SIPTA Summer School:
Machine Learning and Uncertainty Quantification

Alan Chau, Eyke Hüllermeier, Yusuf Sale

SIPTA Summer School on Imprecise Probabilities,
Munich, 31 July, 2026

Notation

$\boldsymbol{x} \in \mathcal{X}$	instance, instance space
$y \in \mathcal{Y}$	outcome, outcome/label space
$\mathbb{P}(\mathcal{Y})$, Δ_K	set of distributions on $\mathcal{Y}$, probability simplex
p, P	probability distribution, measure
$\mathcal{D} \in \mathbb{D}$	sample (training data) , sample space
L	loss function ($\mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ or $\mathcal{Y} \times \mathbb{P}(\mathcal{Y}) \longrightarrow \mathbb{R}$)
$h \in \mathcal{H}$	hypothesis (map $\mathcal{X} \longrightarrow \mathbb{Y}$ or $\mathcal{X} \longrightarrow \mathbb{P}(\mathbb{Y})$), hypothesis space
$\mathbb{E}$	expectation
$\mathbb{P}(\mathbb{P}(\mathcal{Y}))$	set of second-order distributions
$Q = H(\boldsymbol{x})$	second-order prediction /predictor
L_E	second-order (epistemic) loss
TU, AU, EU	total, aleatoric, epistemic uncertainty

Machine learning and model induction

- **Model induction**: passing from a specific data sample to a general (though hypothetical) model describing the data generating process, or at least certain properties of this process
- A learning (data analysis) algorithm **A** is given as input a set

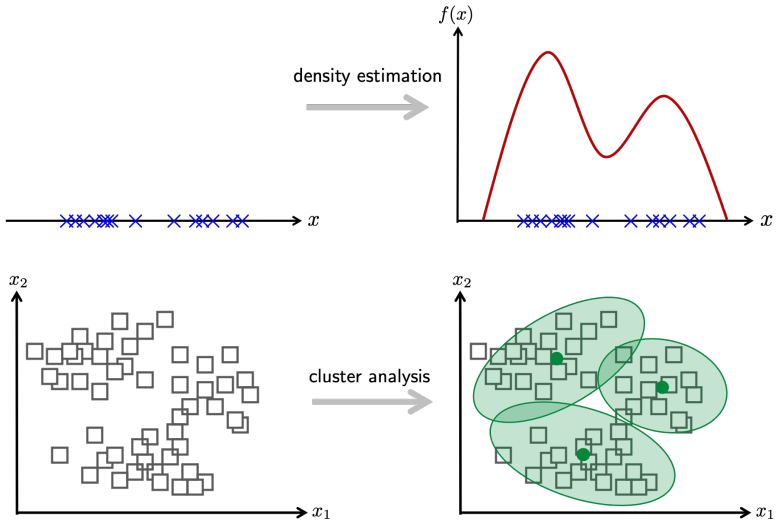
$$\mathcal{D} = \left\{ \boldsymbol{z}_i \right\}_{i=1}^{N} \in \mathcal{Z}^N$$

  of data points $\boldsymbol{z}_i \in \mathcal{Z}$; as output, it produces a **model** $h \in \mathcal{H}$, where $\mathcal{H}$ is a predefined model class (hypothesis space).
- Formally, the algorithm can hence be seen as a mapping

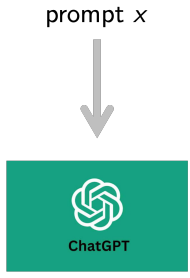
$$\mathbf{A} : \mathbb{D} \longrightarrow \mathcal{H} \ ,$$

  where $\mathbb{D}$ is the space of potentially observable data samples (the sample space).

Machine learning and model induction



Machine learning and model induction



Supervised learning

- In **supervised learning**, the data space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ comprises a distinguished dimension for a specific **target variable** Y.
- The interest is to learn a mapping (hypothesis)

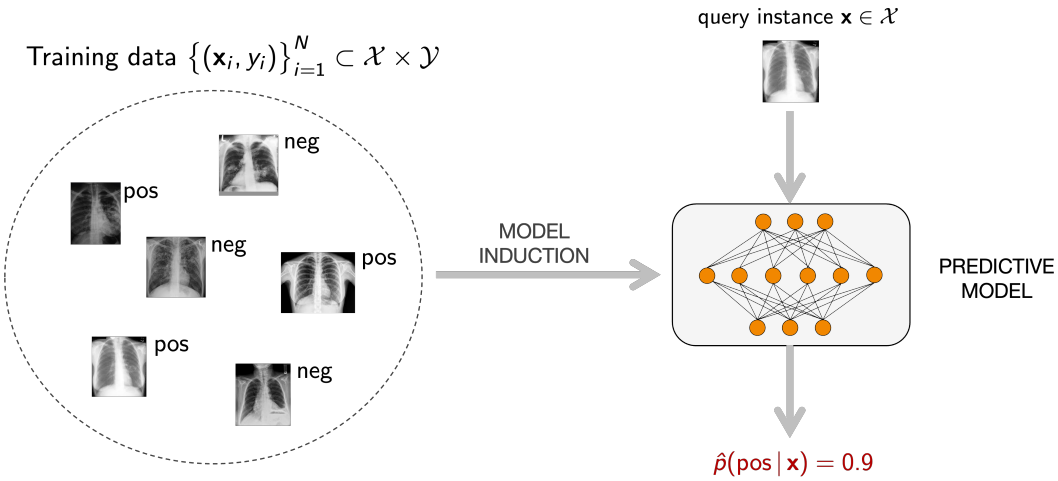
$$h : \mathcal{X} \longrightarrow \mathcal{Y} \quad \text{or} \quad h : \mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$$

  that models the dependence of the target $y \in \mathcal{Y}$ (outcome, response) on the context $\boldsymbol{x} \in \mathcal{X}$ (input, instance, covariates).
- The **hypothesis space** $\mathcal{H}$ consists of a class of such mappings.
- To this end, the learning algorithm **A** is given a set

$$\mathcal{D} = \left\{ (\boldsymbol{x}_i, y_i) \right\}_{i=1}^{N} \in (\mathcal{X} \times \mathcal{Y})^N$$

  of **training examples** $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ as input.

Machine learning: the setting of supervised learning



Supervised machine learning

- Given **training data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$, typically assumed to be i.i.d. according to some (unknown) measure P on $\mathcal{X} \times \mathcal{Y}$, a **hypothesis space** $\mathcal{H}$ of predictors $\mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$, and a **loss function**

$$L : \mathcal{Y} \times \mathbb{P}(\mathcal{Y}) \longrightarrow \mathbb{R}\,,$$

the learner seeks to find

$$h^* \in \arg\min_{h \in \mathcal{H}} R(h)\,,$$

i.e., a predictor $h : \mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$ with **minimal risk** (expected loss)

$$R(h) := \mathbb{E}_{(\boldsymbol{x},y)\sim P} L(y, h(\boldsymbol{x})) = \int_{\mathcal{X}\times\mathcal{Y}} L(y, h(\boldsymbol{x}))\, d\, P(\boldsymbol{x}, y)\,. \tag{1}$$

Risk minimization

- Obviously, the risk of a model h cannot be computed directly, since the probability measure P in (1), which specifies the data generating process, is unknown.
- What is often minimized as a substitute, therefore, is the **empirical risk**

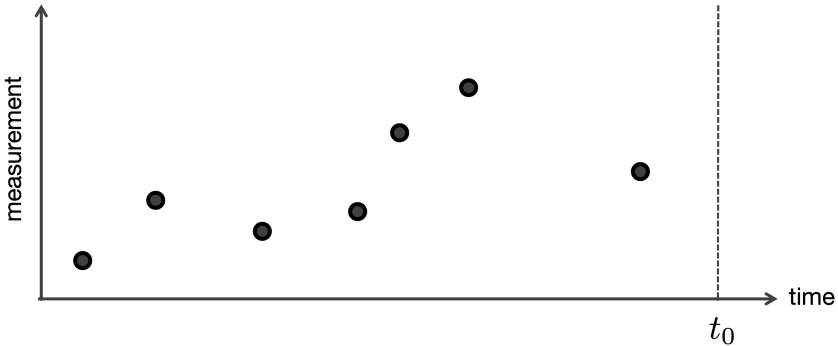
$$\mathcal{R}_{emp}(h) = \frac{1}{N} \sum_{i=1}^{N} L\big(y_i, h(\boldsymbol{x}_i)\big) \ , \tag{2}$$

i.e., the average loss on the training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ — thus, the idea is to approximate the true risk minimizer h^* by the **empirical risk minimizer**.

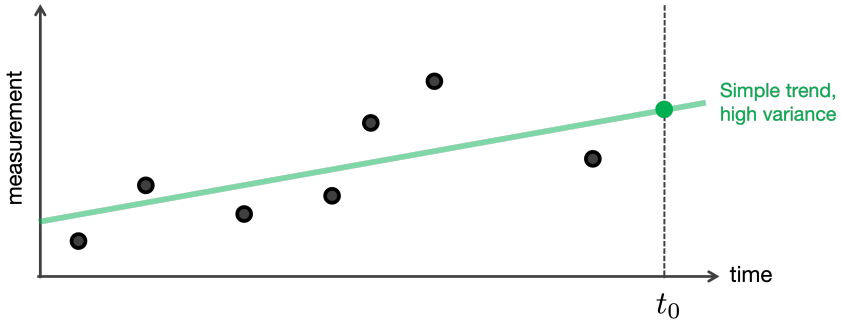
Generalization

- A major theme is the question of **generalization**: How should the model be trained so as to achieve a strong generalization performance, i.e., a low risk (1)?
- An important insight is that a low **empirical risk** (2) does not imply a strong generalization performance, i.e., low **risk** (1).
- Moreover, the generalization performance strongly depends on the right choice of the **model class** $\mathcal{H}$.
- This class should have the right **capacity**, which is well attuned to the complexity of the target function and the amount of training data.
- If the capacity of $\mathcal{H}$ is too high, there is a tendency to **overfit** the data (low empirical risk but poor generalization); the other way around, if the capacity is too low, the learning algorithm will **underfit** the data.

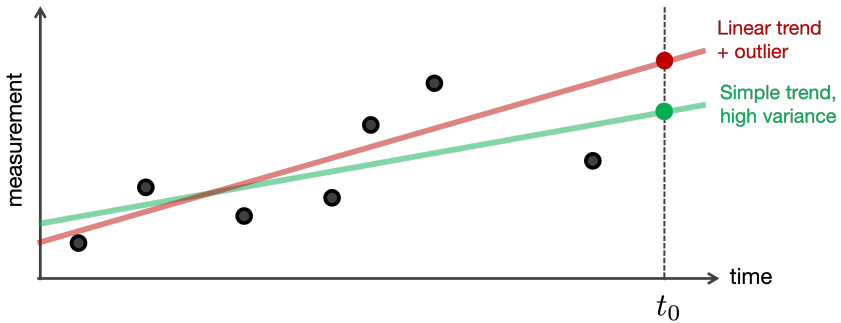
Generalization



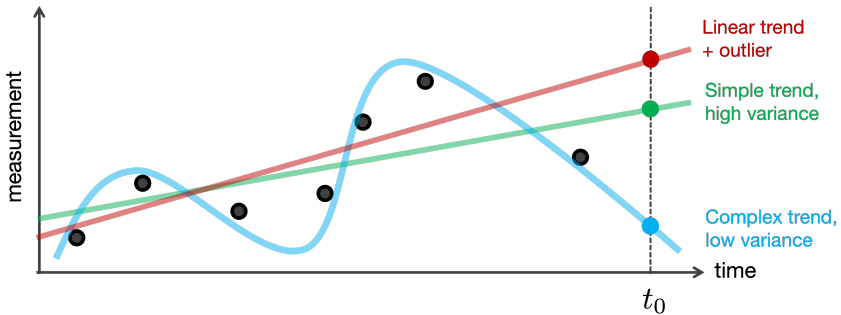
Generalization



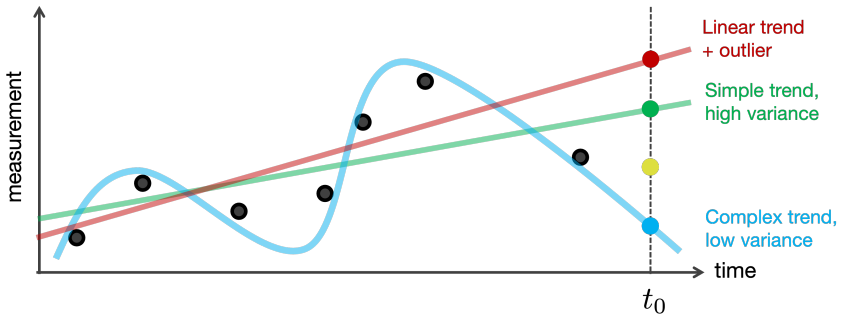
Generalization



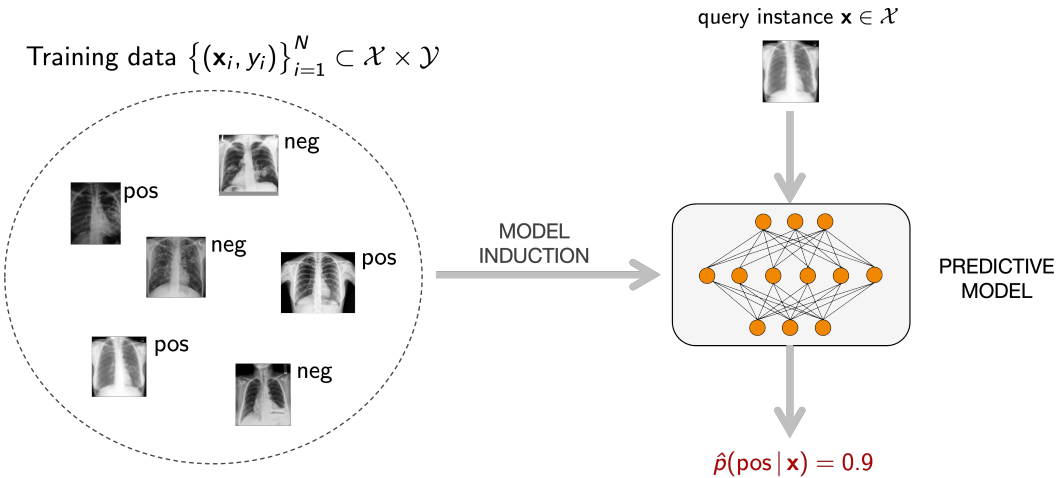
Generalization



Generalization



Machine learning: the setting of supervised learning



Lack of uncertainty-awareness of ML systems

- Predictions by state-of-the-art neural network:

"umbrella" with confidence 97 %

"skirt" with confidence 96 %

Lack of uncertainty-awareness of ML systems

- Hallucination in large language models (Shorinwa *et al.*, 2024):

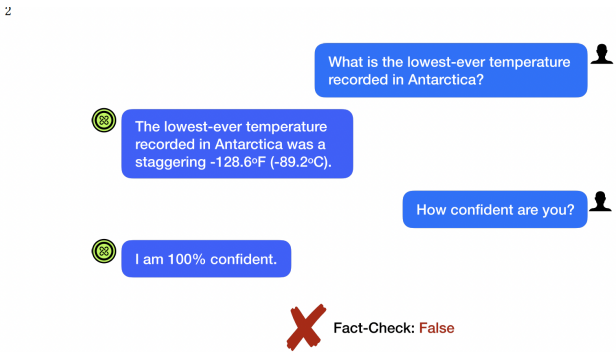
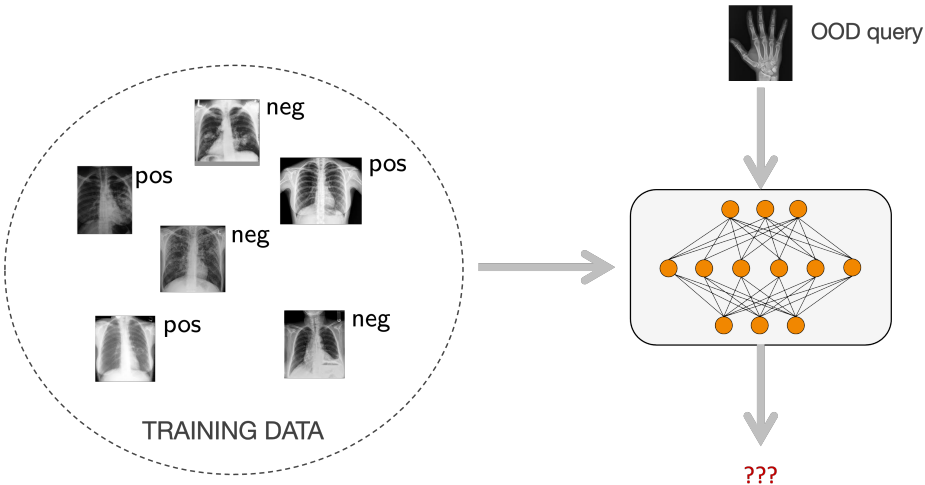


Fig. 1. A user asks an LLM the question: *What is the lowest-ever temperature recorded in Antarctica?*; in response, the LLM answers definitively. Afterwards, the user asks the LLM how confident the LLM is. Although the LLM states that it is "100% confident," the LLM's response fails to pass a fact-check test. Confidence scores provided by LLMs are generally miscalibrated. UQ methods seek to provide calibrated estimates of the confidence of LLMs in their interaction with users.

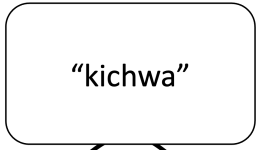
Out-of-distribution data: "I don't know"



Aleatoric versus epistemic uncertainty

Landling Heads $1/2$

Landling Tails $1/2$



Means Head $1/2$

Means Tail $1/2$

"Not knowing the chance of mutually exclusive events and knowing the chance to be equal are two quite different states of knowledge"

Ronald Fisher (1890-1962)

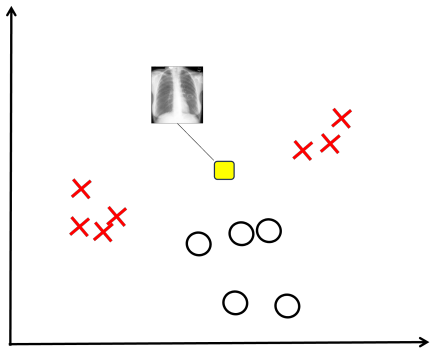
Aleatoric versus epistemic uncertainty

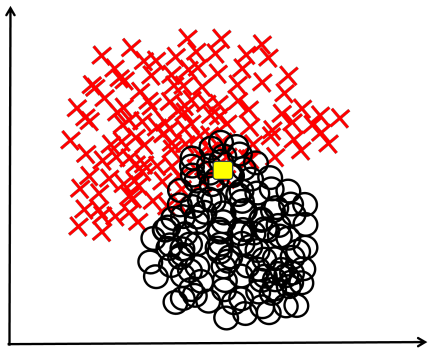
- **Aleatoric** (statistical) uncertainty
  - refers to the notion of **randomness**, that is, the variability in the outcome which is due to inherently random effects,
  - is a property of the **data-generating process**,
  - and as such **irreducible**.

- **Epistemic** (systematic) uncertainty
  - refers to uncertainty caused by a **lack of knowledge**, i.e.,
  - to the epistemic state of the **agent** (e.g., learning algorithm),
  - can in principle be **reduced** on the basis of additional information.

Aleatoric versus epistemic uncertainty in ML

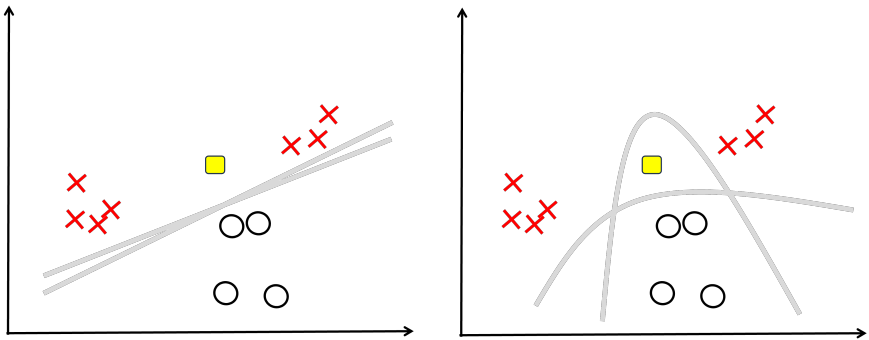
- This distinction is relevant for machine learning, too, where the learner's state of knowledge depends (*inter alia*) on the amount of data seen so far:





Assumptions and prior knowledge

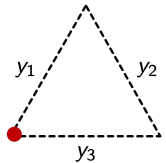
- Assuming a linear model (left), the query can be safely classified as positive, while relaxing the model assumptions leads to increased uncertainty (right):



Uncertainty representation and levels of uncertainty-awareness

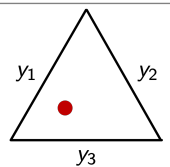
Deterministic predictor
$h : \mathcal{X} \longrightarrow \mathcal{Y}$

$\mathbf{x} \mapsto$



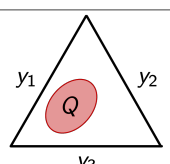
Probabilistic predictor
$h : \mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$

$\mathbf{x} \mapsto$



Second-order predictor
$H : \mathcal{X} \longrightarrow \mathbb{Q}(\mathbb{P}(\mathcal{Y}))$

$\mathbf{x} \mapsto$



Uncertainty representation and levels of uncertainty-awareness

Deterministic predictor

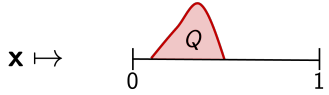
$h : \mathcal{X} \longrightarrow \mathcal{Y}$

Probabilistic predictor

$h : \mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$

Second-order predictor

$H : \mathcal{X} \longrightarrow \mathbb{Q}(\mathbb{P}(\mathcal{Y}))$



Agenda

1. Introduction
2. **First-order uncertainty representation**
3. Second-order uncertainty representation
4. Uncertainty quantification and evaluation
5. Summary and outlook

Aleatoric versus epistemic uncertainty in ML

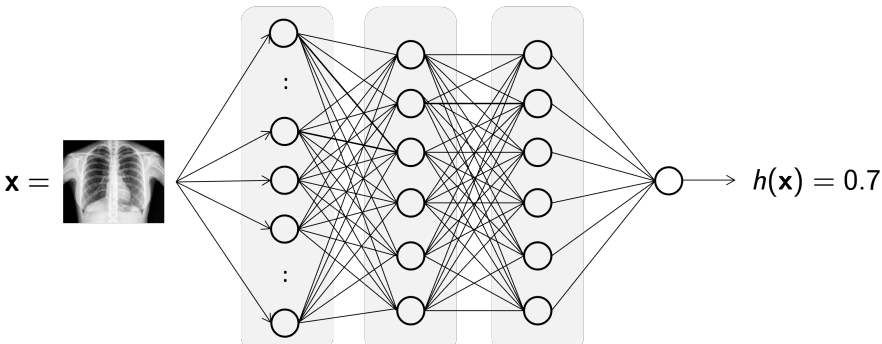
- Assuming that data is generated (i.i.d.) according to a **joint probability distribution** $p \in \mathbb{P}(\mathcal{X} \times \mathcal{Y})$, the true probability of outcome $y \in \mathcal{Y}$ given $\boldsymbol{x} \in \mathcal{X}$ is given by

$$p(y \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}, y)}{\sum_{y' \in \mathcal{Y}} p(\boldsymbol{x}, y')} \,. \qquad (\star)$$

- Thus, even if p is known, **aleatoric uncerainty** about the outcome remains, unless $p(y \mid \boldsymbol{x})$ degenerates to a Dirac distribution (deterministic case).
- Additional **epistemic uncertainty** is caused by the learner's lack of knowledge of p.
- Instead of $(\star)$, the learner delivers an **approximation** that comes from an empirically induced predictor $h : \mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$:

$$\hat{p}(y \mid \boldsymbol{x}) = h(\boldsymbol{x})$$

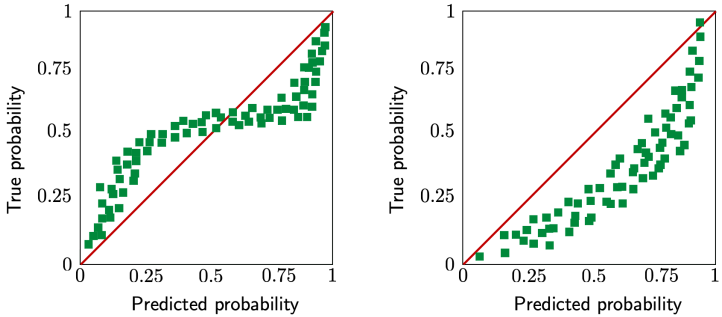
Probabilistic classification: (deep) neural networks



Should theoretically deliver **well-calibrated probabilities** $h(\boldsymbol{x}) \approx P(y = 1 \mid \boldsymbol{x})$ if trained with **proper scoring rules** as loss functions, such as $L(y, p) = (y - p)^2$ ...

Doesn't always work in practice, however ...

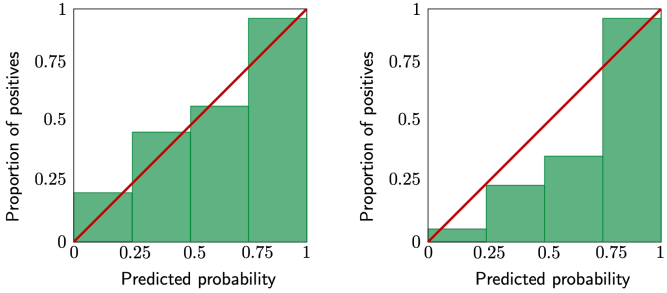
- Examples of overconfidence (left) and systematic overestimation (right):



- A (binary) classifier $h : \mathcal{X} \longrightarrow [0, 1]$ is called **calibrated** if, for all $0 \leq \alpha \leq 1$,

$$P\Big(y = 1 \mid \underbrace{h(\boldsymbol{x})}_{\hat{p}(y=1 \mid \boldsymbol{x})} = \alpha\Big) = \alpha\,.$$

Estimation and visual evaluation of calibration via reliability diagrams



- Expected calibration error:

$$ECE = \sum_{j=1}^{B} \frac{|\mathcal{B}_j|}{N} \left| \bar{p}(\mathcal{B}_j) - \hat{p}(\mathcal{B}_j) \right| ,$$

with average observed and average predicted probabilities

$$\bar{p}(\mathcal{B}_j) = \frac{1}{|\mathcal{B}_j|} \sum_{i \in \mathcal{B}_j} y_i \,, \qquad \hat{p}(\mathcal{B}_j) = \frac{1}{|\mathcal{B}_j|} \sum_{i \in \mathcal{B}_j} h(\boldsymbol{x}_i) \,.$$

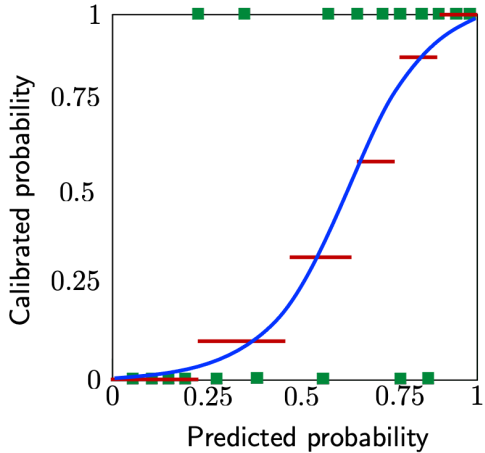
Post-hoc calibration methods (Filho *et al.*, 2023)

- Platt scaling
- Temperature scaling (multi-class)
- Histogram binning and isotonic regression
- Bayesian binning
- Beta calibration
- Dirichlet calibration (multi-class)
- ...

... all fit a **calibration map**

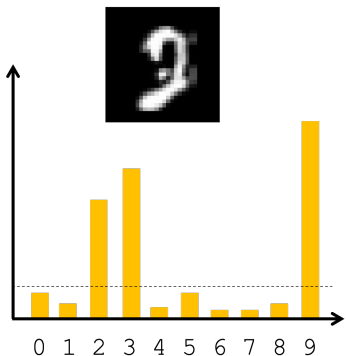
$c : \mathbb{R} \longrightarrow [0,1]$ resp. $\mathbb{R}^K \longrightarrow [0,1]^K$

to data $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \{0,1\}$.



Set-valued prediction

- From (calibrated) probability estimation to **set-valued prediction**:



$\longrightarrow \qquad P\Big(y \in Y = \{2,3,9\}\Big)$ w.h.p.

Conformal prediction

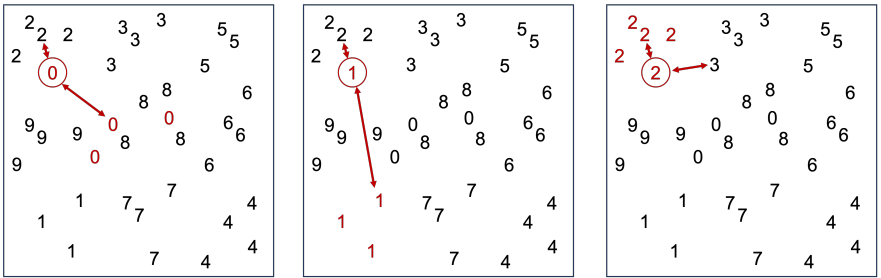
- **Conformal prediction** (Balasubramanian *et al.*, 2014) is a distribution-free framework for uncertainty quantification and reliable prediction that is rooted in classical frequentist statistics and hypothesis testing.
- Instead of point predictions, CP makes **set-valued predictions** $C \subseteq \mathcal{Y}$ covering the true outcome with high probability:

Could be a 0? **NO**
Could be a 1? **NO**
Could be a 2?
⋮
Could be a 8? **NO**
Could be a 9?

$\longrightarrow \quad P\Big(y \in C = \{2, 3, 9\}\Big)$ w.h.p.

Nonconformity

- Hypothesis testing relies on relative order of **nonconformity scores** $\alpha = f(\boldsymbol{x}, y)$ assigned to each (hypothetical) data point $(\boldsymbol{x}, y)$:



- Example: $f(\boldsymbol{x}, y) =$ distance to class $y \div$ distance to any other class
- Deriving nonconformity from a predictor h, e.g., $f(\boldsymbol{x}, y) = 1 - p_h(y \mid \boldsymbol{x})$, allows for "conformalising" learning methods in a systematic way.

Conformal prediction

- Applying a nonconformity function to the sequence of observations, with a hypothetical choice of $y = y_{N+1}$, yields a sequence of scores $\alpha_i = f(\boldsymbol{x}_i, y_i)$:

$$\begin{array}{rcccccc} f: & (\boldsymbol{x}_1, y_1), & (\boldsymbol{x}_2, y_2), & \ldots & (\boldsymbol{x}_N, y_N), & (\boldsymbol{x}_{N+1}, y) \\ & \alpha_1, & \alpha_2, & \ldots & \alpha_N, & \alpha_{N+1} \end{array}$$

- Denote by σ the permutation of $\{1, \ldots, N+1\}$ that sorts the scores in increasing order, i.e., such that

$$\alpha_{\sigma(1)} \leq \ldots \leq \alpha_{\sigma(N+1)} \,.$$

- Under the assumption that the hypothetical choice of y_{N+1} is in agreement with the true data-generating process, and that this process has the property of **exchangeability**, every permutation σ has the same probability of occurrence.

Excursus: Exchangeability

- A (finite) **sequence of random variables** $Z_1, Z_2, \ldots, Z_N$ is exchangeable, if the following holds for any permutation $\sigma : [N] \longrightarrow [N]$: the joint probability distribution of the random variables is the same as the joint distribution of the permuted sequence $Z_{\sigma(1)}, Z_{\sigma(2)}, \ldots, Z_{\sigma(N)}$.
- This means that the cumulative distribution function

$$F_{Z_1,\ldots,Z_N} : \mathbb{R}^N \longrightarrow [0, 1]$$

  is **symmetric** in its arguments.
- Exchangeability is weaker than the i.i.d. assumption (the latter implies the former but not the other way around).

Conformal prediction

- Consequently, under the null hypothesis $y_{N+1} = y$, the probability that α_{N+1} is among the $\epsilon\,\%$ highest nonconformity scores should be low.
- This notion can be captured by the **p-values** associated with the candidate y, defined as

$$p(y) := \frac{\#\{i \in \{1, \ldots, N+1\} \mid \alpha_i \geq \alpha_{N+1}\}}{N+1}$$

- According to what we said, the probability that $p(y) < \epsilon$ (i.e., α_{N+1} is among the $\epsilon\,\%$ highest α-values) is upper-bounded by ϵ.
- Thus, the null hypothesis can be rejected for those candidates y for which $p(y) < \epsilon$.

Conformal prediction

- **Theorem (marginal coverage)**: Consider a random sequence of data points $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_N, Y_N), (\boldsymbol{X}_{N+1}, Y_{N+1})$ and denote by $\alpha_i = f(\boldsymbol{X}_i, Y_i)$ the corresponding nonconformity scores. For $\epsilon \in (0, 1)$, let

$$C(\boldsymbol{X}_{N+1}) = \left\{ y \,|\, \alpha_{N+1} = f(\boldsymbol{X}_{N+1}, y) \leq \hat{q}_{N+1} \right\},$$

where $\hat{q}_{N+1}$ is the $\lceil (N+1)(1-\epsilon) \rceil$-th empirical quantile of $\{\alpha_1, \ldots, \alpha_N, \infty\}$. Then, under the assumption of exchangeability,

$$P\big(Y_{N+1} \in C(\boldsymbol{X}_{N+1})\big) \geq 1 - \epsilon\,.$$

Inductive conformal prediction

- In its original form, conformal prediction realises **transductive inference**, i.e., inference directly targeting a query instance (known by the learner) without inducing a general model beforehand.
- This is computationally costly, as it possibly requires refitting a model to the data in each iteration.
- **Inductive conformal prediction** (ICP) or **split conformal prediction** is an alternative that is computationally less expensive.
- ICP splits the training data into
  - proper training data $\mathcal{D}_T = \left\{(\boldsymbol{x}_i, y_i)\right\}_{i=1}^{M}$,
  - calibration data $\mathcal{D}_C = \left\{(\boldsymbol{x}_j, y_j)\right\}_{j=M+1}^{N}$.

Inductive conformal prediction

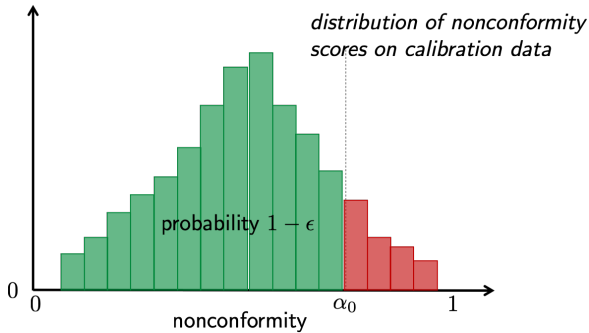
- A **predictive model** is trained on the proper training set.
- Using this model, the non-conformity scores $\alpha_{M+1}, \ldots, \alpha_N$ are computed for the examples in the **calibration data**.
- Given a new query $\boldsymbol{x}_{N+1}$ and a candidate $y \in \mathcal{Y}$, the score α_{N+1} is determined in the same way.
- The p-value for y is then given by

$$p = \frac{1}{N - M + 1} \sum_{j=M+1}^{N+1} [\![\alpha_j \geq \alpha_{N+1}]\!] .$$

Conformal prediction

- The **calibration data** is used to determine critical values α_0 (w.h.p., non-conformity of a "real" data point is $\leq \alpha_0$):

$\boldsymbol{x}$	y	$f(\boldsymbol{x}, y)$
$\boldsymbol{x}_1$	y_1	0.42
$\boldsymbol{x}_2$	y_2	0.26
$\boldsymbol{x}_3$	y_3	0.81
$\boldsymbol{x}_4$	y_4	0.17
$\boldsymbol{x}_5$	y_5	0.73
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{x}_N$	y_N	0.41



- This allows for constructing (valid) **prediction sets**:

$$C(\boldsymbol{x}) = \left\{ y \in \mathcal{Y} \,|\, f(\boldsymbol{x}, y) \leq \alpha_0 \right\} = \mathcal{Y} \setminus \left\{ y \in \mathcal{Y} \,|\, f(\boldsymbol{x}, y) > \alpha_0 \right\}$$

Agenda

1. Introduction
2. First-order uncertainty representation
3. **Second-order uncertainty representation**
   - **The Bayesian approach**
   - The Credal approach
4. Uncertainty quantification and evaluation
5. Summary and outlook

Background and Perspective

Background

- PhD in Statistics with Dino Sejdinovic at Oxford UK
- Postdoc with Krikamol Muandet at CISPA Germany
- Now Assistant Professor at Nanyang Technological University Singapore

Research Interests

- Probabilistic and **Imprecise probabilistic** machine learning

Today's agenda

- Examples of learning algorithms that considers **second-order uncertainty**
- Demonstrate Why and how **imprecise probabilities** provide a natural framework for uncertainty-aware prediction.

Predictive uncertainty

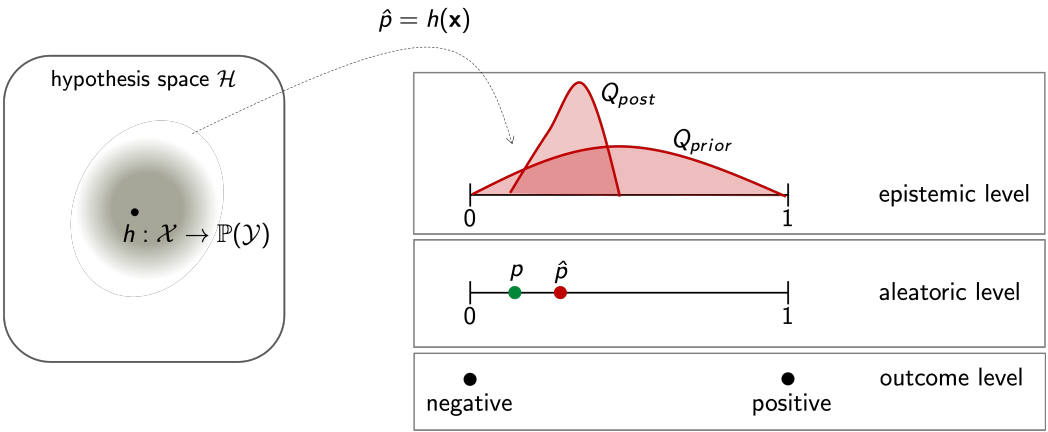
- In machine learning, we are mainly interested in (per-instance) **predictive uncertainty**: Instead of a deterministic prediction $\hat{y}$ of the outcome for a query instance $\boldsymbol{x}$, or a probabilistic prediction $\hat{p} = h(\boldsymbol{x})$, we seek a prediction

$$Q = H(\boldsymbol{x})$$

  adequately representing the learner's epistemic uncertainty.

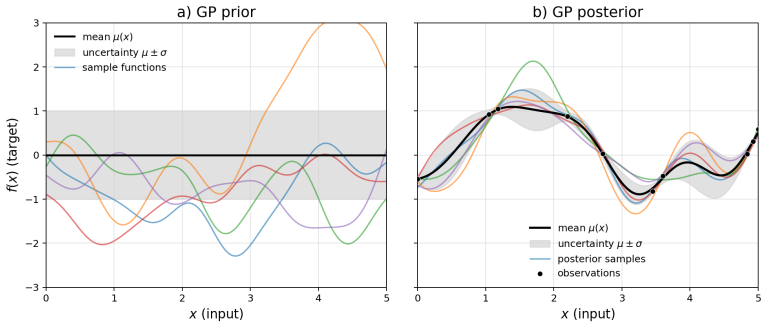
- Various **approaches** have been proposed in the literature:
  - The Bayesian approach
  - The Ensemble approach
  - The Credal approach
  - Generalised loss minimisation
  - ...

The Bayesian approach: a posterior over predictive distributions



Example: Gaussian processes

- Overview: Gaussian process (GP) places prior $p(f)$ over random functions f; through conjugate prior-likelihood to perform tractable posterior update $p(f \mid D)$.
- GPs are perhaps the clearest bridge between Bayesian statistics and ML: they inherit the full Bayesian treatment of uncertainty while providing a practical algorithm for learning **complex nonlinear functions**.



Gaussian processes regression

- A **Gaussian processes (GP)** is a stochastic process characterised by mean function $m : \mathcal{X} \to \mathbb{R}$ and covariance function (kernel) $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, denoted as $\mathcal{GP}(m, k)$.
- For any finite set of elements $\mathbf{x}_1, \ldots, \mathbf{x}_m$, the random vector $[f(\mathbf{x_1}), \ldots, f(\mathbf{x_m})]$ has prior distribution

$$\begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_1) \\ m(\mathbf{x}_2) \\ \vdots \\ m(\mathbf{x}_m) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_m, \mathbf{x}_1) & \cdots & k(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix} \right).$$

- Suppose we observe noisy function values (the likelihood)

$$\mathbf{y} = (y_1, \ldots, y_n)^\top, \qquad y_i = f(\mathbf{x}_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

Gaussian process regression

- To predict the latent function $f_* = f(x_*)$ at a new input $\mathbf{x}_*$, define the joint Gaussian prior

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_X \\ m_* \end{bmatrix}, \begin{bmatrix} K_{XX} + \sigma^2 I & K_{X*} \\ K_{*X} & K_{**} \end{bmatrix} \right),$$

  where

$$K_{XX} = k(X, X), \qquad K_{X*} = k(X, \mathbf{x}_*), \qquad K_{**} = k(\mathbf{x}_*, \mathbf{x}_*).$$

- Since they are jointly Gaussian, **conditioning admits also a closed form:**

$$f_* \mid \mathbf{y}, \sim \mathcal{N}(\mu_*, \sigma_*^2).$$

  where

$$\mu_* = m_* + K_{*X}(K_{XX} + \sigma^2 I)^{-1}(\mathbf{y} - \mathbf{m}_X)$$
$$\sigma_*^2 = K_{**} - K_{*X}(K_{XX} + \sigma^2 I)^{-1} K_{X*}$$

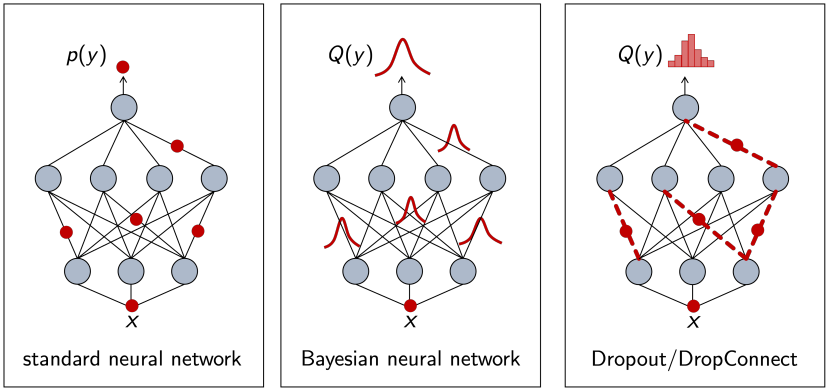
- And subsequently we have $\mathrm{Var}(y_* \mid \mathbf{y}) = \underbrace{\sigma_*^2}_{\textbf{epistemic}} + \underbrace{\sigma^2}_{\textbf{aleatoric}}$

GP is great but suffers from **scalability** and **flexibility** that modern deep neural nets bring.

Can we combine the **representation learning** of deep networks with the **uncertainty quantification** of Bayesian inference?

Example: Bayesian neural networks

- Bayesian neural networks instead place a prior $\theta \sim p(\theta)$, representing uncertainty over the model parameters.



Bayesian learning of neural networks

- Given data $D = \{(x_i, y_i)\}_{i=1}^n$, prior $p(\theta)$, likelihood $p(D \mid \theta) = \prod_{i=1}^n N(y_i; f_\theta(x_i), \sigma^2)$, Bayes' rule gives

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}.$$

- The denominator

$$p(D) = \int p(D|\theta)p(\theta)\, d\theta$$

  is the **model evidence** (marginal likelihood), intractable due to high dimensional parameters, non-conjugacy, and non-linear likelihood.

- Requires approximate methods such as **MCMC**, **Laplace approximation**, or **variational inference**.

Variational inference

- Rather than computing the exact posterior, choose a tractable family $\{q_\phi(\theta) : \phi \in \Phi\}$, e.g. Gaussian distributions.
- Find the member of this family that is closest to the true posterior:

$$q^*_\phi(\theta) = \arg\min_{q_\phi} D_{\mathrm{KL}}(q_\phi(\theta) \| p(\theta|D)) \,.$$

- Unfortunately, computing $p(\theta|D)$ requires the intractable evidence term $p(D)$.
- Instead, maximise the **Evidence Lower Bound (ELBO)**, which is equivalent to the above optimisation.

$$\boxed{\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\phi}[\log p(D|\theta)] - D_{\mathrm{KL}}(q_\phi(\theta) \| p(\theta)).}$$

Why maximise the ELBO?

$$D_{\mathrm{KL}}(q_\phi(\theta)\|p(\theta|D)) = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\theta)}{p(\theta|D)}\right] = \mathbb{E}_{q_\phi}\left[\log q_\phi(\theta) - \log \frac{p(D|\theta)p(\theta)}{p(D)}\right].$$

Since $\log p(D)$ does not depend on θ,

$$D_{\mathrm{KL}} = \log p(D) - \underbrace{\left[\mathbb{E}_{q_\phi}[\log p(D|\theta)] - D_{\mathrm{KL}}(q_\phi(\theta)\|p(\theta))\right]}_{\mathcal{L}_{\mathrm{ELBO}}}.$$

Hence,

$$\boxed{D_{\mathrm{KL}}(q_\phi(\theta)\|p(\theta|D)) = \log p(D) - \mathcal{L}_{\mathrm{ELBO}}.}$$

Since the evidence $\log p(D)$ is constant with respect to q_ϕ,

$$\boxed{\arg\min_{q_\phi} D_{\mathrm{KL}}(q_\phi\|p(\theta|D)) = \arg\max_{q_\phi} \mathcal{L}_{\mathrm{ELBO}}.}$$

Weight-space prior for NNs.. is it worth it?

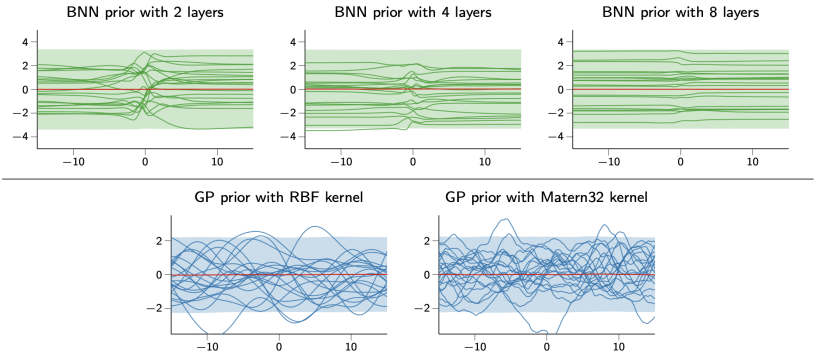


Figure 1: (**Top**) Sample functions of a fully-connected BNN with 2, 4 and 8 layers obtained by placing a Gaussian prior on the weights. (**Bottom**) Samples from a GP prior with two different kernels.

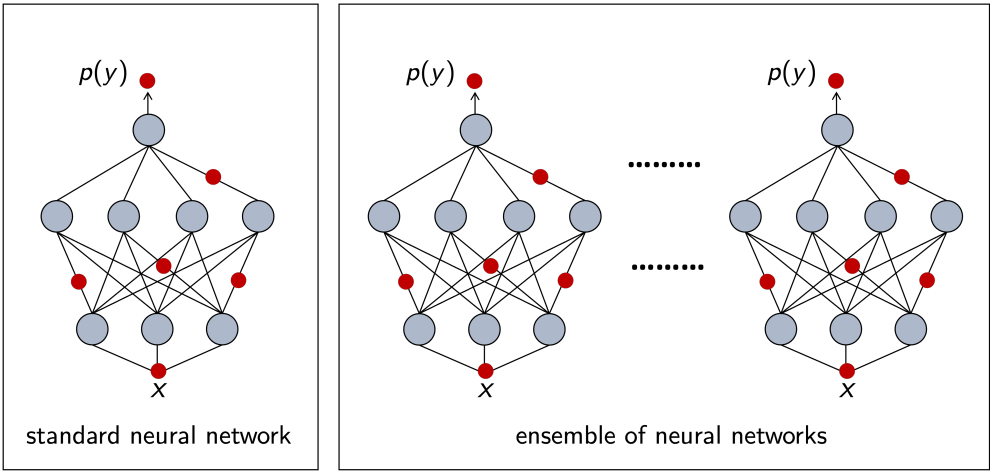
Figure: Simple weight priors can induce opaque function-space priors. We might want more pragmatic approaches.

BNNs provide a principled Bayesian treatment of uncertainty, but posterior inference is **computationally challenging**.

This naturally raises the question: **can we obtain meaningful uncertainty estimates without performing Bayesian inference?**

Example: Deep Ensemble

- Ensembles of (deep) neural networks (Lakshminarayanan *et al.*, 2017) is a simpler though still costly alternative:



Deep ensembles

- A simple alternative is to **independently train** $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}$ from different random initialisations (and optionally different data orderings).
- At prediction time, average the individual predictions:

$$p(y \mid x, D) \approx \frac{1}{M} \sum_{m=1}^{M} p(y \mid x, \theta^{(m)}).$$

- **Model disagreement** provides a practical measure of epistemic uncertainty:

$$\text{large disagreement} \implies \text{high uncertainty.}$$

- **Advantages:** easy to implement, highly scalable, and often competitive with approximate Bayesian neural networks.

Are Deep Ensembles Approximately Bayesian?

- Deep ensembles are trained by **empirical risk minimisation**, whereas Bayesian methods learn through **Bayes' rule**.
- Nevertheless, Bayesian inference admits the **variational characterisation**

$$\pi(\cdot \mid \mathbf{D}) = \arg \min_{q \in \mathcal{P}(\Theta)} \left\{ \mathbb{E}_{\theta \sim q}\left[-\log p(\mathbf{D} \mid \theta) \right] + D_{\mathrm{KL}}(q \| \pi) \right\}.$$

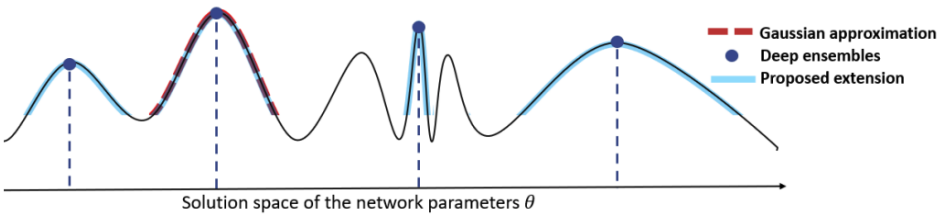
- For i.i.d. observations,

$$q^\star = \arg\min_q \{ \underbrace{\sum_{i=1}^{n} \mathbb{E}_{\theta \sim q}\left[\ell(\theta; X_i, Y_i) \right]}_{\text{expected empirical loss}} + \underbrace{D_{\mathrm{KL}}(q \| \pi)}_{\text{prior regularisation}} \}.$$

- This analogy motivates Bayesian interpretations of deep ensembles, although ordinary ensemble training is not, by itself, exact posterior inference.

Deep ensemble as approximate Bayesian (cont.)

- Upon specific prior $p(\theta)$, DE solutions can be interpret as samples from multiple modals of the posterior weight distribution.



Agenda

1. Introduction
2. First-order uncertainty representation
3. **Second-order uncertainty representation**
   - The Bayesian approach
   - **The Credal approach**
4. Uncertainty quantification and evaluation
5. Summary and outlook

From distributions over probabilities to sets of probabilities

- A Bayesian second-order predictor assigns a **probability distribution over predictive distributions**:

$$Q_x \in \mathbb{P}\big(\mathbb{P}(\mathcal{Y})\big).$$

- A credal predictor instead returns a **set of admissible predictive distributions**:

$$Q_x \subseteq \mathbb{P}(\mathcal{Y}).$$

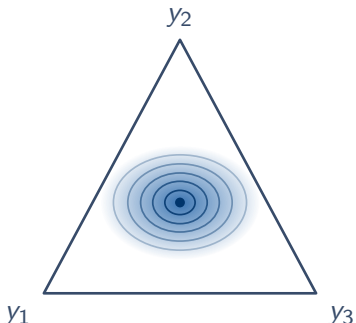
- For example, a deep ensemble already supplies a finite collection of predictive distributions

$$\mathcal{S}_x = \Big\{ p^{(1)}(\cdot \mid x), \ldots, p^{(M)}(\cdot \mid x) \Big\} \subseteq \Delta_K.$$

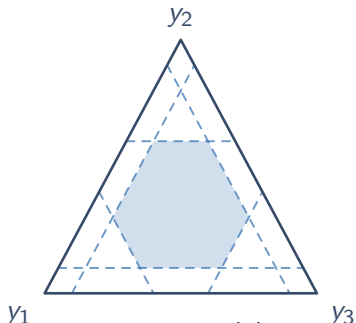
  No need to enforce a second order distribution interpretation to it...

From distributions over probabilities to sets of probabilities

- The two representations answer different questions:
  - How should **plausibility be distributed** among candidate models?
  - Which candidate models should **remain admissible**?



distribution over probabilities



credal set $\mathcal{K}(x)$

Credal prediction involves three separate choices

- A credal prediction pipeline can be viewed as three consecutive modelling decisions.
  1. How should we construct/learn a meaningful collection of **probabilistic judges**

$$\mathcal{S}_x = \{p^{(1)}(\cdot|x), \ldots, p^{(M)}(\cdot|x)\} \subseteq \Delta_K?$$

  2. Given these judges, how should we construct a **credal set**

$$\mathcal{K}(x) \subseteq \Delta_K?$$

  3. Given a credal set, how should we perform **decision making**?
     - ★ quantify uncertainty,
     - ★ predict a label or probability,
     - ★ construct a prediction set.

$$\underbrace{\mathcal{S}_x}_{\text{probabilistic judges}} \implies \underbrace{\mathcal{K}(x)}_{\text{credal set}} \implies \underbrace{\text{prediction / uncertainty / decision}}_{\text{actionable output}}$$

Step #2: Constructing a credal prediction: probability bounds

Suppose a learning procedure produces

$$\mathcal{S}_x = \left\{ p^{(1)}(x), \ldots, p^{(M)}(x) \right\} \subseteq \Delta_K.$$

- For each class k, take the smallest and largest probability from $\mathcal{S}_x$:

$$\underline{p}_k(x) = \min_{m=1,\ldots,M} p_k^{(m)}(x), \qquad \overline{p}_k(x) = \max_{m=1,\ldots,M} p_k^{(m)}(x).$$

- These class-wise intervals define the **box credal set**

$$\boxed{\mathcal{K}_{\text{box}}(x) = \left\{ p \in \Delta_K \;\middle|\; \underline{p}_k(x) \leq p_k \leq \overline{p}_k(x), \quad k = 1, \ldots, K \right\}.}$$

Step #2: Constructing a credal prediction: convex hull

Using the same collection

$$\mathcal{S}_x = \left\{ p^{(1)}(x), \ldots, p^{(M)}(x) \right\} \subseteq \Delta_K,$$

we may instead retain all mixtures of the candidate predictions. Giving us a **Robust Bayesian interpretation** (all possible Bayesian moving averages).

- Define the **finitely generated credal set**

$$\boxed{\mathcal{K}_{\text{hull}}(x) = \text{conv} \left\{ p^{(1)}(x), \ldots, p^{(M)}(x) \right\}.}$$

- Equivalently,

$$\mathcal{K}_{\text{hull}}(x) = \left\{ \sum_{m=1}^{M} \lambda_m p^{(m)}(x) \;\middle|\; \lambda_m \geq 0, \quad \sum_{m=1}^{M} \lambda_m = 1 \right\}.$$

Step #3: Using a credal prediction: precise reduction

A credal set $\mathcal{K}(x)$ may be retained as the final prediction, or reduced to a representative precise probability.

- Choose a transformation $\widehat{p}(x) = T\big(\mathcal{K}(x)\big) \in \Delta_K$.
- Possible precise-probability transformations include:
  - the average of finite generators.
  - a **geometric centroid** or another representative centre of $\mathcal{K}(x)$;
  - the **pignistic probability** $\mathrm{BetP}(m)$ for a belief-function representation;
  - an **intersection probability** $\mathrm{IntP}(\underline{p}, \overline{p})$ for probability intervals.

> Reducing $\mathcal{K}(x)$ to one probability is useful for conventional classification, but necessarily discards some of the represented epistemic uncertainty.

Step #3: Using a credal prediction: prediction is a decision

- In Bayesian decision theory, even a probability forecast is an **action**. Given a strictly proper loss $\ell(q, Y)$, the Bayes-optimal forecast under a precise predictive distribution p is

$$q_p^* \in \arg\min_{q \in \Delta_K} \mathbb{E}_{Y \sim p}[\ell(q, Y)], \qquad q_p^* = p.$$

- Replacing p by a credal predictive set $\mathcal{K}(x)$, different decision criteria produce different predictions:

$$\Gamma\text{-minimax:} \qquad q_\Gamma(x) \in \arg\min_{q \in \Delta_K} \sup_{p \in \mathcal{K}(x)} \mathbb{E}_{Y \sim p}[\ell(q, Y)],$$

$$E\text{-admissibility:} \qquad \mathcal{Q}_E(x) = \bigcup_{p \in \mathcal{K}(x)} \arg\min_{q \in \Delta_K} \mathbb{E}_{Y \sim p}[\ell(q, Y)].$$

  For a strictly proper loss,

$$\mathcal{Q}_E(x) = \mathcal{K}(x).$$

What about step #1? How should we "learn" a meaningful collection of **probabilistic judges** in the first place?

Example 1: Credal wrapper (Wang *et al.*, 2025)

- Start with **any ensemble** of probabilistic predictors

$$h_1, \ldots, h_M : \mathcal{X} \longrightarrow \Delta_K.$$

- At a query x, collect their probabilistic judgements

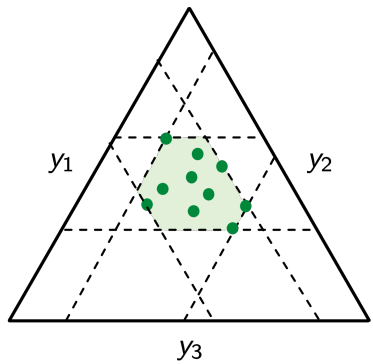
$$p^{(m)}(x) = h_m(x), \qquad m = 1, \ldots, M.$$

- Define class-wise lower and upper probabilities

$$\underline{p}_k(x) = \min_m p_k^{(m)}(x), \qquad \overline{p}_k(x) = \max_m p_k^{(m)}(x).$$

- Return the box credal set

$$\mathcal{K}_{\text{box}}(x) = \left\{ p \in \Delta_K : \underline{p}_k(x) \leq p_k \leq \overline{p}_k(x),\ \forall k \right\}.$$



Credal wrapper

Table 3: OOD detection AUROC and AUPRC performance (%) of the classical and credal wrapper version of DEs using EU as the metric. Results are from 15 runs, based on EffB2 and ViT-B backbones. Best scores are in bold.

		CIFAR 10 (ID)				CIFAR 100 (ID)			
		SVHN (OOD)		Tiny-ImageNet (OOD)		SVHN (OOD)		Tiny-ImageNet (OOD)	
		AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
EffB2	Baseline	95.76±0.59	97.43±0.47	92.32±0.14	90.72±0.22	87.52±1.52	93.81±0.80	85.29±0.15	82.98±0.24
	Ours	**97.60±0.25**	**98.74±0.14**	**93.55±0.15**	**93.17±0.16**	**89.20±1.19**	**94.58±0.56**	**86.23±0.10**	**83.69±0.21**
ViT-B	Baseline	77.71±1.67	85.82±1.14	82.27±0.79	78.85±0.81	81.41±1.33	89.00±0.96	81.18±0.31	77.46±0.49
	Ours	**80.71±1.83**	**88.43±1.26**	**84.28±0.76**	**82.82±0.85**	**84.87±1.06**	**91.40±0.67**	**82.53±0.32**	**79.04±0.39**

Figure: Empirical evidence demonstrates credal perspective buys us more.

Can we still be **Bayesian** while being humble to our assumptions?

Example 2: Credal Bayesian deep learning (Caprio *et al.*, 2023)

- Motivation: combat potential **prior and likelihood misspecification**.
- Specify a finite prior set and likelihood set

$$\Pi = \{\pi_1, \ldots, \pi_M\}, \qquad \mathcal{L} = \{\ell_1, \ldots, \ell_J\}.$$

- Perform a **combinatorial Bayes update**:

$$q_{ij}(\theta \mid D) = \frac{\ell_j(D \mid \theta)\pi_i(\theta)}{\int \ell_j(D \mid \vartheta)\pi_i(\vartheta)\, d\vartheta}, \qquad i = 1, \ldots, M, \quad j = 1, \ldots, J.$$

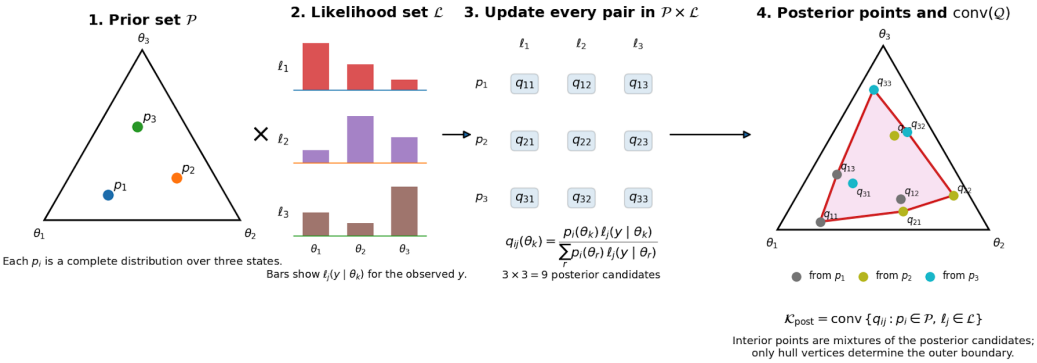
- Form the finitely generated posterior credal set

$$\mathcal{K}_{\text{post}} = \operatorname{conv}\{q_{ij}(\cdot \mid D)\}_{i,j}.$$

- Push the posterior candidates through the predictive map:

$$\mathcal{K}(x) = \operatorname{conv}\left\{\int p(\cdot \mid x, \theta) q_{ij}(d\theta \mid D)\right\}_{i,j}.$$

CBDL cont.



- Contamination prior and posterior sets can also be used for Bayesian sensitivity analysis (Caprio *et al.*, 2026).

Can we have other meaningful **guiding/learning principles** for constructing candidate sets?

Example 3: Relative likelihood (Löhr *et al.*, 2026)

Question: Can we generate probabilistic judges using a **statistical principle**, rather than random initialisation?

- Let $L(h)$ denote the likelihood of predictor $h \in \mathcal{H}$.

$$\gamma(h) = \frac{L(h)}{\sup_{g \in \mathcal{H}} L(g)} \in [0, 1]$$

  measures how plausible a model is relative to the best one.

- Keep only models satisfying

$$\mathcal{C}_\alpha = \{h \in \mathcal{H} : \gamma(h) \geq \alpha\},$$

  called the **relative-likelihood α-cut**.

Instead of treating every trained network equally, retain only models whose likelihood remains sufficiently close to the optimum.

Approximating the α-cut

- In principle, the desired credal prediction is

$$\mathcal{Q}_{x,\alpha} = \{h(x) : h \in \mathcal{C}_\alpha\}.$$

- Unfortunately,

$$\mathcal{C}_\alpha = \{h : \gamma(h) \geq \alpha\}$$

  is a huge subset of a high-dimensional neural-network parameter space.

- Enumerating or optimising over **all** plausible models is computationally infeasible.

How can we approximate $\mathcal{C}_\alpha$ without searching the entire hypothesis space?

Approximating relative-likelihood cuts

- Approximate $\mathcal{C}_\alpha$ by training an ensemble of models.
- Rather than training every network to convergence, assign each member a target **relative likelihood level**

$$\tau_i \in [\alpha, 1],$$

and stop training once

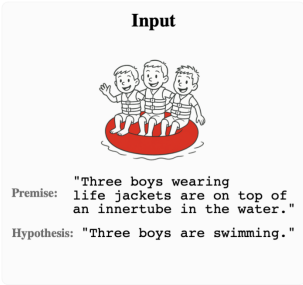
$$\hat{\gamma}(h_i) \approx \tau_i.$$

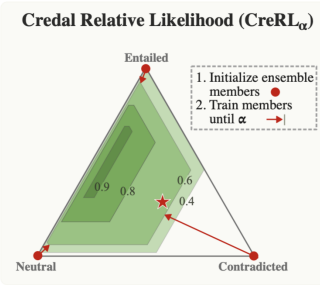
- Their predictions are summarised as lower and upper probability intervals.

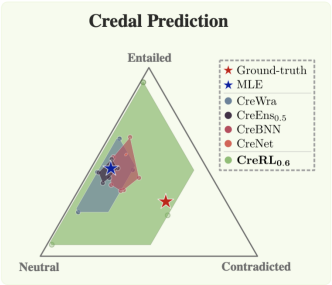
Approximate the set of statistically plausible models using early stopping at different likelihood thresholds.

CreRL Illustration

Example: ChaosNLI, three class classification problem: premise and hypothesis are: entailed, contradicted or neutral.







Training multiple models are **too costly**.. Can we do better?

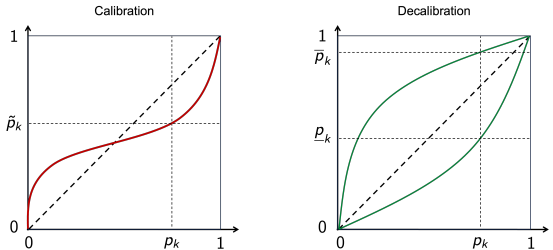
Example 4: Decalibration (Hofman *et al.*, 2025)

Motivation: Can we approximate the plausible-model region without repeatedly training many neural networks?

- Exact relative-likelihood bounds require optimisation over a large neural-network hypothesis space:

$$\inf_{h:\gamma(h)\geq\alpha} h_k(x), \qquad \sup_{h:\gamma(h)\geq\alpha} h_k(x).$$

- Instead, train only the maximum-likelihood predictor $h_{\mathrm{ML}}(x) = \mathrm{softmax}\big(z_{\mathrm{ML}}(x)\big)$ and "decalibrate" the model.



Decalibration (cont.)

- Generate a one-dimensional family of globally decalibrated predictors:

$$\boxed{h_{k,t}(x) = \mathrm{softmax}\big(z_{\mathrm{ML}}(x) + te_k\big), \qquad t \in \mathbb{R}.}$$

  The same class bias t is applied to every input.

- Retain only shifts satisfying

$$\gamma(h_{k,t}) \geq \alpha.$$

  If the feasible shifts form

$$\mathcal{T}_{k,\alpha} = [t_k^-, t_k^+],$$

  then

$$\underline{p}_k(x) = h_{k,t_k^-}(x)_k, \qquad \overline{p}_k(x) = h_{k,t_k^+}(x)_k.$$

One trained network and a collection of one-dimensional searches replace the repeated training of many candidate models.

What does scalability buys us?

- We can now obtain credal sets from Tabular Foundation models having approximately **11 million parameters!**
- Imprecise probabilities at SCALE!

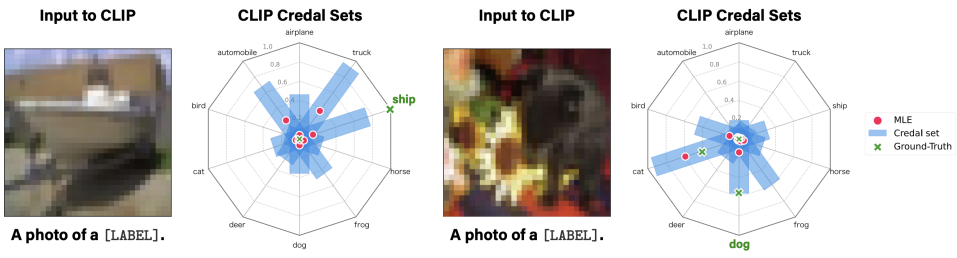


Figure 5: Credal Prediction with `CLIP`. Examples from CIFAR-10H with high epistemic uncertainty (**left**) and high aleatoric uncertainty (**right**) as predicted by applying our method on `CLIP`.

Besides perturbing models, can we perturb **optimisation objectives** to induce meaningful disagreemnts?

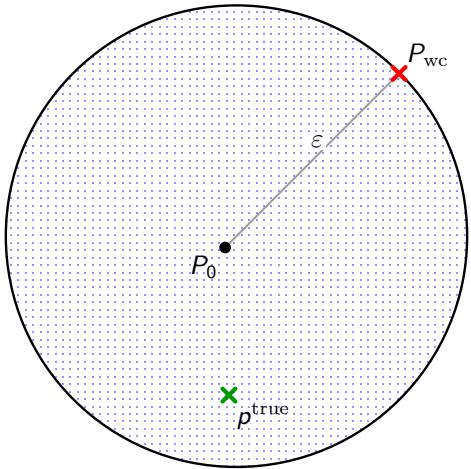
Example 5: Based on Distributionally robust optimisation (DRO)

- **Motivation:** Training typically assumes the training and test data are i.i.d., yet this assumption is often violated in deployment.
- Instead of minimising the empirical risk, distributionally robust optimisation (DRO) seeks a predictor that performs well under a family of plausible test distributions.

$$\min_{\theta} \sup_{P \in \mathcal{U}} \mathbb{E}_{(x,y)\sim P}[L(h_\theta(x), y)]\,,$$

  where $\mathcal{U}$ is an **ambiguity set** describing possible train–test distribution shifts.
- Different choices of $\mathcal{U}$ correspond to different assumptions about how deployment may differ from training.

Distributional ambiguity set



Space containing candidate distributions

Nominal distribution

Worst-case distribution

True underlying uncertainty probability distribution

Example 5: CreDRO (Wang *et al.*, 2026)

- CreDRO instantiates the ambiguity set using the **Conditional Value-at-Risk (CVaR)** formulation.
- The resulting optimisation admits a simple closed-form training objective:

$$\mathrm{CVaR}_\delta = \frac{1}{\delta N} \sum_{n \in S_\delta} L(h_\theta(x_n), y_n),$$

  where S_δ contains the δN samples with the largest losses.
- Training therefore consists of only three steps:

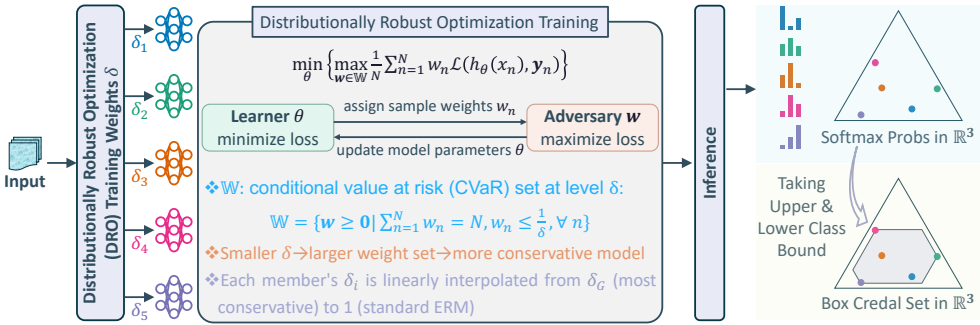
Compute losses $\rightarrow$ Rank $\rightarrow$ Backpropagate on top-δ

CreDRO cont.

- CreDRO assigns different ensemble members different values

$$\delta_i = \delta_G + \frac{i-1}{M-1}(1-\delta_G),$$

so each network represents a different degree of relaxation of the train–test i.i.d. assumption.



CreDRO Results

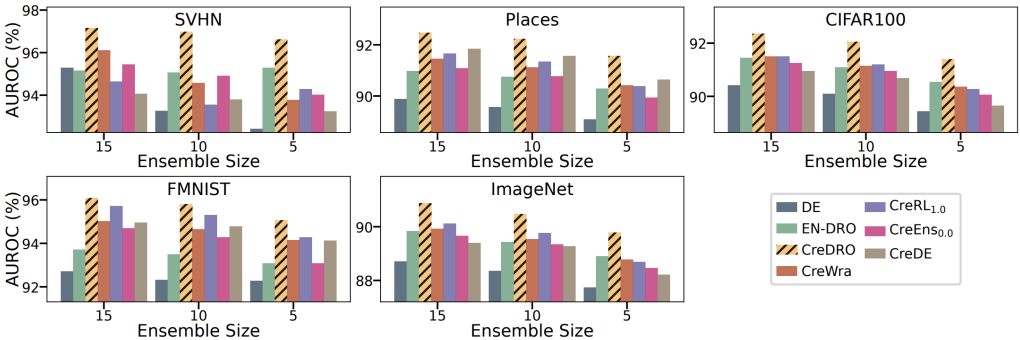


Figure 2. AUROC (%) for OOD detection using EU, across methods and ensemble sizes.

Figure: Empirical evidence suggests objective perturbations buys us more than model perturbations.

Going beyond supervised learning, can we also obtain second-order uncertainty representation from **Language models**?

Black-box uncertainty representations

Black-box methods only require **query access** to the language model.

- **Sampling-based representations**

$$\hat{y}^{(1)}, \ldots, \hat{y}^{(M)} \sim P(\cdot|x),$$

where disagreement between generations represents uncertainty.

- **Semantic representations** Cluster responses according to meaning rather than exact wording:

$$P(\text{semantic answer} \mid x),$$

which motivates measures such as semantic entropy.

- **Verbalised representations** Prompt the model to explicitly express its **belief**:

$$P(\text{answer is correct} \mid x),$$

without requiring access to internal probabilities.

What belief is verbalised probability representing?

$$P(\text{“answer” is correct} \mid x),$$

- belief = *the confidence in a prediction reflecting the probability that the prediction is correct (Guo et al., 2017).*
- A single verbalised probability is a first-order confidence statement about correctness. By itself, it does not identify or separate aleatoric and epistemic uncertainty.
- How to obtain the uncertainty over these verbalized uncertainties?
- Verbalise **IMPRECISE PROBABILITIES!** (Yang *et al.*, 2026)

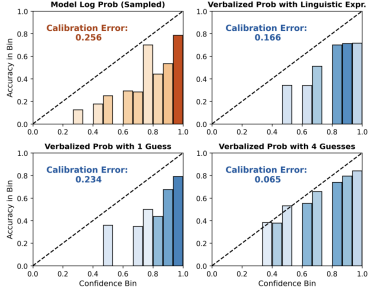


Figure 1: **Verbalized confidence scores (blue) are better-calibrated than log probabilities (orange) for `gpt-3.5-turbo`.** Raw model probabilities **(top-left)** are consistently over-confident. Verbalized numerical probabilities **(bottom)** are better-calibrated. Considering more answer choices **(bottom-right)** further improves verbalized calibration (as in 'Considering the Opposite' in psychology; Lord et al. (1985)). Verbalized *expressions of likelihood* **(top-right)** also provide improved calibration. Bar height is average accuracy of predictions in bin. Darker bars mean more predictions fall in that confidence range. Results computed on SciQ.

De Finetti's Inspired Verbalisation Technique

Assign a **buy price** *(between \$0.00 and \$1.00) for each answer representing the maximum amount you would pay for* **a bet on that answer being correct**. *If an answer is correct, the bet pays \$1.00; if incorrect, it pays \$0.00, and the price paid is lost. Assign prices that maximize your overall profit. The prices must sum to exactly \$1.00 across all answers.*

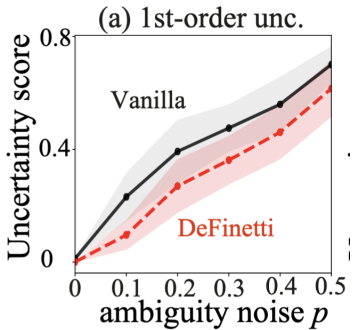


Figure: Correlates nicely with verbalised probability.

Verbalising Imprecise Probabilities from LLMs

Probability intervals

> Provide a lower and upper probability (each between 0.0 and 1.0) indicating how likely the answer is correct. Interpret the probabilities as follows:
>
> - Lower Probability: the smallest probability you consider plausible that the answer is correct.
> - Upper Probability: the largest probability you consider defensible that the answer is correct.

Credal sets

- Repeat De Finetti prompt multiple times with different seeds.

Possibility function

> Provide a possibility score which captures how plausible the answer correctly answers the question. The possibility should be between 0.0 and 1.0.
>
> A possibility score of 1.0 means "fully plausible," and 0.0 means "impossible."

Applications in Q&A ambiguity detection suggests these representations are informative (Yang *et al.*, 2026).

Second-order methods

- (Approximate) Bayesian machine learning
  - BNNs
- Ensemble methods and pseudo-ensemble methods
  - Deep ensembles, Dropout, DropConnect, HyperDM
- Second-order loss minimisation
  - Evidential deep learning (ENN, PriorNN, PostNN, EDD)
- Evidential machine learning
  - DS theory, belief functions
- Direct Epistemic Uncertainty Prediction (DEUP)
  - Train additional model that predicts excess risk (gap to Bayes)
- Credal set prediction
  - Credal wrapper, credal BNN, naïve credal classifier, credal decision trees
- Interval-based methods
  - Interval parameters and/or interval predictions

Takeaway

1. **Machine learning is statistical inference under computational constraints**
   - Uncertainty methods must be theoretically meaningful, but also scalable, optimisable, and usable at prediction time.
2. **Second-order uncertainty is a representation, not a number.**
   - Meta-distributions, ensembles, credal sets, probability intervals, and possibility models retain different information.
3. **The source of disagreement gives the representation its meaning.**
   - Posterior uncertainty, random initialisation, near-optimal likelihood, deployment shift, and prompt variation...
4. **Representation and prediction are separate choices.**
   - The same collection of candidate predictions may be averaged, convexified, bounded by intervals, or otherwise represented.

Agenda

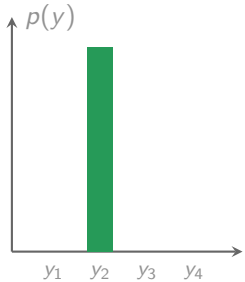
1. Introduction
2. First-order uncertainty representation
3. Second-order uncertainty representation
4. **Uncertainty quantification and evaluation**
   - Quantification
   - Evaluation
5. Summary and outlook

Agenda

1. Introduction
2. First-order uncertainty representation
3. Second-order uncertainty representation
4. **Uncertainty quantification and evaluation**
   - Quantification
   - Evaluation
5. Summary and outlook

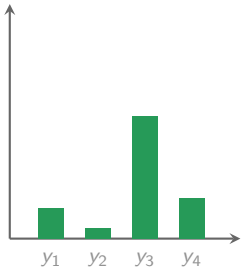
Uncertainty quantification

Shannon entropy: $S(p) = -\sum_i p_i \cdot \log_2(p_i)$



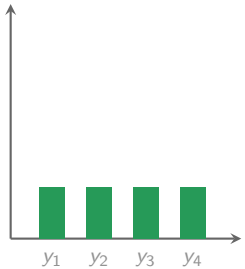
low entropy

$S(p) = 0$



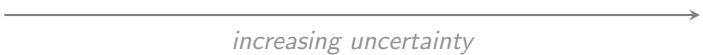
medium entropy

$S(p) \approx 1.5$



high entropy

$S(p) = \log_2 4 = 2$



Shannon entropy: axioms

- Shannon entropy can be justified **axiomatically** (Shannon, 1948; Aczél *et al.*, 1974).
- It is implied (up to a constant multiplicative factor) by the following properties:[1]

  A1 **Continuity**: S is a continuous function

  A2 **Symmetry**: for any permutation τ,

  $$S(p_1, \ldots, p_K) = S(p_{\tau(1)}, \ldots, p_{\tau(K)})$$

  A3 **Additivity**: for all distributions p, p' and their product $p \times p'$,

  $$S(p \times p') = S(p) + S(p')$$

  A4 **Sub-additivity**: for any joint distribution p with marginals p' and p'',

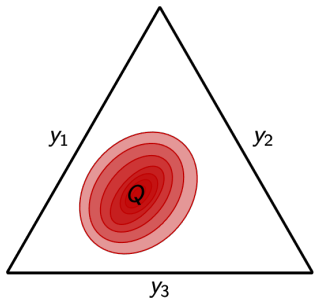
  $$S(p) \leq S(p') + S(p'')$$

[1] More precisely (Aczél *et al.*, 1974): symmetry, expansibility (S unchanged by zero-probability outcomes), additivity, and sub-additivity characterize exactly the combinations $a\, S_{\text{Shannon}} + b\, S_{\text{Hartley}}$ with $a, b \geq 0$; continuity then eliminates the (discontinuous) Hartley part.

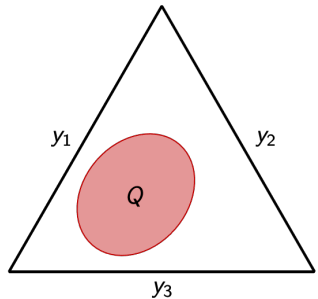
Uncertainty quantification

- **Uncertainty quantification** (UQ) seeks to measure the amount of total, aleatoric, and epistemic uncertainty of a prediction Q in terms of (axiomatically justified) numerical measures, ideally such that

$$\mathrm{TU}(Q) = \mathrm{AU}(Q) + \mathrm{EU}(Q)\,.$$



$\mathrm{TU} = 0.6$, $\mathrm{AU} = 0.2$, $\mathrm{EU} = 0.4$



$\mathrm{TU} = 0.5$, $\mathrm{AU} = 0.2$, $\mathrm{EU} = 0.3$

More axioms (Wimmer *et al.*, 2023; Sale *et al.*, 2024)

A0 TU, AU, and EU are non-negative

A1 $\mathrm{AU}(\delta_{\mathrm{Unif}(\mathcal{Y})}) \geq \mathrm{AU}(\delta_p) \geq \mathrm{AU}(\delta_{\delta_y}) = 0$ holds for any $y \in \mathcal{Y}$ and $p \in \mathbb{P}(\mathcal{Y})$

A2 $\mathrm{EU}(Q) \geq \mathrm{EU}(\delta_p) = 0$ holds for any $Q \in \mathbb{P}(\mathbb{P}(\mathcal{Y}))$ and $p \in \mathbb{P}(\mathcal{Y})$; further, $\mathrm{AU}(Q) = 0$ implies $\mathrm{EU}(Q') \geq \mathrm{EU}(Q)$ for Q' such that $Q'(\delta_y) = \frac{1}{K}$ for all $y \in \mathcal{Y}$

A3 $\mathrm{AU}(Q) \leq \mathrm{TU}(Q)$ and $\mathrm{EU}(Q) \leq \mathrm{TU}(Q)$ holds for any $Q \in \mathbb{P}(\mathbb{P}(\mathcal{Y}))$

A4 $\mathrm{TU}(Q)$ is maximal for Q being the continuous second-order uniform distribution

A5 If Q' is a mean-preserving spread of Q, then $\mathrm{EU}(Q') \geq \mathrm{EU}(Q)$ (weak version) or $\mathrm{EU}(Q') > \mathrm{EU}(Q)$ (strict version)

A6 If Q' is a spread-preserving location shift of Q, then $\mathrm{EU}(Q') = \mathrm{EU}(Q)$

A7 $\mathrm{TU}_{\mathcal{Y}}(Q) \leq \mathrm{TU}_{\mathcal{Y}_1}(Q_{|\mathcal{Y}_1}) + \mathrm{TU}_{\mathcal{Y}_2}(Q_{|\mathcal{Y}_2})$ for $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2$

A8 $\mathrm{TU}_{\mathcal{Y}}(Q_{|\mathcal{Y}_1} \otimes Q_{|\mathcal{Y}_2}) = \mathrm{TU}_{\mathcal{Y}_1}(Q_{|\mathcal{Y}_1}) + \mathrm{TU}_{\mathcal{Y}_2}(Q_{|\mathcal{Y}_2})$, with $\otimes$ the product measure

Uncertainty quantification: information-theoretic

- Common approach (Depeweg *et al.*, 2018) for second-order predictions $Q = H(\boldsymbol{x}) \in \mathbb{P}(\mathbb{P}(\mathcal{Y}))$, treating first-order predictions $p \in \mathbb{P}(\mathcal{Y})$ as random variables distributed according to Q:

$$Y \sim p \sim Q$$

  - TU = **Shannon entropy** of the probabilistic prediction $Y \sim \bar{p}$, where $\bar{p}$ is the predictive distribution (averaged over models):

$$\mathrm{TU} = \mathrm{ENT}(Y) = \mathrm{ENT}(\bar{p}) = \mathrm{ENT}\left(\int p \, dQ(p)\right)$$

  - AU = **conditional entropy** (of prediction given model):

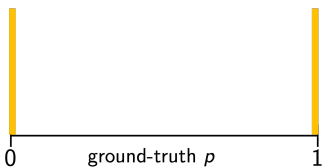
$$\mathrm{AU} = \mathrm{ENT}(Y \mid P) = \int \mathrm{ENT}(p) \, dQ(p)$$

  - EU = **mutual information** $I(Y, P) = \mathrm{ENT}(Y) - \mathrm{ENT}(Y \mid P)$.

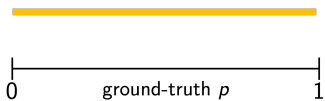
- The decomposition $\mathrm{TU} = \mathrm{AU} + \mathrm{EU}$ holds **by construction**.

Violation of axiomatic properties

- This approach has recently been criticised by Wimmer *et al.* (2023).
- For example, when should EU be highest?



Mutual information is maximal, but EU should (arguably) not



Mutual information is lower, but EU should (arguably) be maximal

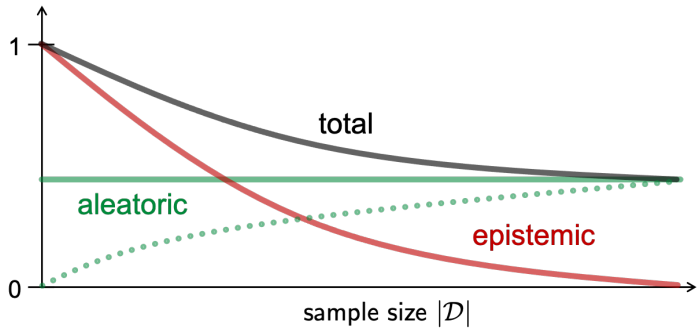
Disentangling aleatoric and epistemic uncertainty

- The **distinction between aleatoric and epistemic uncertainty** can be difficult.
- **Predict the next number**: 116, 304, 194, 341, 224, 654, 609, 625, 533, 91, 205, 35, 527, 611, 128, 235, 348, 912, 582, 52, 672, 20, 856, 904, 628, 273, 615, 105, 610, 862, 384, 705, 73, 794, 775, 156, **??**

$$x \leftarrow x \times 237 \bmod 971$$

- **Epistemic uncertainty implies uncertainty about** the data-generating process, and hence about the (true) **aleatoric uncertainty**.
- Hence precise quantification of AU is not possible unless EU $= 0$ ...

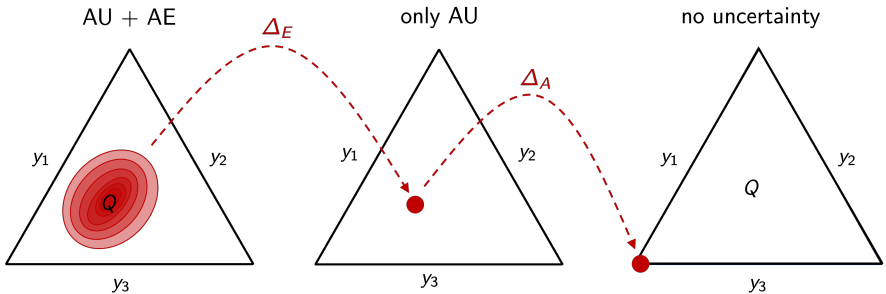
Additive decomposition



Qualitative behaviour of total, aleatoric, and epistemic uncertainty depending on the sample size. Dotted line = total − epistemic uncertainty (= aleatoric uncertainty when stipulating an additive decomposition).

Uncertainty quantification: distance-based

- Sale *et al.* (2024) propose an alternative approach based on the notion of (Wasserstein) **distance**: How much transport is needed to turn a second-order distribution into a distribution representing
  - (i) no epistemic and
  - (ii) no aleatoric uncertainty?



Uncertainty quantification: loss-based

- Another proposal by Hofman *et al.* (2024) is **loss-based**, namely, based on the decomposition of **proper scoring rules** ℓ into a **divergence** (from ground-truth q) and an **entropy** term:

$$L_\ell(p, q) = \mathbb{E}_{Y\sim q}\, \ell(p, Y) = \underbrace{D_\ell(p, q)}_{\text{epistemic}} + \underbrace{H_\ell(q, q)}_{\text{aleatoric}}$$

- For a **second-order distribution** Q, it is sensible to define

$$\begin{aligned}\mathrm{EU}(Q) &= \mathbb{E}_{q\sim Q}[D_\ell(\bar{q}, q)] = \mathbb{E}_{q\sim Q}[L_\ell(\bar{q}, q) - L_\ell(q, q)] \\ &= \underbrace{\mathbb{E}_{q\sim Q}[L_\ell(\bar{q}, q)]}_{\mathrm{TU}(Q)} - \underbrace{\mathbb{E}_{q\sim Q}[L_\ell(q, q)]}_{\mathrm{AU}(Q)} .\end{aligned}$$

- EU is the **gain** (loss reduction) to be expected when predicting, not on the basis of the uncertain Q, but only after being revealed the true q.
- $\ell =$ log-loss recovers entropy-based measures

Uncertainty measures for credal sets: axiomatics

- Basic properties that a measure of uncertainty for credal sets should satisfy:

  A1 **Non-negativity, range**: U is non-negative and upper-bounded by some value $r \in \mathbb{R}$, for example $r = \log(K)$, which is assumed for $Q = \Delta_K$ (the case of complete ignorance).

  A2 **Continuity**: U is a continuous functional.

  A3 **Monotonicity**: If $Q \subseteq Q'$ for credal sets Q, Q', then $U(Q) \leq U(Q')$.

  A4 **Probability consistency**: U reduces to standard Shannon entropy in the case where Q reduces to a single probability distribution.

  A5 **Sub-additivity**: For a (joint) credal set Q on a product space $\mathcal{Y}' \times \mathcal{Y}''$ with marginals Q' resp. Q'',

  $$U(Q) \leq U(Q') + U(Q'')\,. \qquad (\star)$$

  A6 **Additivity**: The inequality $(\star)$ is an equality in the case where Q' and Q'' are independent.

Uncertainty measures for credal sets

- A well-founded generalisation of entropy and natural measure of **total uncertainty** represented by a credal set is the **upper entropy**:

$$S^*(Q) := \max_{q \in Q} S(q)$$

- A well-founded measure of **epistemic uncertainty** is the **generalised Hartley measure**

$$\mathrm{GH}(Q) := \sum_{A \subseteq \mathcal{Y}} \mathrm{m}_Q(A) \log(|A|)\,,$$

  which extends the Hartley measure ($\log(|A|)$) from sets to graded sets.

- Although an equally well-justified measure of **aleatoric uncertainty** (conflict) in the form of an extension of Shannon entropy has not been found so far (Klir, 2005), the **lower entropy** is a natural measure of **irreducible uncertainty**:

$$S_*(Q) := \min_{q \in Q} S(q)$$

Disaggregation

- There is no **additive decomposition**

$$\mathrm{TU}(Q) = \mathrm{AU}(Q) + \mathrm{EU}(Q)$$

  such that all three measures behave well.

- Idea: Fix two "good" measures and **derive** the third one in terms of the **difference**

$$S^*(Q) = \underbrace{\left(S^*(Q) - \mathrm{GH}(Q)\right)}_{\text{generalised entropy}} + \mathrm{GH}(Q)$$

$$S^*(Q) = S_*(Q) + \left(S^*(Q) - S_*(Q)\right)$$

- Empirically, the derived measures show relatively poor performance

Agenda

1. Introduction and first-order uncertainty representation
2. Conformal prediction
3. Second-order uncertainty representation
4. **Uncertainty quantification and evaluation**
   - Quantification
   - **Evaluation**
5. Summary and outlook

Evaluation of uncertainty quantification

- Evaluation of uncertainty representation/quantification is difficult, mainly due to a **lack of a ground truth**.
- Formally, uncertainty measures can be justified in terms of suitable **axioms**.
- Empirically, one often considers how they help improve the performance of a machine learning pipeline in **downstream tasks**:
  - Selective classification
  - Out-of-distribution (OoD) detection
  - Active learning

Selective prediction

- In **selective prediction**, the learner can abstain from making a prediction on some inputs of its choice
- Consider a test set $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, with Q_i the prediction for $\boldsymbol{x}_i$
- For $\alpha \in [0, 1]$, let $k = \lfloor \alpha N \rfloor$ be a (fixed) **rejection level**
- Given measure U, define the permutation π of $\{1, \ldots, N\}$ through

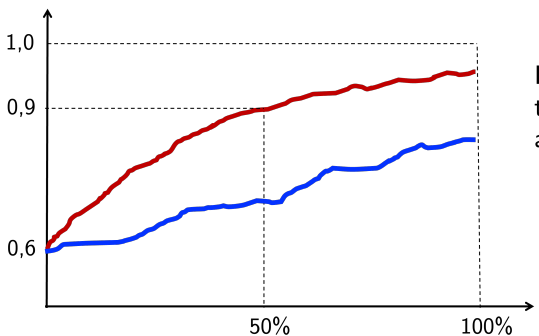
$$U(Q_{\pi(1)}) \leq U(Q_{\pi(2)}) \leq \cdots \leq U(Q_{\pi(N)})$$

- The **accuracy-rejection curve** is then defined as

$$\alpha \mapsto \frac{1}{\lfloor (1-\alpha)N \rfloor} \sum_{j=1}^{\lfloor (1-\alpha)N \rfloor} \text{acc}(q_{\pi(j)}, y_{\pi(j)})$$

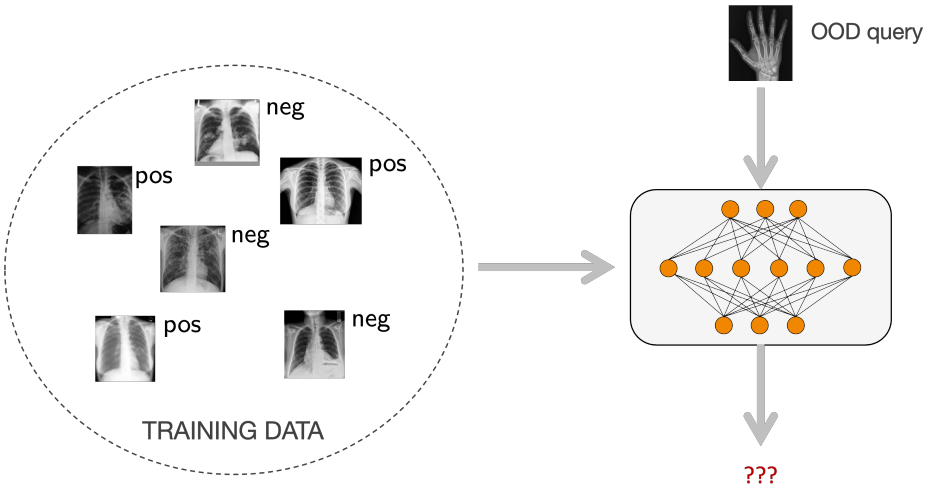
Empirical evaluation

- **Accuracy-rejection curves**: Allow the learner to reject the α % presumably most uncertain test cases and measure accuracy only on the remaining ones



If allowed to reject 50% of the cases, the red learner manages to increase accuracy from 0,6 to 0,9

Out-of-distribution data: "I don't know"

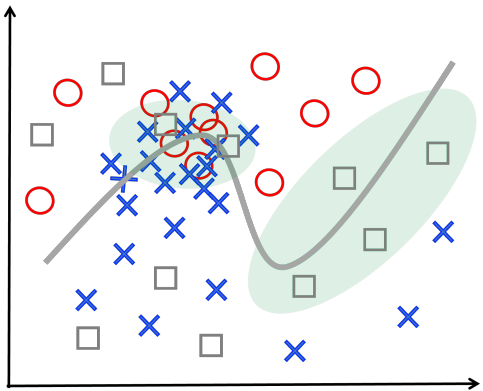


Active learning

- **Active learning** is another popular downstream task
- The goal is to identify examples for which **supervision is most beneficial**, thereby reaching strong performance with as few labels as possible
- Starting with a (small) labeled seed set, the learner iteratively selects unlabeled examples to be annotated by an oracle (and retrains on the augmented data)
- **Epistemic uncertainty** should arguably be a key criterion (Nguyen *et al.*, 2022) — note the connection to bandits (unlike epistemic uncertainty, aleatoric uncertainty cannot be reduced by more supervision)

Active learning

- Principle of **uncertainty sampling**: query those datapoints for which the (current) predictive uncertainty is highest — but which uncertainty?



Summary and outlook

- **Learning reliable predictors** that represent their uncertainty in a faithful way is an important task, but also challenging, both conceptually and computationally
- **Distinguishing different types of uncertainty**, aleatoric and epistemic, is useful, though it seems that second-order uncertainty is hard to tackle
- **Quantifying predictive uncertainty** in a theoretically sound manner, and disentangling total into aleatoric and epistemic uncertainty, is difficult, too
- Various other **open problems**: model uncertainty, generalised settings (eg., OOD data), uncertainty in GenAI and LLMs, evaluation, other forms of uncertainty (e.g., in the data), applications, etc.

References

J. Aczél, B. Forte, and C.T. Ng. Why the Shannon and Hartley entropies are 'natural'. *Advances in Applied Probability*, 6(1):131–146, 1974.

V. Balasubramanian, S.S. Ho, and V. Vovk, editors. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Morgan Kaufmann, 2014.

Michele Caprio, Souradeep Dutta, Kuk Jin Jang, Vivian Lin, Radoslav Ivanov, Oleg Sokolsky, and Insup Lee. Credal bayesian deep learning. *arXiv preprint arXiv:2302.09656*, 2023.

Michele Caprio, Katerina Papagiannouli, Siu Lun Chau, and Sayan Mukherjee. Robust predictive uncertainty and double descent in contaminated bayesian random features. *arXiv preprint arXiv:2602.19126*, 2026.

S. Depeweg, J.M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *Proc. ICML, 35th Int. Conf. on Machine Learning*, Stockholm, Sweden, 2018.

T. Silva Filho, H. Song, M. Perelló-Nieto, R. Santos-Rodríguez, M. Kull, and P.A. Flach. Classifier calibration: How to assess and improve predicted class probabilities: a survey. *Machine Learning*, 112:3211–3260, 2023.

P. Hofman, Y. Sake, and E. H. Quantifying aleatoric and epistemic uncertainty with proper scoring rules. *arXiv preprint arXiv:2404.12215*, 2024.

Paul Hofman, Timo Löhr, Maximilian Muschalik, Yusuf Sale, and Eyke Hüllermeier. Efficient credal prediction through decalibration. 2025.

G.J. Klir. *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley, 2005.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. NeurIPS, 31st Conf. on Neural Information Processing Systems*, Long Beach, California, USA, 2017.

Timo Löhr, Paul Hofman, Felix Mohr, and Eyke Hüllermeier. Credal prediction based on relative likelihood. *Advances in Neural Information Processing Systems*, 38:145184–145213, 2026.

V.L. Nguyen, M.H. Shaker, and E. H. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122, 2022.

Y. Sale, V. Bengs, M. Caprio, and E. Hüllermeier. Second-order uncertainty quantification: A distance-based approach. In *ICML*, 2024.

C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

O. Shorinwa, Z. Mei, J. Lidard, A.Z. Ren, and A. Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges and future directions, 2024.

Kaizheng Wang, Fabio Cuzzolin, Keivan Shariatmadar, David Moens, and Hans Hallez. Credal wrapper of model averaging for uncertainty estimation in classification. In *International Conference on Learning Representations*, volume 2025, pages 70290–70318, 2025.

Kaizheng Wang, Ghifari Adam Faza, Fabio Cuzzolin, Siu Lun Chau, David Moens, and Hans Hallez. Learning credal ensembles via distributionally robust optimization. *arXiv preprint arXiv:2602.08470*, 2026.

L. Wimmer, Y. Sale, P. Hofmann, B. Bischl, and E. H. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *UAI*, 2023.

Anita Yang, Krikamol Muandet, Michele Caprio, Siu Lun Chau, and Masaki Adachi. Verbalizing llm's higher-order uncertainty via imprecise probabilities. *arXiv preprint arXiv:2603.10396*, 2026.