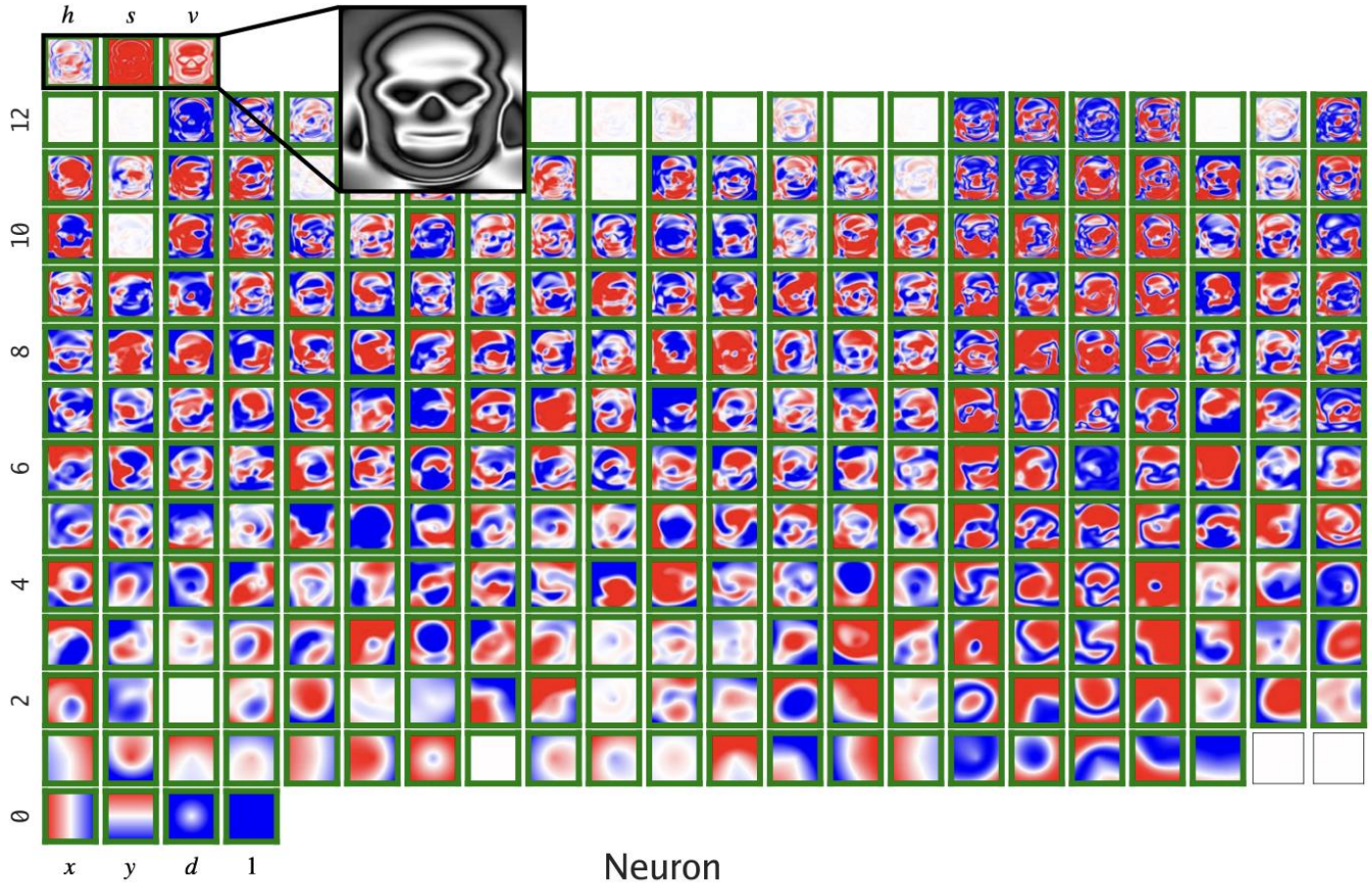
[1]MIT, [2]University of British Columbia, [3]Vector Institute, [4]University of Oxford, [5]Lila Sciences

NNs trained via SGD vs NNs trained through an open ended search

Fractured entangled representation vs unified factored representation

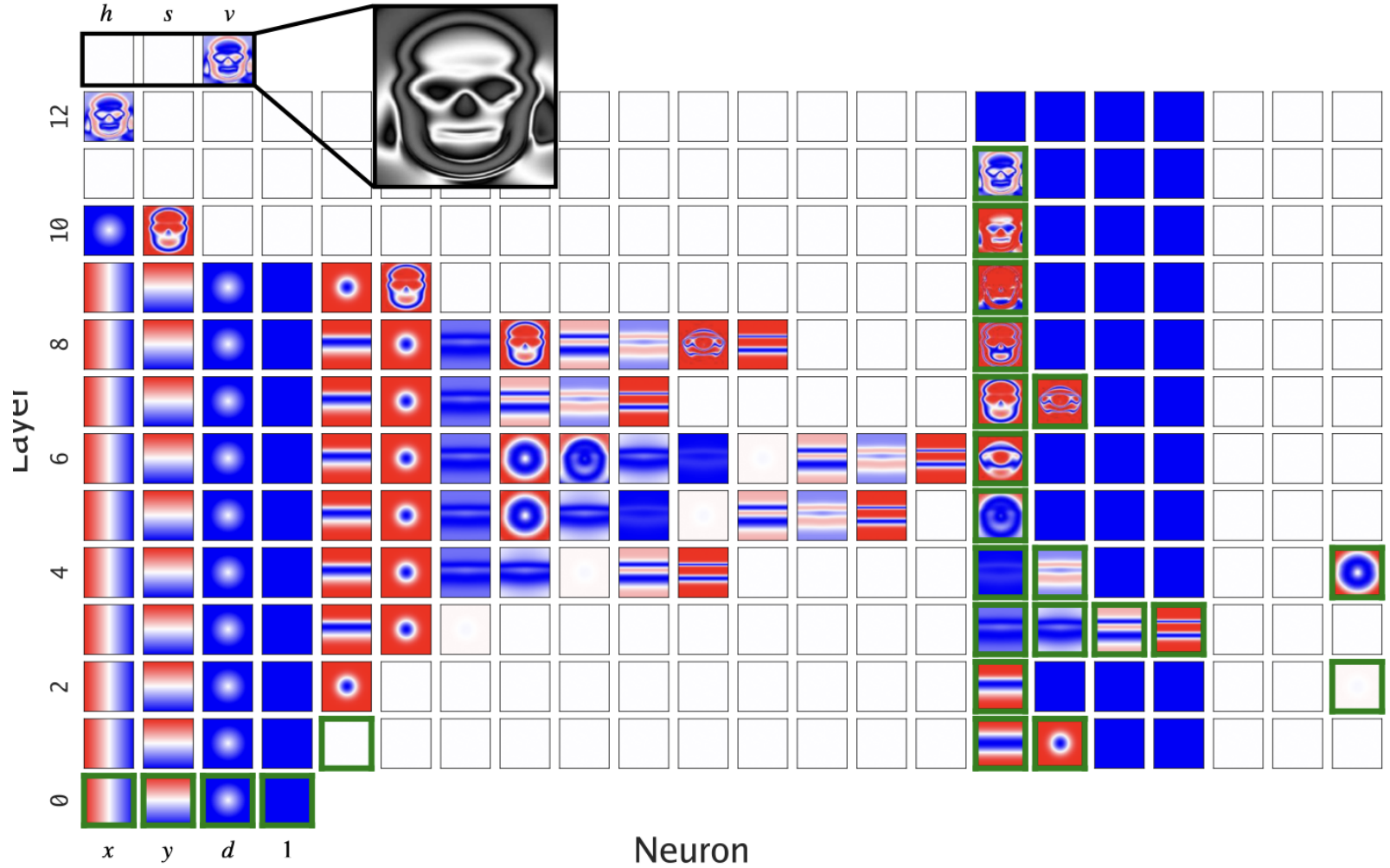
https://arxiv.org/pdf/2505.11581

OPTION 1: **EXPLAINABILITY – HUMAN UNDERSTANDABLE EXPLANATIONS**



(b) Conventional SGD CPPN

Each image visualizes the latent representation of all inputs at a specific layer and neuron (red and blue represent low and high activation, respectively).



(a) Picbreeder CPPN

ARITHMETIC WITHOUT ALGORITHMS: LANGUAGE MODELS SOLVE MATH WITH A BAG OF HEURISTICS

Yaniv Nikankin[1]* **Anja Reusch**[1] **Aaron Mueller**[1,2] **Yonatan Belinkov**[1]
[1]Technion – Israel Institute of Technology [2]Northeastern University

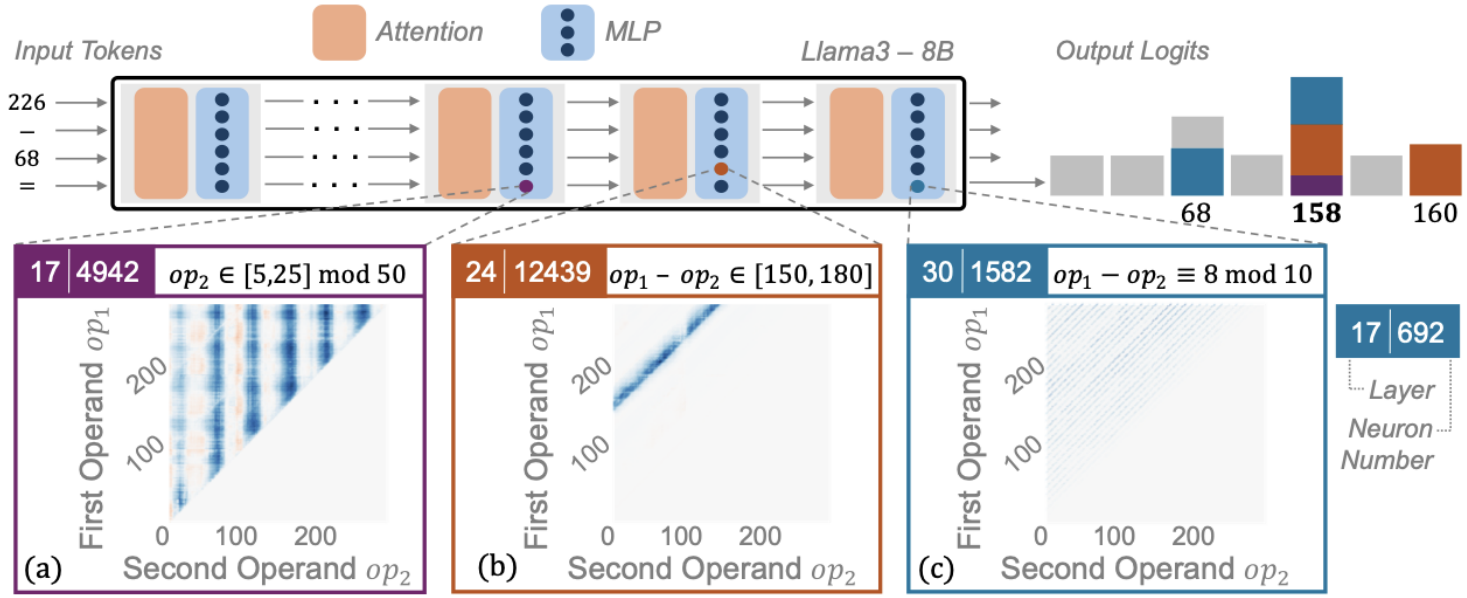


Figure 1: **Bag of heuristics visualization.** We show that transformer LLMs solve arithmetic prompts by combining several unrelated heuristics, each activating according to rules based on the input values of operands, and boosting the logits of corresponding result tokens. These heuristics are manifested in single MLP neurons in mid to late layers.

https://arxiv.org/html/2410.21272v1

OPTION 1: **EXPLAINABILITY**

In the light of the ML models not being perfectly accurate, are explanations serving only to confirm our existing biases?

Solution: add uncertainty quantification and only explain predictions in which the model has low uncertainty

TUTORIAL PART ONE

Notebook 1:

1. Train an MLP classifier
2. Plot loss
3. Calculate accuracy, F1, etc.
4. Error analysis (plot misclassified images)
5. Error analysis plot embeddings and
6. Load OOD data sets, visualize
7. Predict classes, visualize in the embedding space

THEORY PART TWO: **UNCERTAINTY QUANTIFICATION**

Digression Three

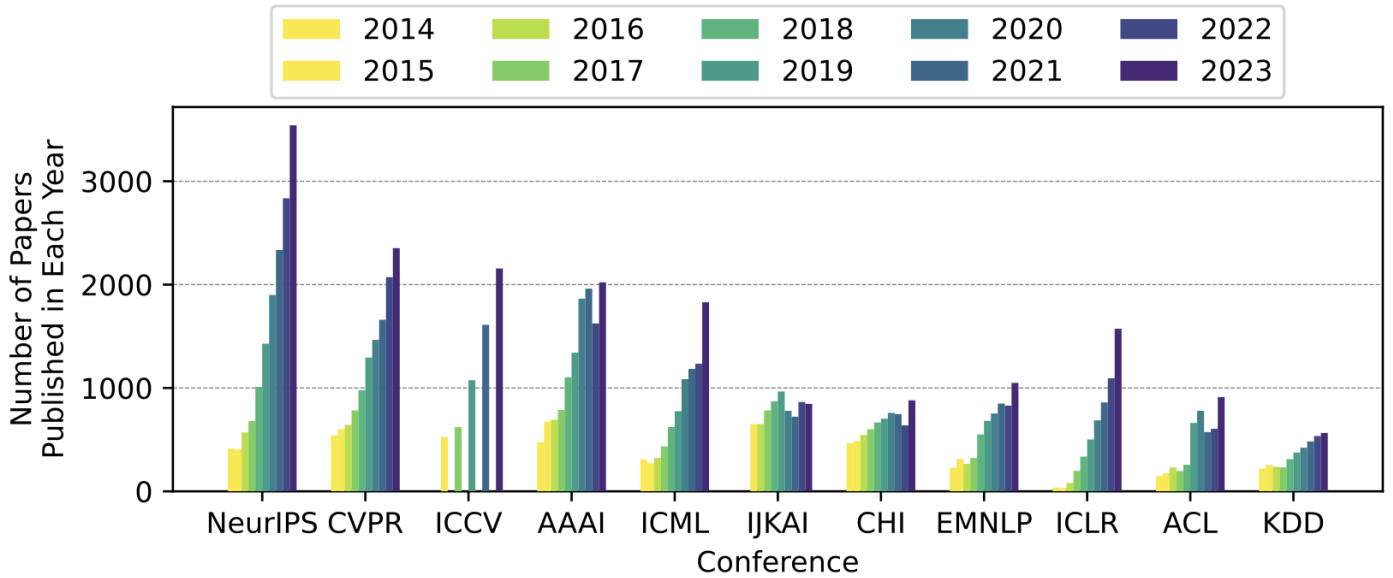


Figure 1. The number of papers published each year in top AI and ML conferences.

Azad et al 24

The amount of knowledge in AI doubles every 24 months

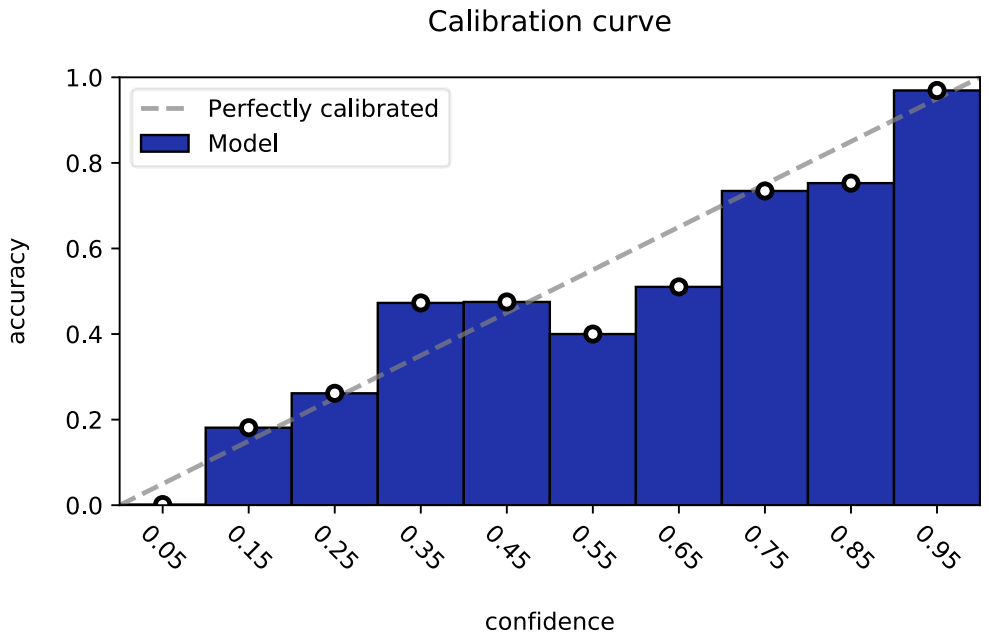
4% 96%

x, 2x, 4x, 8x, 16x, 32x, 64x, 128x,

2014 2024, 2026, 2028

THEORY PART TWO: **UNCERTAINTY QUANTIFICATION**

Model Calibration



Guo et al. propose “binning” as a calibration method .
Bm – containing all samples that have confidence that fall Into the interval Im = (m-1/M, m/M]

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i),$$

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i,$$

Goal is to have ECE ~ 0

https://arxiv.org/abs/1706.04599

Model Calibration: Expected Calibration Error

$$\mathop{\mathbb{E}}_{\hat{P}}\left[\left|\mathbb{P}\left(\hat{Y}=Y \mid \hat{P}=p\right)-p\right|\right]$$

$$\mathrm{ECE}=\sum_{m=1}^{M}\frac{|B_m|}{n}\left|\mathrm{acc}(B_m)-\mathrm{conf}(B_m)\right|$$

Goal is to have ECE ~ 0

https://arxiv.org/pdf/1706.04599

Model Calibration: Temperature Scaling

WHAT IS TEMPERATURE SCALING?

For classification problems, the neural network output a vector known as the **logits**. The logits vector is passed through a softmax function to get class probabilities. Temperature scaling simply divides the logits vector by a learned scalar parameter, i.e.

$$P(\hat{\mathbf{y}}) = \frac{e^{\mathbf{z}/T}}{\sum_j e^{z_j/T}}$$

where $\hat{y}$ is the prediction, where $\mathbf{z}$ is the logit, and T is the learned parameter. We learn this parameter on a validation set, where T is chosen to minimize negative log likelihood. Intuitively, temperature scaling simply softens the neural network outputs. This makes the network slightly less confident, which makes the confidence scores reflect true probabilities.

Model Calibration: Temperature Scaling

Learn T on held out data set post-hoc

NOTEBOOK TWO: **Calibration**

1. Write post hoc calibration training
2. Visualise uncertainties

Model Calibration: Problems

Calibration Methods	Strengthens	Limitations
Post-hoc Calibration	(1) Simple and effective: this line of work can be easily implemented and integrated into existing models. On the other hand, it shows its effectiveness and accuracy-preserving (Zhang et al., 2020) with highly competitive predictive and calibration performance.	(1) Suboptimal performance: Post-hoc calibration techniques may not fully optimize the calibration of the model, potentially leading to suboptimal calibration performance. Additionally, Chidambaram et al., (Chidambaram & Ge, 2024) recently suggested that the performance of temperature scaling degrades with the amount of overlap between classes.
Regularization	(1) Training-time calibration, as calibration is performed during the training, we can obtain a well calibrated model directly after training, omitting the post-hoc steps. (2) Alleviating overfitting: regularization methods improves calibration through alleviated overfitting such as L2 regularization and Focal loss (Lin et al., 2017). Thus this line of methods can also improve generalization performance.	(1) Implementation and Tuning Complexity. Implementing regularization method is less straightforward as compared to post-hoc ones. Also this line of work usually introduce hyper-parameters, for example, smoothing factor in label smoothing (Müller et al., 2019) and (Mukhoti et al., 2020) empirically showed that the importance of γ in calibrating deep models. However, selecting the appropriate hyper-parameters can be challenging and may require extensive hyper-parameter tuning.
Uncertainty Estimation (UE)	(1) Uncertainty quantification. Bayesian-method based calibration can quantify the uncertainty in the calibrated model (Muehleisen & Bergerson, 2016). (2) Incorporating prior information. UE methods allow incorporation of prior knowledge or beliefs about the parameters being estimated and provide more robust estimates, especially when dealing with limited data (Kennedy & O'Hagan, 2001).	(1) Computational complexity. Many uncertainty estimation methods involve computationally intensive procedures, which can be impractical for large datasets or real-time applications, limiting their scalability. For example MC-dropout (Gal & Ghahramani, 2016) requires multiple inference runs and perform averaging on predictions.
Hybrid Methods	(1) Combines strengths: Hybrid methods can leverage the advantages of multiple approaches, potentially leading to better overall performance by addressing the limitations of individual techniques. Section 4.4 describes some of these methods	(1) Increased complexity: Hybrid methods usually introduce additional complexity in model calibration, Integrating multiple techniques into a cohesive framework may pose implementation challenges and increase the risk of errors requiring more extensive validation and testing to ensure robustness and effectiveness.

Survey: https://arxiv.org/pdf/2308.01222

THEORY PART THREE: **UNCERTAINTY DEFINTION**

UNCERTAINTY DEFINTION

Assume $f : \mathcal{X} \to \mathcal{Y}$

Total uncertainty of r.v. $Y \in \mathcal{Y}$ is $H(Y)$

Sources of uncertainty:

- Data (Aleatoric Uncertainty)
- Model knowledge (Epistemic Uncertainty)

BAYESIAN INFERENCE FOR DNNs: RECAP

Given an ML problem, where :
y denotes class label,
x input features (image),
w model parameters of NN
\mathcal{D} data,
we typically want to find the posterior predictive distribution:

$$p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw,$$

BAYESIAN INFERENCE FOR DNNs: RECAP

Given an ML problem, where :
y denotes class label,
x input features (image),
w model parameters of NN
\mathcal{D} data,
we typically want to find the posterior predictive distribution:

$$p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw,$$

Model uncertainty

Data uncertainty

The Bayesian framework offers a practical tool to reason about uncertainty in deep learning [61]. In Bayesian modeling, the model uncertainty is formalized as a probability distribution over the model parameters θ, while the data uncertainty is formalized as a probability distribution over the model outputs y^*, given a parameterized model f_θ. The distribution over a prediction y^*, the predictive distribution, is then given by

$$p(y^*|x^*, D) = \int \underbrace{p(y^*|x^*, \theta)}_{\text{Data}} \underbrace{p(\theta|D)}_{\text{Model}} d\theta \ . \tag{11}$$

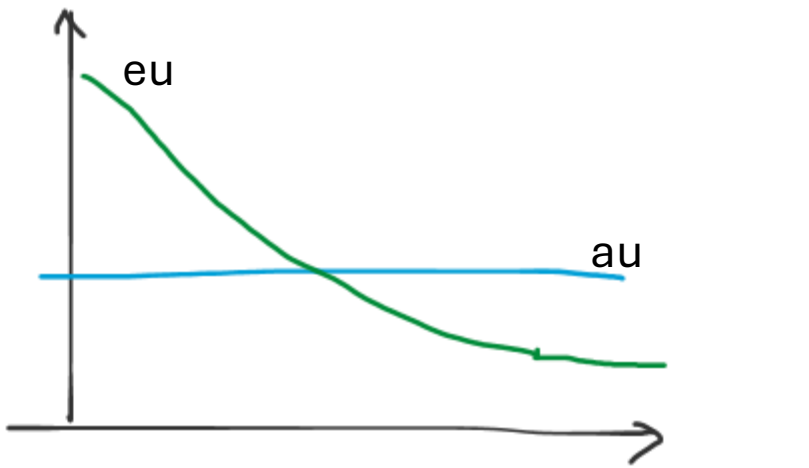
2) Distributional Uncertainty: Depending on the approaches that are used to quantify the uncertainty in y^*, the formulation of the predictive distribution might be further separated into data, distributional, and model parts [32]:

$$p(y^*|x^*, D) = \int \int \underbrace{p(y|\mu)}_{\text{Data}} \underbrace{p(\mu|x^*, \theta)}_{\text{Distributional}} \underbrace{p(\theta|D)}_{\text{Model}} d\mu d\theta \ . \tag{13}$$

Let’s try to understand the properties of these uncertainties

$$\underbrace{\text{Total uncertainty}}_{\text{TU}} = \underbrace{\text{aleatoric uncertainty}}_{\text{AU}} + \underbrace{\text{epistemic uncertainty}}_{\text{EU}}$$

- **TU**, **EU** maximal for total ignorance
- **EU** $\to 0$ for $n \to \infty$
- **AU** $\equiv c \in \mathbb{R}$



UNCERTAINTY DEFINTION

Total uncertainty = aleatoric + epistemic

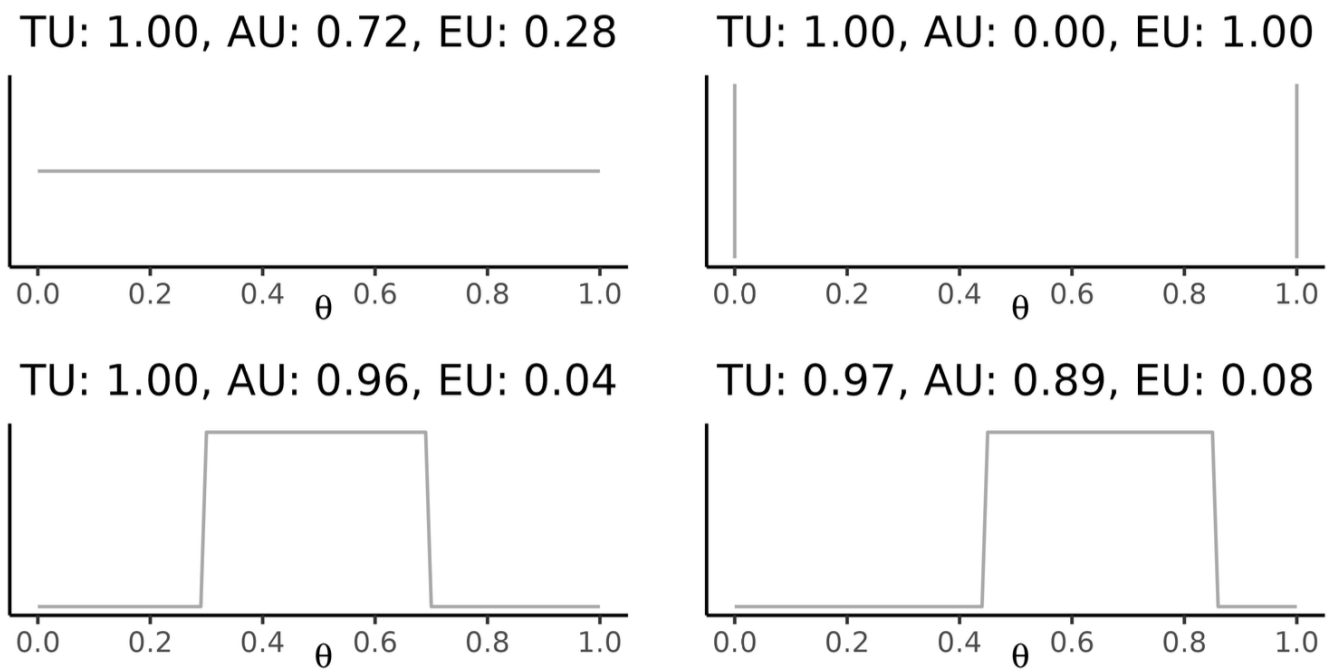
Some people also talk about “distributional uncertainty”

Total uncertainty: $H(Y) = H(\mathbb{E}_Q[Y \mid \boldsymbol{\theta}]) = -\sum_y p(y) \cdot \log p(y)$

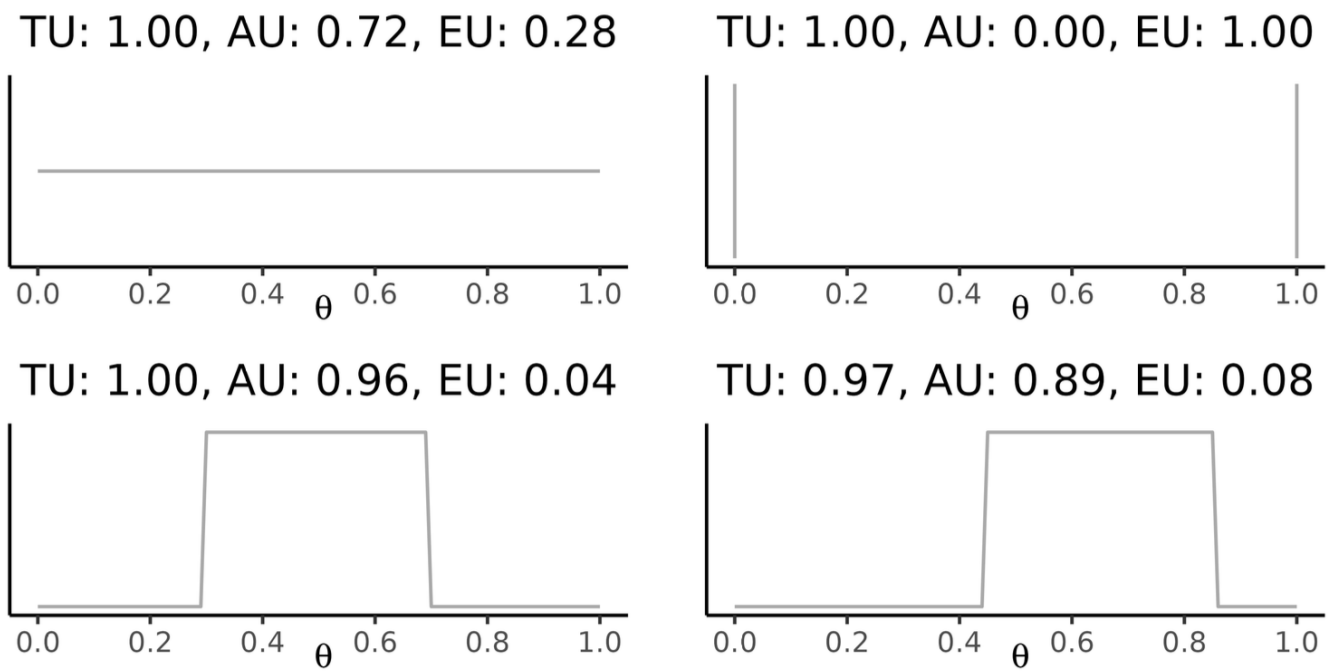
Aleatoric Uncertainty (aka conditional entropy): $H(Y \mid \Theta) = \mathbb{E}_Q[H(Y \mid \boldsymbol{\theta})] = \mathbb{E}_Q\left[-\sum_y p(y|\boldsymbol{\theta}) \log p(y|\boldsymbol{\theta})\right]$

Epistemic Uncertainty (mutual information): $I(Y, \Theta) = H(Y) - H(Y \mid \Theta)$

Example that challenges our intuition about epistemic uncertainty being a “model/knowledge” uncertainty



Example that challenges our intuition about epistemic uncertainty being a “model/knowledge” uncertainty



EU measures the “conflict” in the hypothesis, rather than the measure of “ignorance”

THEORY PART THREE: **BAYSIAN ENSEMBLE**

BAYESIAN INFERENCE FOR DNNs: RECAP

Given an ML problem, where :
y denotes class label,
x input features (image),
w model parameters of NN
\mathcal{D} data,
we typically want to find the posterior predictive distribution:

$$p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw,$$

Bayesian model average: rather than use one model with a single set of parameters, we average many models weighted by their posterior probabilities.

BAYESIAN INFERENCE FOR DNNs: RECAP

Typically approx. the integral $p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw,$ via simple MC:

$$p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw \approx \frac{1}{J}\sum_j p(y|x, w_j)\,, \quad w_j \sim p(w|\mathcal{D}).$$

Using :
stochastic approach: MCMC
Deterministic approaches (that try to approximate the posterior and then sample from it):
Laplace Approximation, Variational Inference

BAYESIAN INFERENCE FOR DNNs: ensemble method

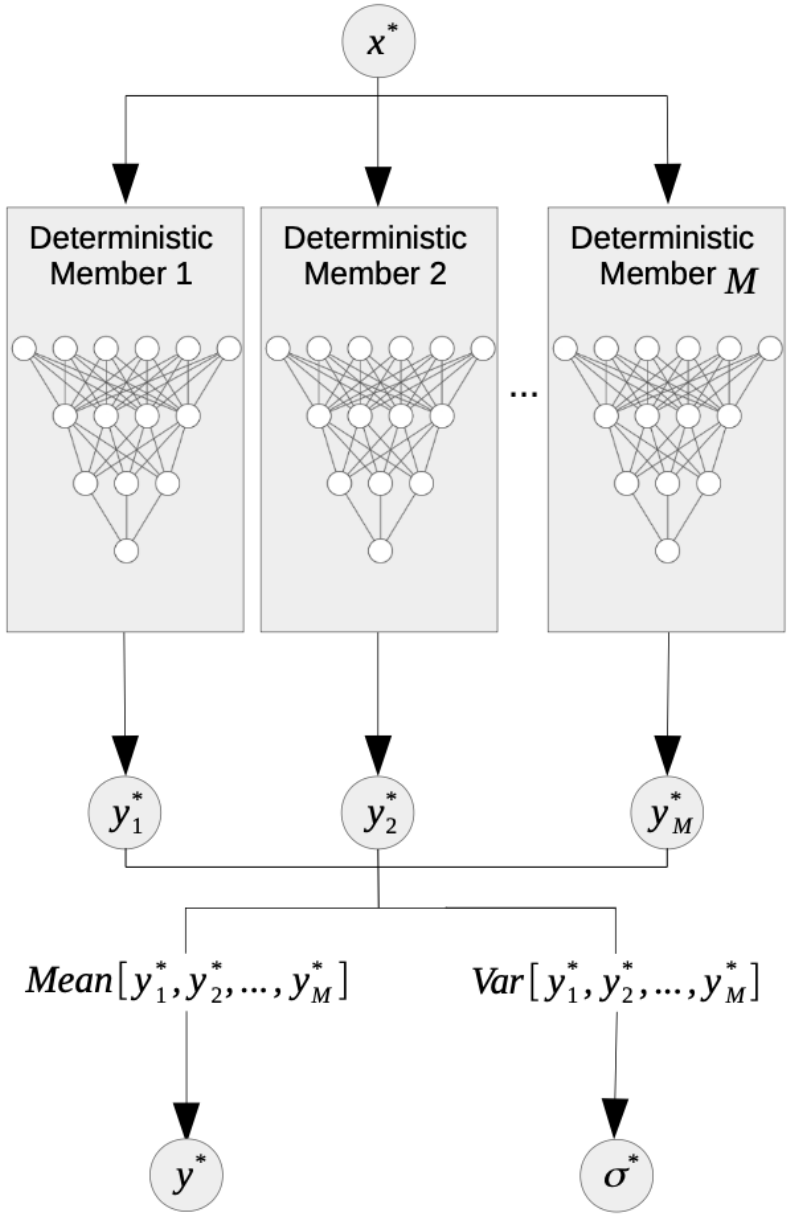
DL - integral we are trying to compute is over a multi-million dimensional parameter space
posterior is highly non-Gaussian and multi-modal.
We typically sample only limited number of points in the parameter space.

Ideally we want to select for two key properties when sampling:

- we would want to find ***typical*** points in the posterior, representing regions where there is a lot of mass (integral);
- we would also want a ***diversity*** of points, such that the different parameters we select give rise to different functions.

$$p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw,$$

BAYESIAN Ensemble Method



Diversity: we re-train our neural network multiple times with different initializations and typically find different low loss solutions in different basins of attractions.

High Mass: with SGD optimisation we make sure that these are “typical points”, centred in large basins of attraction

$$p(y|x, D) = 1/J \sum p(y|x, w_j)$$

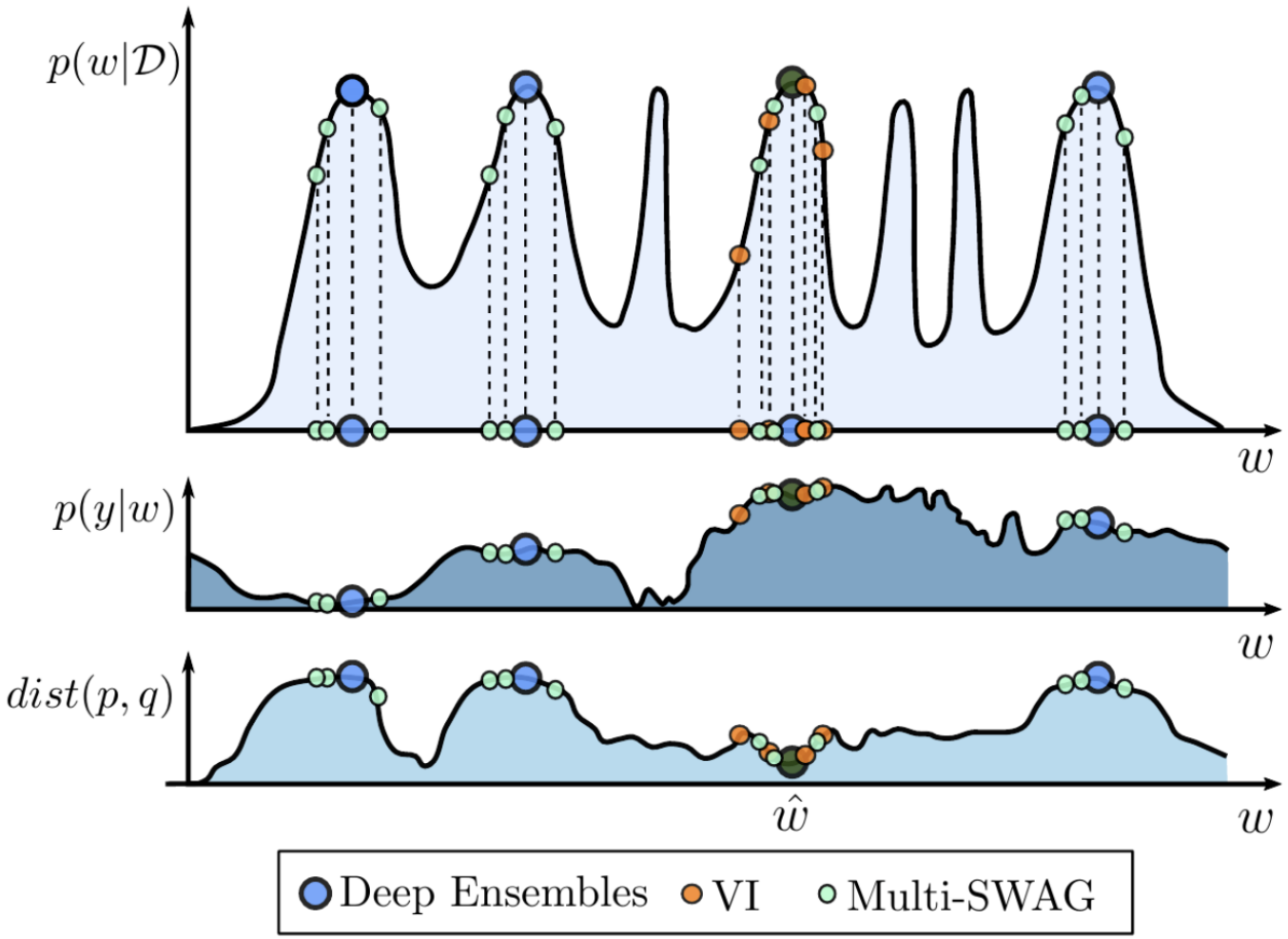


Figure 1. $p(y|x, D) = \int p(y|x, w)p(w|D)dw$. **Top**: $p(w|D)$, with representations from VI (orange) deep ensembles (blue), MultiSWAG (red). **Middle**: $p(y|x, w)$ as a function of w for a test input x. This function does not vary much within modes, but changes significantly between modes. **Bottom**: Distance between the true predictive distribution and the approximation, as a function of representing a posterior at an additional point w, assuming we have sampled the mode in dark green. There is more to be gained by exploring new basins, than continuing to explore the same basin.

https://arxiv.org/pdf/2002.08791

Andrew Gordon Wilson, Pavel Izmailov

Total, Aleatoric and Epistemic uncertainty

M models, each predicts p_{m} \in R^{C}, C=10

- $p_m(x)$ = softmax output of the m-th model
- $\bar{p}(x) = \frac{1}{M} \sum_{m=1}^{M} p_m(x)$ (mean prediction)

$$\text{Total Uncertainty}(x) = H\left[\bar{p}(x)\right] = -\sum_{c=1}^{\bar{}} \bar{p}_c(x) \log \bar{p}_c(x)$$

$$\text{Aleatoric Uncertainty}(x) = \frac{1}{M} \sum_{m=1}^{M} H\left[p_m(x)\right] = -\frac{1}{M} \sum_{m=1}^{M} \sum_{c=1}^{C} p_{m,c}(x) \log p_{m,c}(x)$$

Average/expected entropy per One model (mean of individual model uncertainties) - h

$$\text{Epistemic Uncertainty}(x) = H[\bar{p}(x)] - \frac{1}{M} \sum_{m=1}^{M} H[p_m(x)]$$

NOTEBOOK THREE: **Ensembles**

1. Load MLPs
2. Accuracy of ensemble
3. Plot misclassified with A,E,T - U
4. Plot OOD
5. In theory you can visualize embeddings, but prob not worth doing now

THEORY PART THREE.FIVE : **MC DROPOUT**

What is MC dropout?

- At test time you keep dropout on, sample different dropout masks, and average predictions.
- This corresponds to sampling $w^{(s)} \sim q(w)$, a **variational posterior approximation** (roughly a factorized Bernoulli–Gaussian around the trained weights).
- So the sampling lives in **an approximate posterior** $q(w)$, typically **concentrated near one mode**, and it often **underestimates uncertainty**.

Question – how is MC dropout sampling?

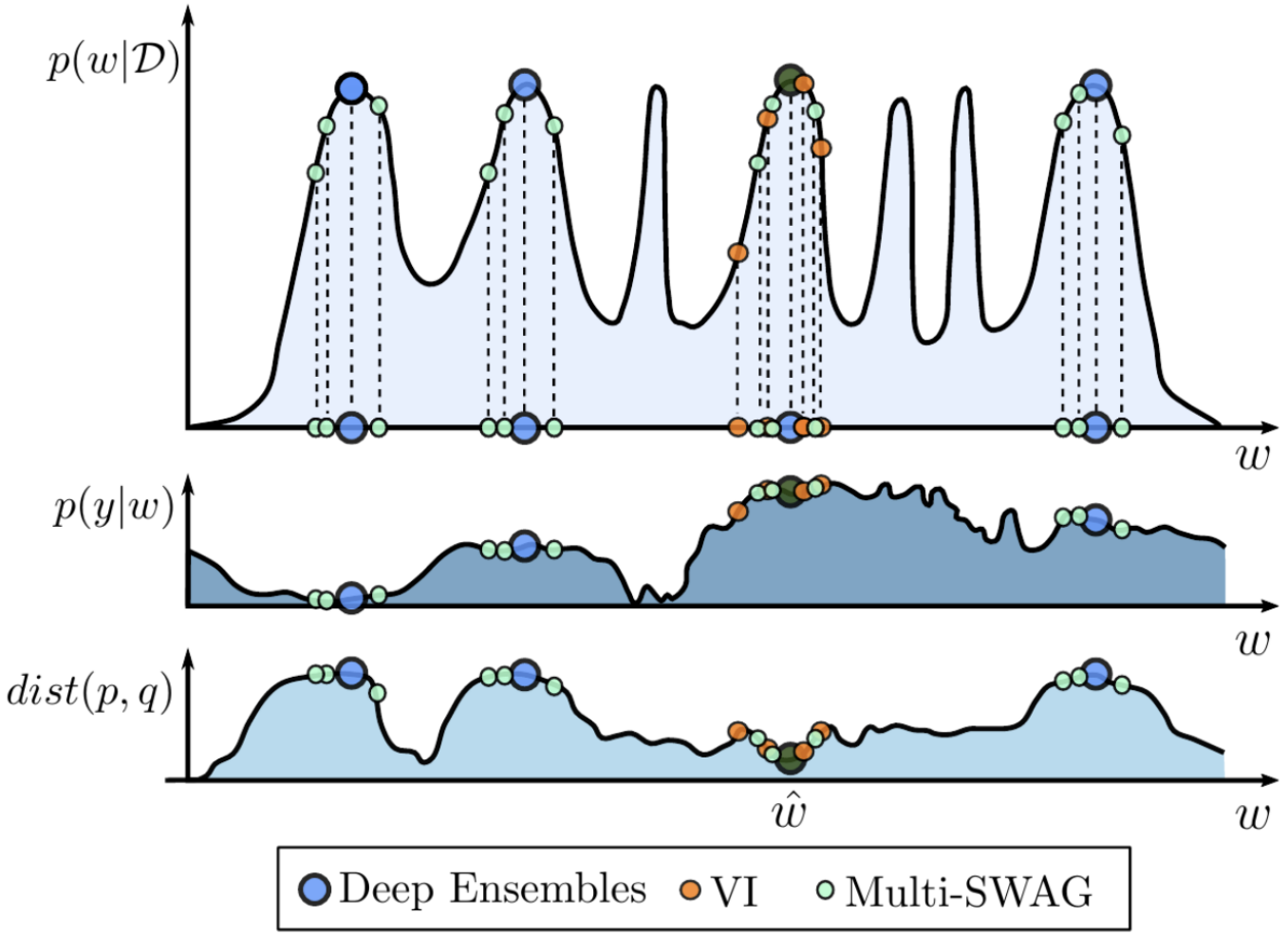


Figure 1. $p(y|x, D) = \int p(y|x, w)p(w|D)dw$. **Top**: $p(w|D)$, with representations from VI (orange) deep ensembles (blue), MultiSWAG (red). **Middle**: $p(y|x, w)$ as a function of w for a test input x. This function does not vary much within modes, but changes significantly between modes. **Bottom**: Distance between the true predictive distribution and the approximation, as a function of representing a posterior at an additional point w, assuming we have sampled the mode in dark green. There is more to be gained by exploring new basins, than continuing to explore the same basin.

https://arxiv.org/pdf/2002.08791

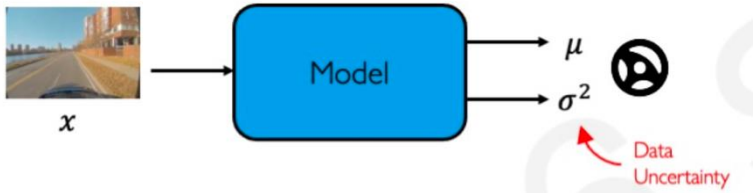
Andrew Gordon Wilson, Pavel Izmailov

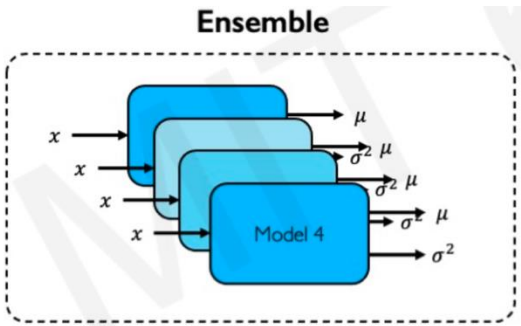
What are its advantages?

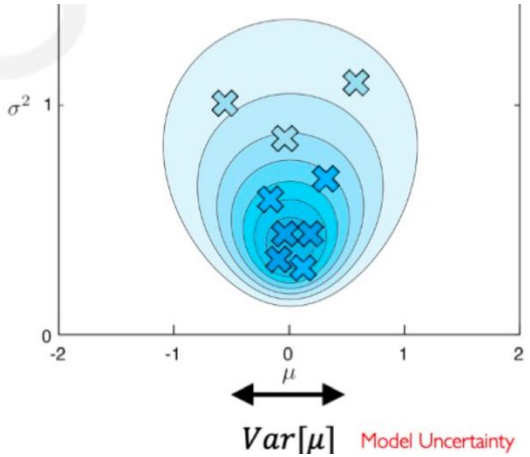
THEORY PART FOUR: EVIDENTIAL DL AND HANDLING OOD

EVIDENTIAL DEEP LEARNING

Instead of ensambling, directly calculate distributional mean and variance







Avi Amini, MIT lecture

EVIDENTIAL DEEP LEARNING for Classification

Sampling from an evidential distribution yields individual new distributions over the data

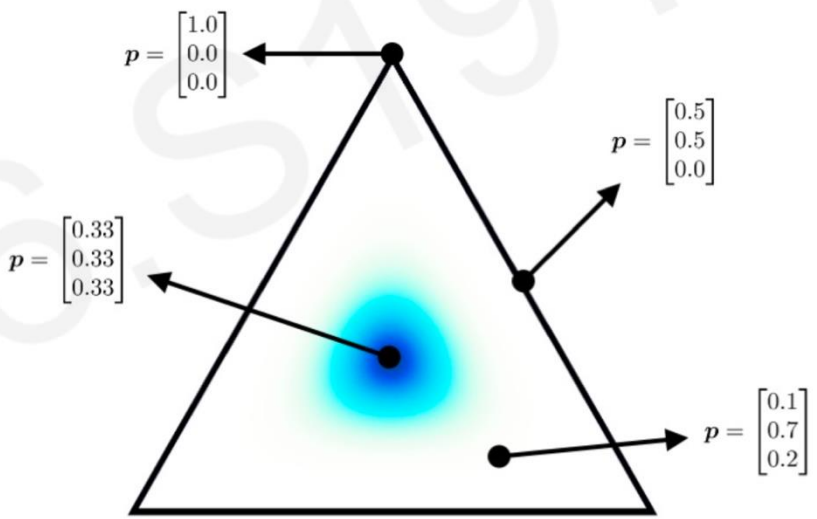
$$y \in \{1, \cdots, K\}$$

$$y \sim \text{Categorical}(\boldsymbol{p})$$

Class Labels | Likelihood function | Distribution parameters (probabilities)

$$\boldsymbol{p} \sim \text{Dirichlet}(\boldsymbol{\alpha})$$

Distribution parameters | Evidential Prior | Model parameters



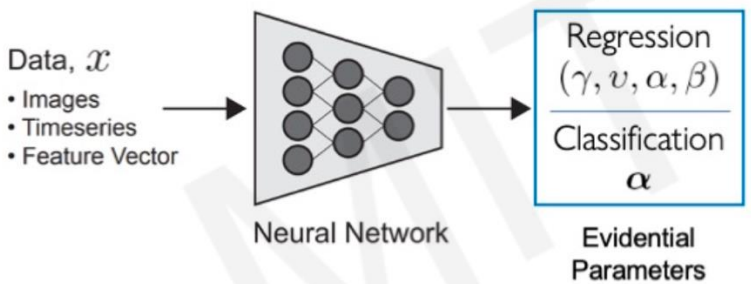
$$K = 3; \quad \boldsymbol{\alpha} = (5, 5, 5)$$

Avi Amini, MIT lecture, 2021

Model and training

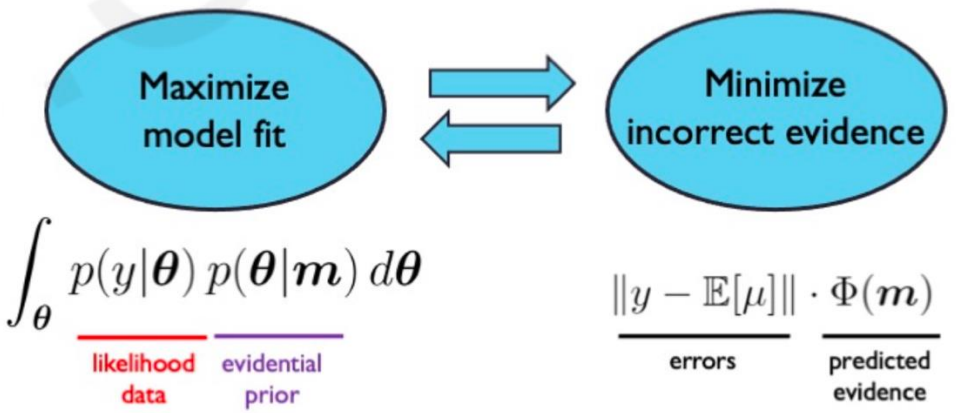
Model

Train the network to output the parameters of an evidential distribution



Optimization

Multi-objective training:



https://arxiv.org/abs/1910.02600
https://github.com/aamini/evidential-deep-learning

OOD – Mahalobious distance

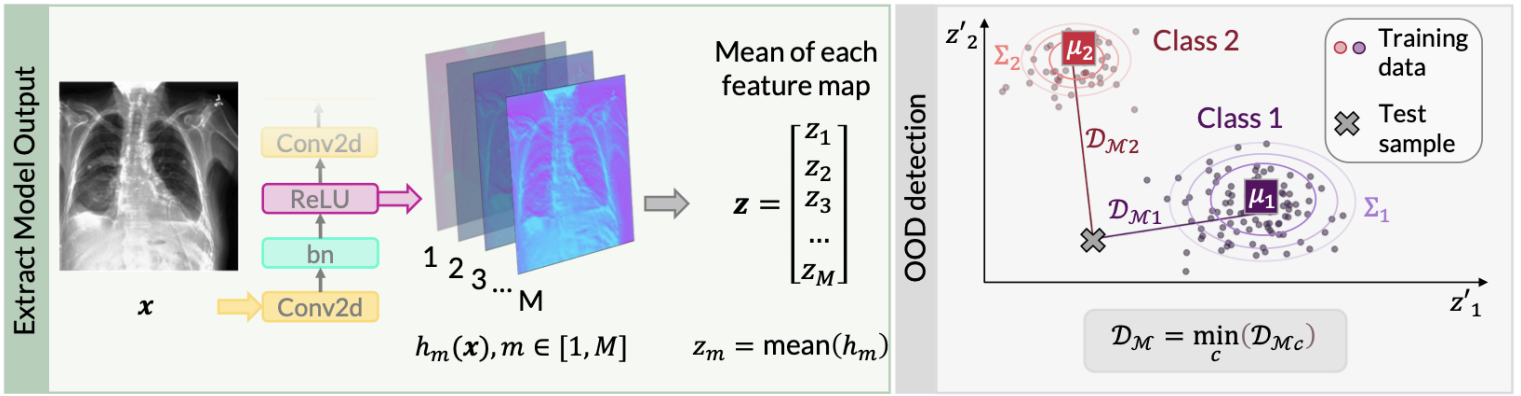


Fig. 1: (Left) Method to extract embeddings after a network module. (Right) Mahalanobis score $\mathcal{D}_{\mathcal{M}}$ of an input to the closest training class centroid.

$$\mathcal{D}_{\mathcal{M}_c}(\mathbf{x}) = \sum_{i=1}^{M} (\mathbf{z} - \boldsymbol{\mu}_\mathbf{c})\ \boldsymbol{\Sigma}_\mathbf{c}^{-1}\ (\mathbf{z} - \boldsymbol{\mu}_\mathbf{c})^T, \qquad \mathcal{D}_{\mathcal{M}}(\mathbf{x}) = \min_c \{\mathcal{D}_{\mathcal{M}_c}(\mathbf{x})\}.$$

https://arxiv.org/pdf/2309.01488

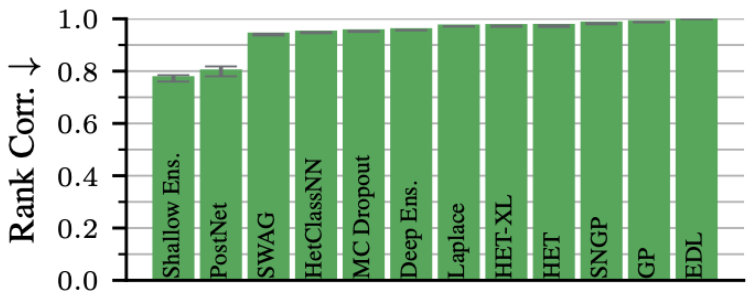
THEORY PART FIVE: WHAT ELSE IS THERE?

Benchmarking Uncertainty Disentanglement: Specialized Uncertainties for Specialized Tasks

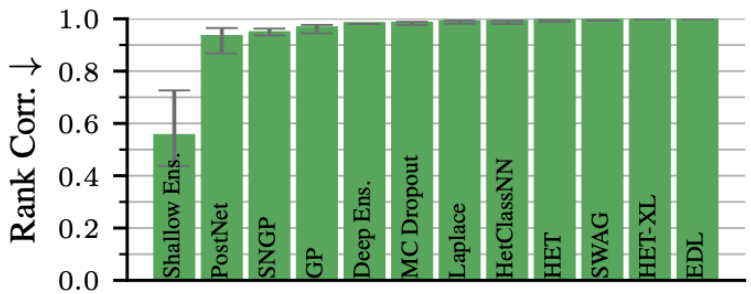
Bálint Mucsányi
University of Tübingen
b.h.mucsanyi@gmail.com

Michael Kirchhof
University of Tübingen

Seong Joon Oh
University of Tübingen
Tübingen AI Center

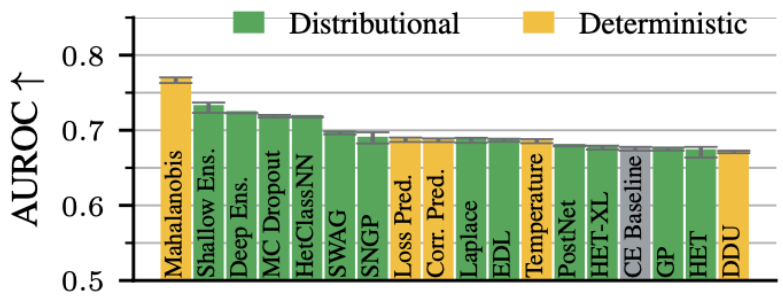


(a) ImageNet results. All twelve distributional methods exhibit a high rank corr. (≥ 0.78).

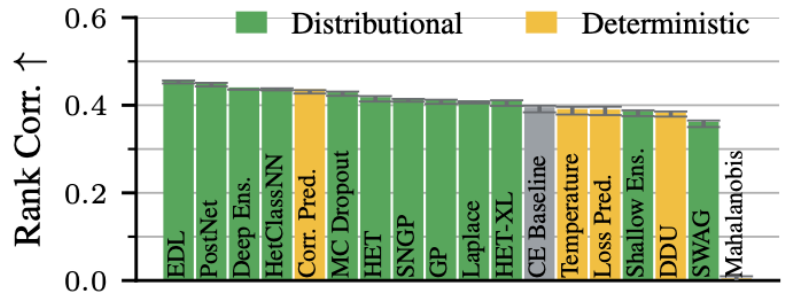


(b) CIFAR-10 results. Eleven out of twelve distributional methods exhibit a strong rank corr. (≥ 0.93).

Figure 2: Rank correlation between the aleatoric and epistemic estimates obtained by the IT decomposition on ImageNet (left) and CIFAR-10 (right). The two uncertainty components are strongly correlated for most methods, violating a necessary condition of their disentanglement.



(a) OOD detection AUROC results. OOD samples are perturbed by ImageNet-C corruptions of severity two. Mahalanobis, the best method, is trained specifically to distinguish OOD data of this severity.



(b) Rank correlation of uncertainty estimators and the GT aleatoric uncertainty on ImageNet. The entropy of the ImageNet-ReaL label distributions is used as GT aleatoric uncertainty.

Figure 3: Performance of uncertainty quantification methods on epistemic (left) and aleatoric (right) uncertainty tasks on the ImageNet validation dataset.

the end