CONFORMAL PREDICTION FOR RELIABLE DECISION-MAKING IN AI

FARHAD POURKAMALI

FARHAD.POURKAMALI@UCDENVER.EDU

DATA SCIENCE AND ARTIFICIAL INTELLIGENCE (DSAI) SYMPOSIUM

UNIVERSITY OF COLORADO DENVER

NOVEMBER 1, 2024

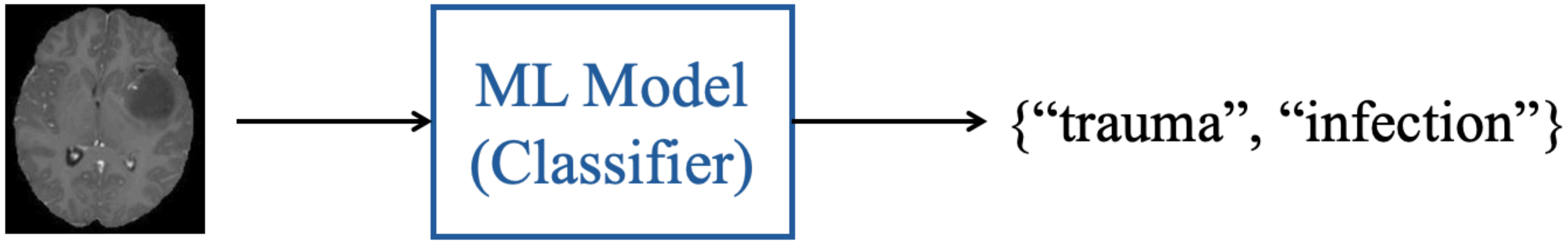
Code/slides on GitHub

WHY UNCERTAINTY MATTERS IN CLASSIFICATION

- While classifiers identify the most likely outcome, there is always some degree of uncertainty in their predictions
- Ignoring model uncertainty is risky, especially with many classes or high-stakes decisions
- Data uncertainty hides in many forms: noise, subjectivity, and ambiguity

CONFORMAL PREDICTION: KEY ADVANTAGES

- Set your own acceptable error rate for a set of plausible outcomes
- Obtain valid predictions regardless of the data distribution
- Make decisions with confidence, not just "best guesses"
- Seamlessly integrate with any underlying model



Prob(true diagnosis ∈ {“trauma”, “infection”})≥ 0.90

PROBLEM FORMULATION

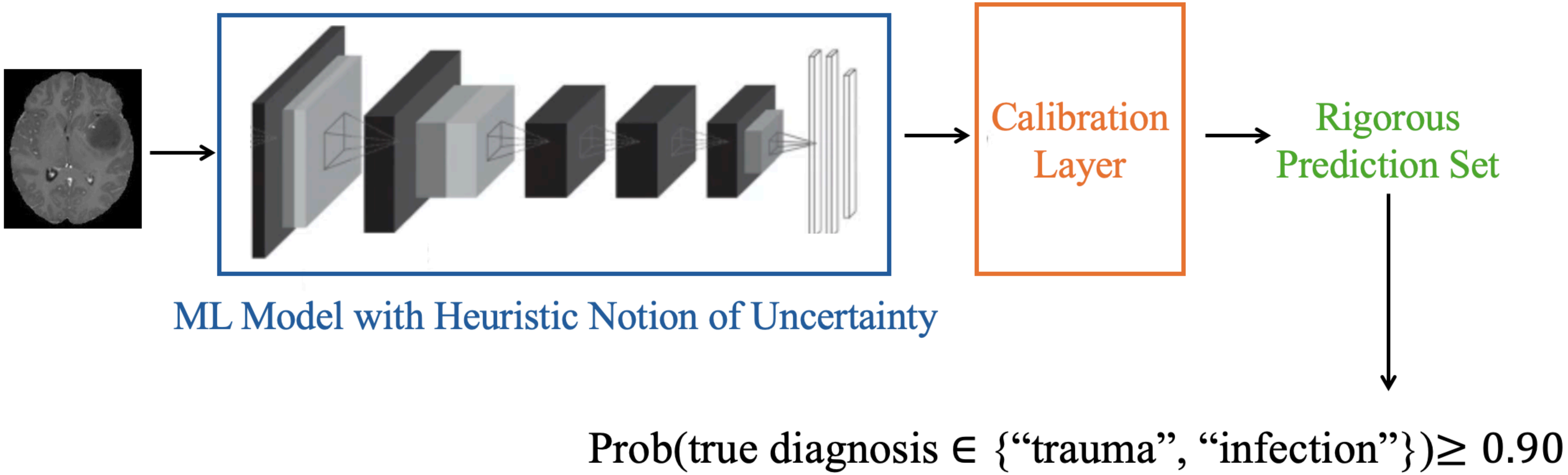
- Trained classifier $\hat{f}$ gives us estimated probabilities for each possible class, i.e., for any x, we have $\hat{f}(x) \in [0,1]^K$
- Calibration data set $(x_1, y_1), \ldots, (x_n, y_n)$ and test data point $(x_{n+1}, ?)$
- Goal: using $\hat{f}$ and calibration data, construct a prediction set $C(x_{n+1}) \subset \{1,\ldots,K\}$ such that

$$\text{Prob}\big(y_{n+1} \in C(x_{n+1})\big) \geq 1 - \alpha \quad \text{(coverage)}$$

- $\alpha \in [0,1]$ is a user-chosen error rate (e.g., $\alpha = 0.1$)

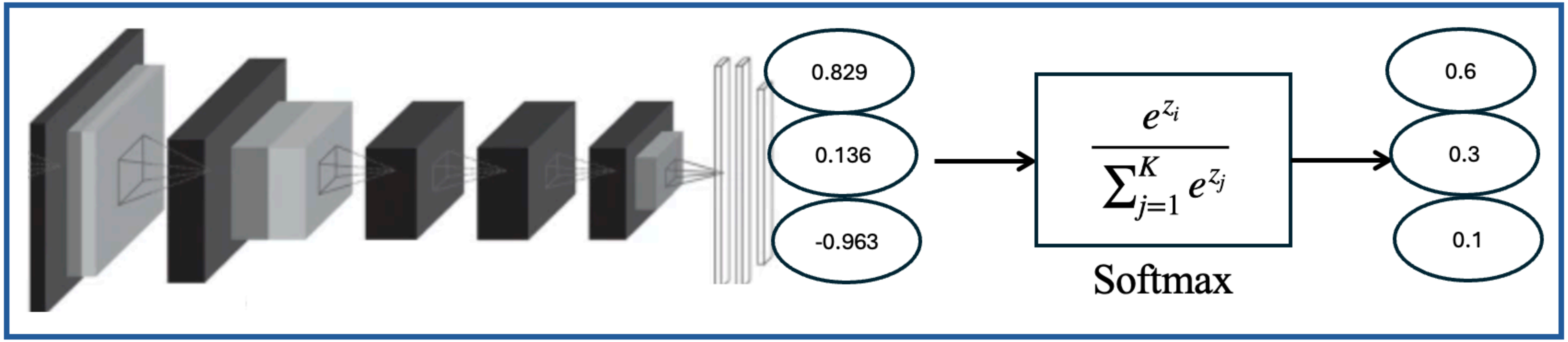
ILLUSTRATING CONFORMAL PREDICTION

- Post-hoc calibration of predictions while leaving the original model architecture and training process untouched



HEURISTIC NOTION OF UNCERTAINTY

- Think of each node in the output layer as giving a "confidence score" for a class (Softmax turns these scores into probabilities)
- Picking top classes is misleading, especially when the model is overly confident
- Calibration is needed: we need a way to adjust these probabilities



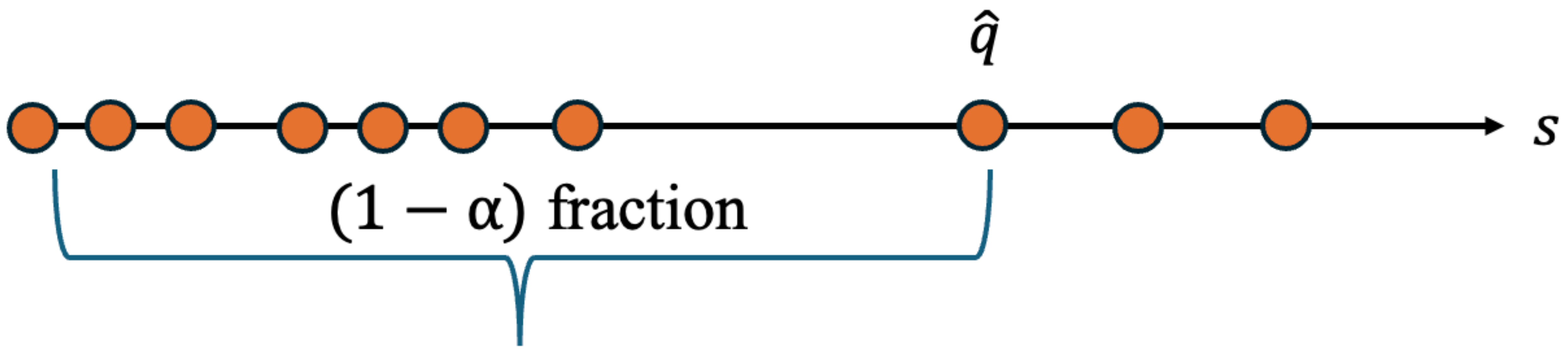
CALIBRATION LAYER

- Calibration data: We start with a set of labeled data called the calibration set $(x_1, y_1), \ldots, (x_n, y_n)$
- Nonconformity score: For each calibration point, we measure the model's confidence using the ground-truth output
  - higher score, less confident
- One way to calculate this score is “1 - softmax probability of the true class”

$$s_i = s(x_i, y_i) = 1 - \hat{f}(x_i)_{y_i}, \; i = 1, \ldots, n$$

CALIBRATION LAYER

- Compute $\hat{q}$ as the $(1-\alpha)$ quantile of the calibration scores $s_1, \ldots, s_n$

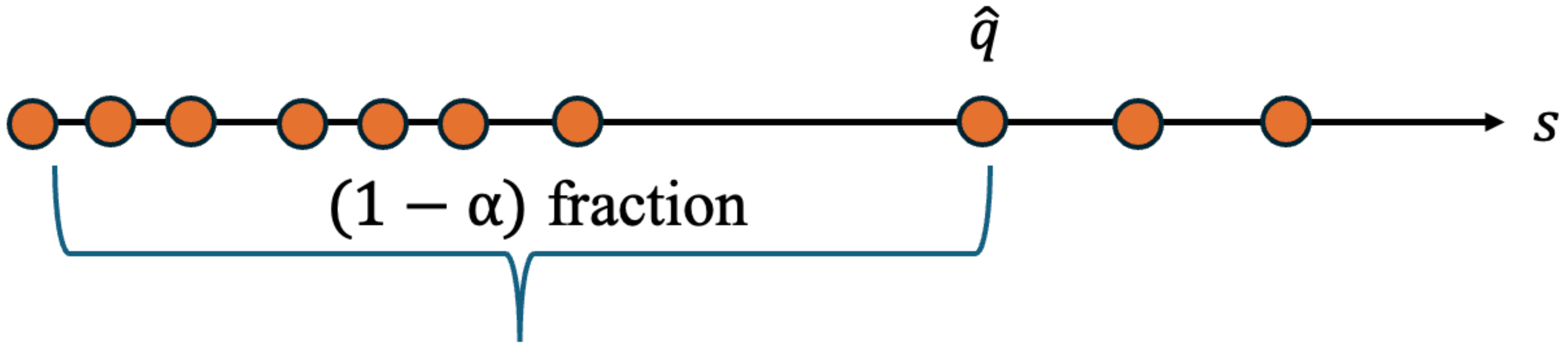


- Use this quantile to form the prediction set
  - Collect all classes with softmax scores above $1-\hat{q}$

$$
\begin{aligned}
C(x_{n+1}) &= \{y : s(x_{n+1}, y) \leq \hat{q}\} \\
&= \{y : 1 - \hat{f}(x_{n+1})_y \leq \hat{q}\} \\
&= \{y : \hat{f}(x_{n+1})_y \geq 1 - \hat{q}\}
\end{aligned}
$$

CALIBRATION LAYER

- Compute $\hat{q}$ as the $(1-\alpha)$ quantile of the calibration scores $s_1, \ldots, s_n$

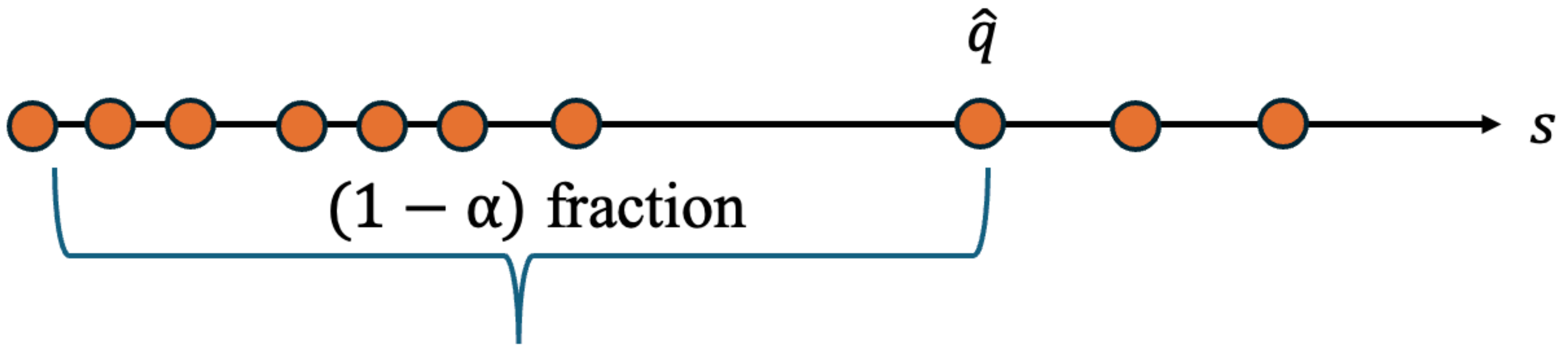


- Use this quantile to form the prediction set
  - Collect all classes with softmax scores above $1-\hat{q}$
  - What is the **fine print?** "the answer can be somewhere between 1 and 1,000,000"

$$
\begin{aligned}
C(x_{n+1}) &= \{y : s(x_{n+1}, y) \leq \hat{q}\} \\
&= \{y : 1 - \hat{f}(x_{n+1})_y \leq \hat{q}\} \\
&= \{y : \hat{f}(x_{n+1})_y \geq 1 - \hat{q}\}
\end{aligned}
$$

WHAT WE NEED

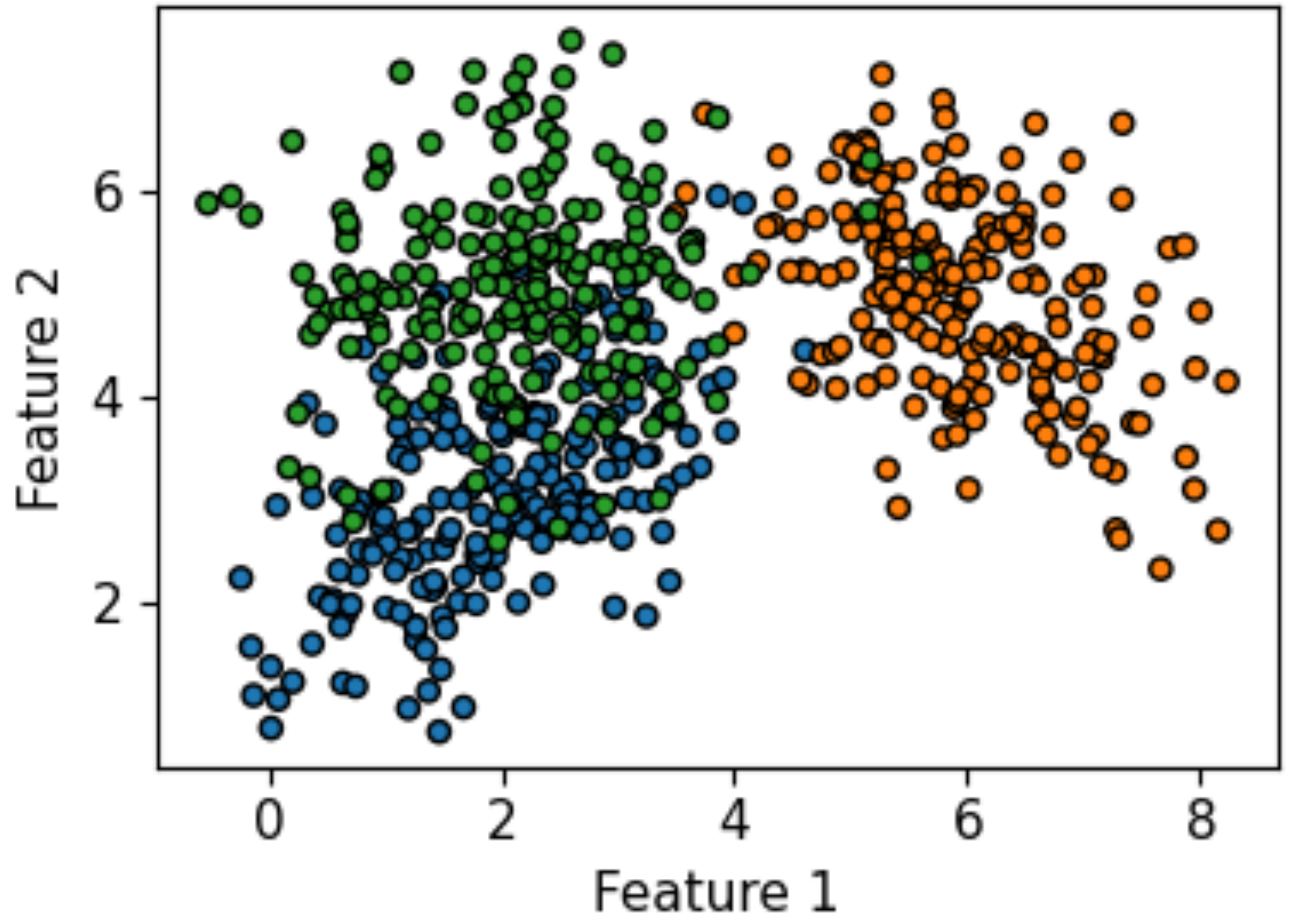
- The scores $s_1, \ldots, s_n, s_{n+1}$ must be exchangeable
  - This means that the order in which we observe these scores doesn't matter

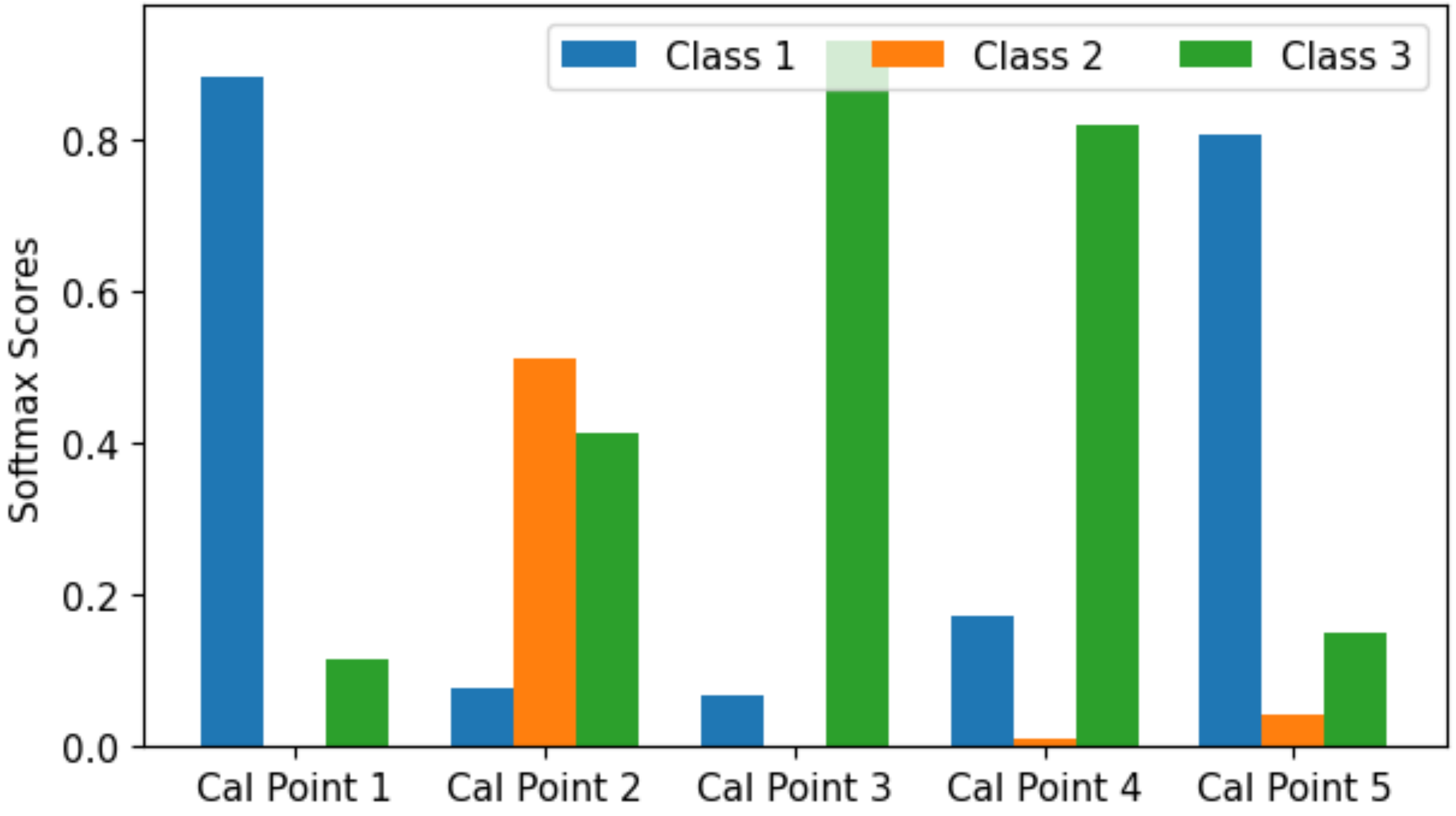


- Formal Definition: $s_1, \ldots, s_n, s_{n+1}$ are exchangeable if for any permutation (reordering) of the indices, the joint probability distribution remains the same

SIMULATED DATA SET WITH THREE CLASSES

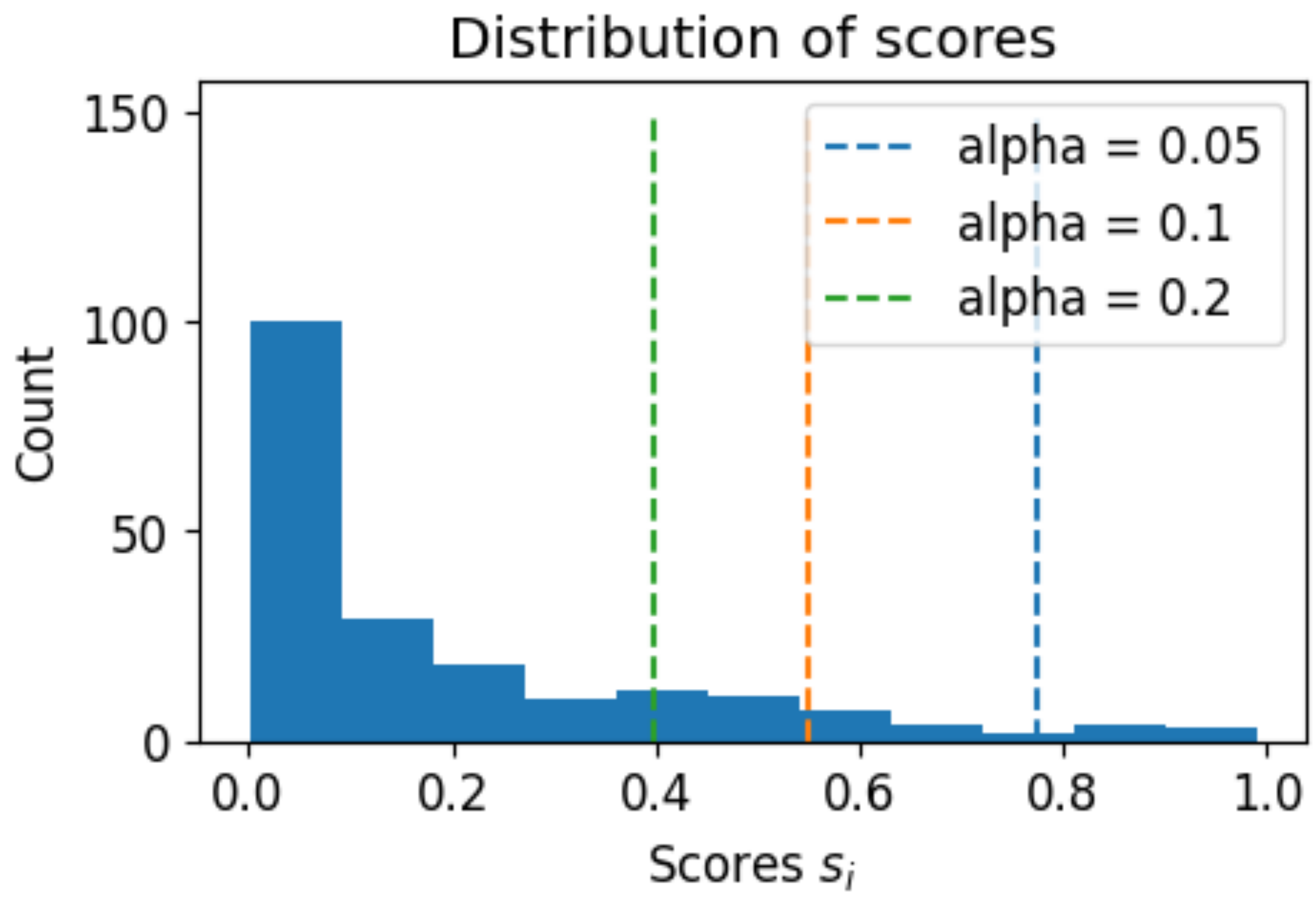
- A data set of 600 data points, encompassing three classes, is split evenly into training, calibration, and test sets
- Logistic regression is used as the base classifier





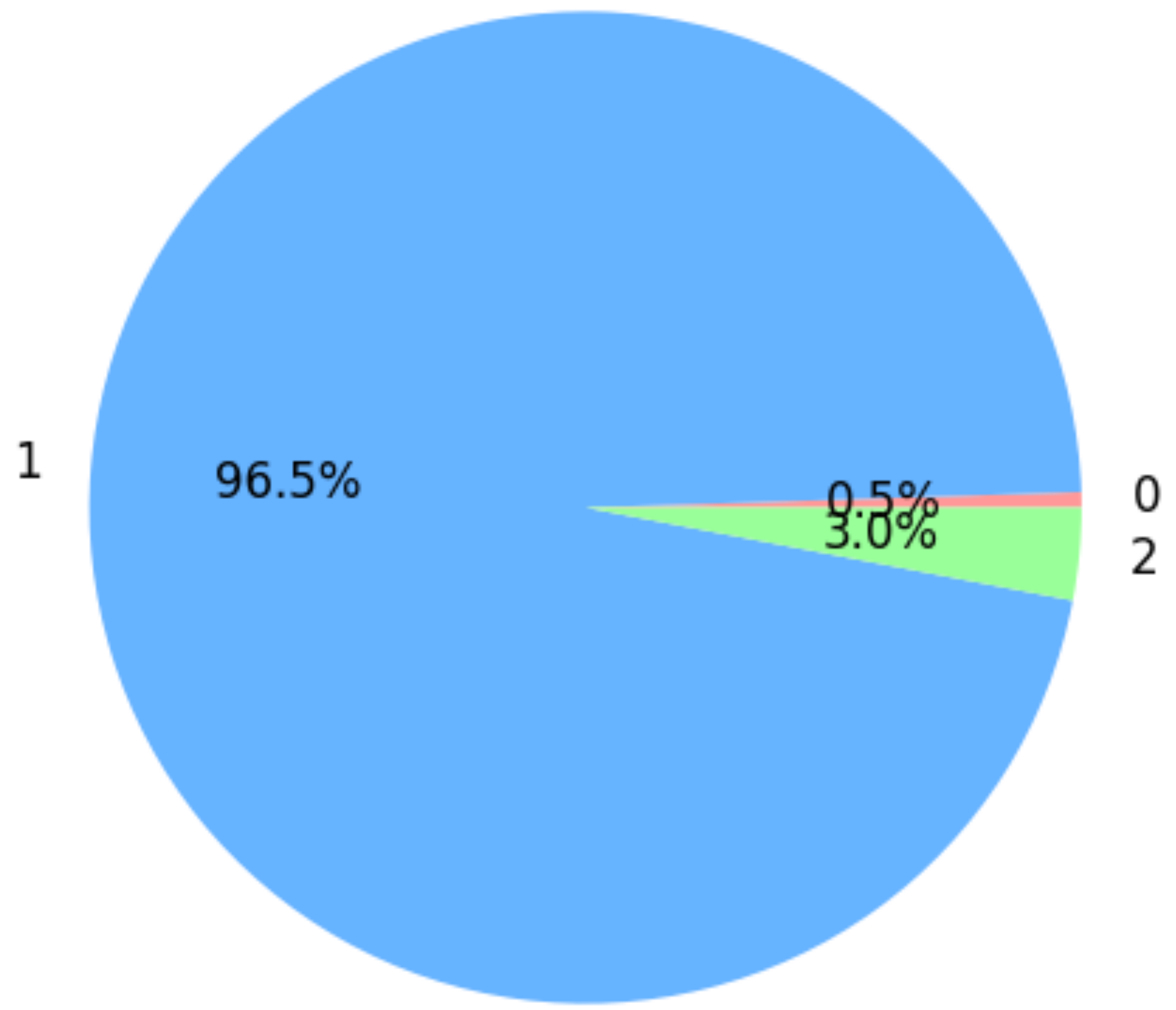
FINDING QUANTILES

- Recall that we have: $s_i = s(x_i, y_i) = 1 - \hat{f}(x_i)_{y_i},\ i = 1,\ldots,n$
- Plot the distribution of these scores and find the $(1-\alpha)$ quantile
- We choose $\alpha = 0.1$



PREDICTION SET FOR TESTING DATA

- The size of the prediction set indicates the model's uncertainty
    - We use a pie chart to assign different colors to each unique category in the prediction set size distribution
    - Conformal prediction abstains when confidence is low

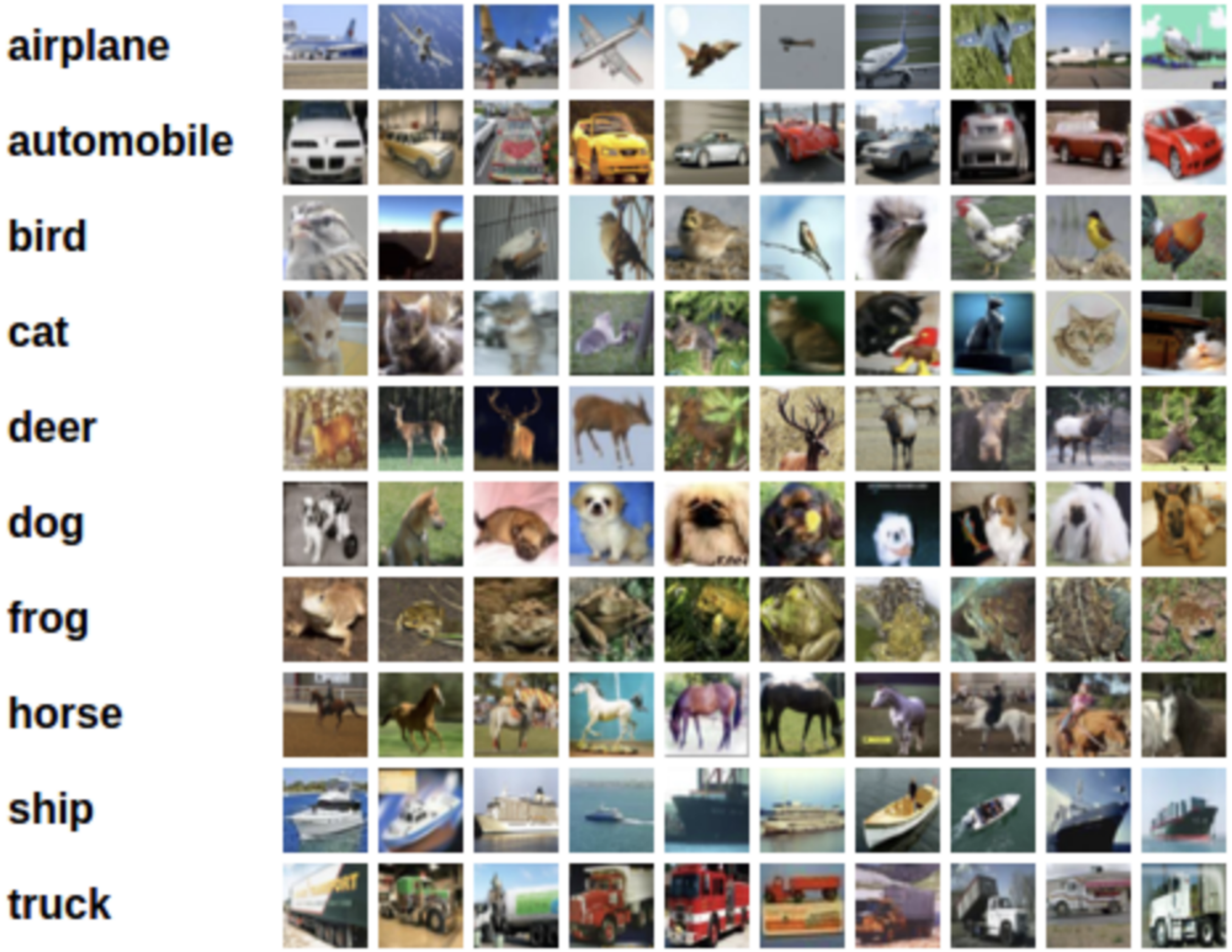


- What is the coverage level? 0.91

IMAGE CLASSIFICATION WITH PRETRAINED MODELS IN PYTORCH

- CIFAR-10: 60,000 color images (32x32 pixels) divided into 10 classes
- PyTorch offers a tutorial on building a CNN for this data
- We can thus verify the integration of conformal prediction without any internal changes

PyTorch



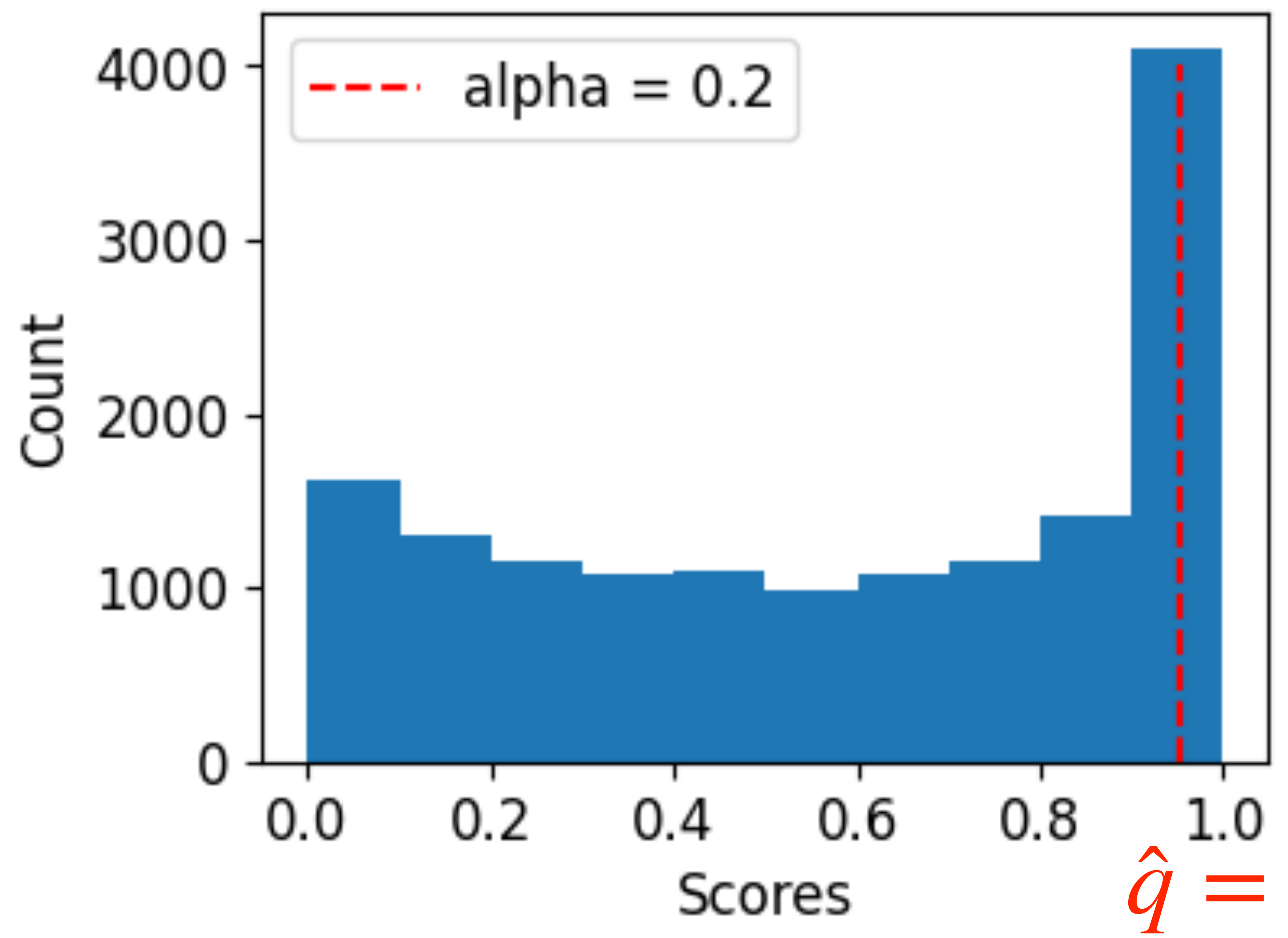
PYTORCH MODEL AND CALIBRATION

```
import torch.nn as nn
import torch.nn.functional as F

class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5,
120)    self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = torch.flatten(x, 1)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
```

→ $\hat{f}(x) \in [0,1]^{10}$ → nonconformity scores for calibration data (15,000 data points)



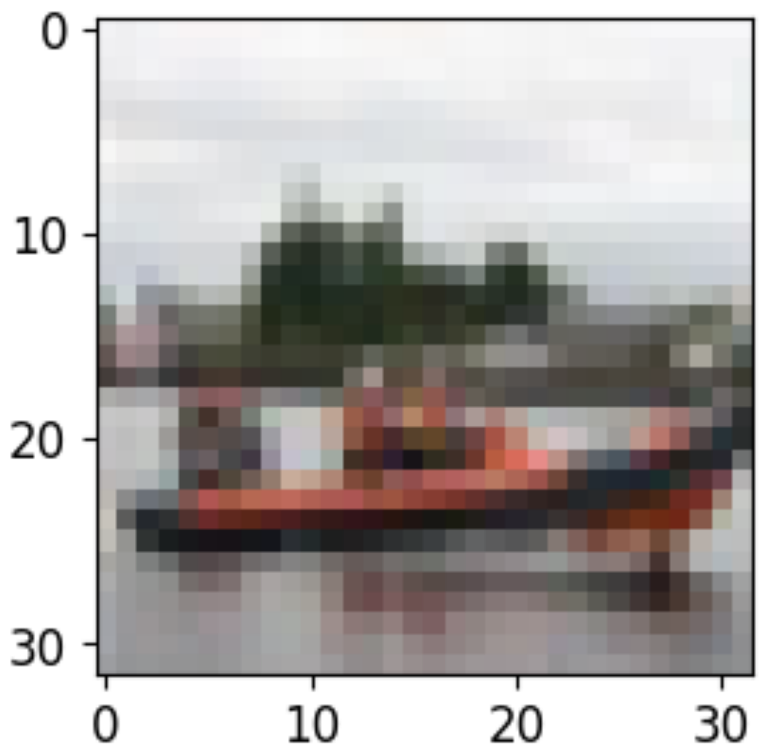
$\hat{q} = 0.95$

PREDICTION SETS FOR TEST IMAGES

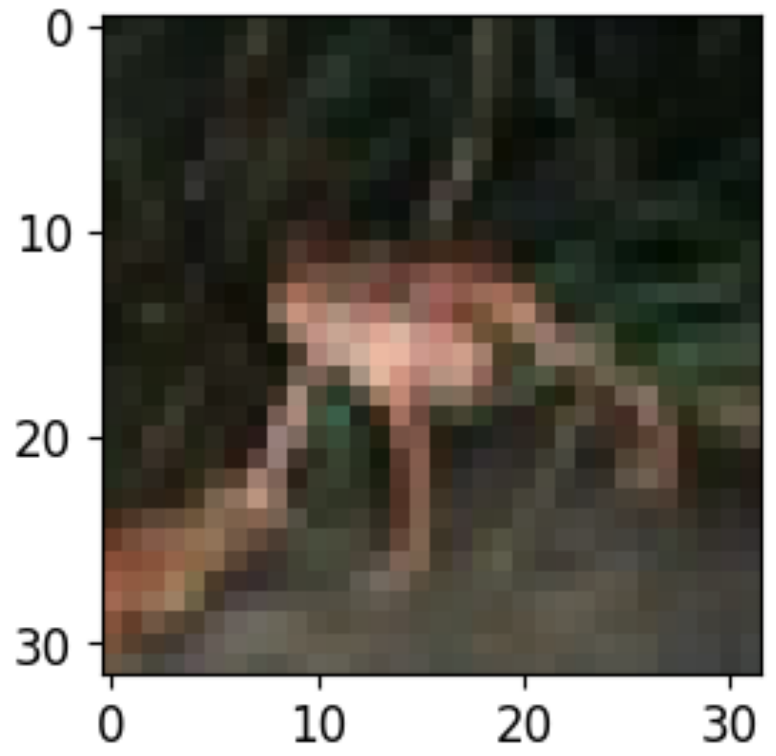
- We construct the prediction set as follows

$$C(x_{n+1}) = \{y : \hat{f}(x_{n+1})_y \geq 1 - \hat{q}\}$$

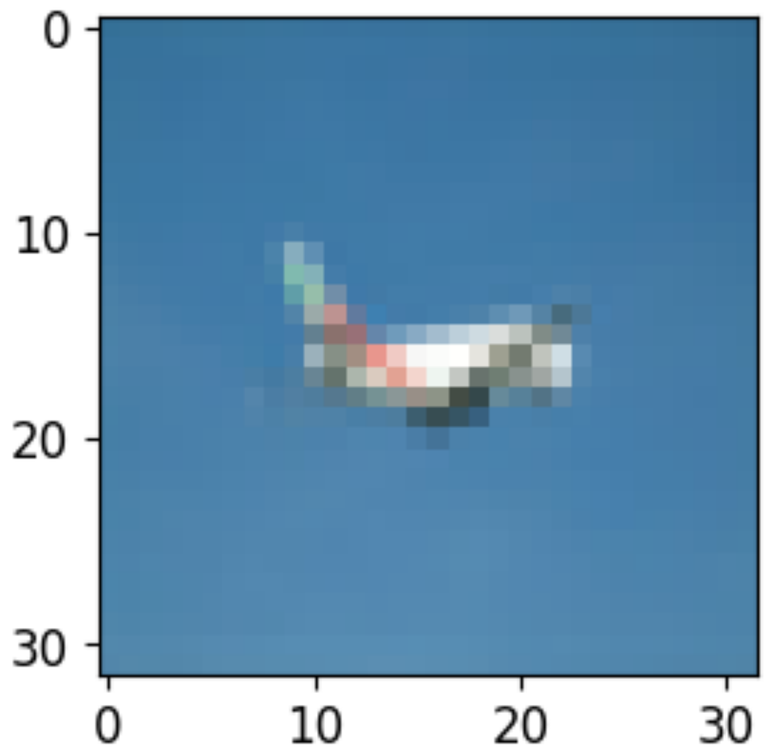
- Four test images and their prediction sets



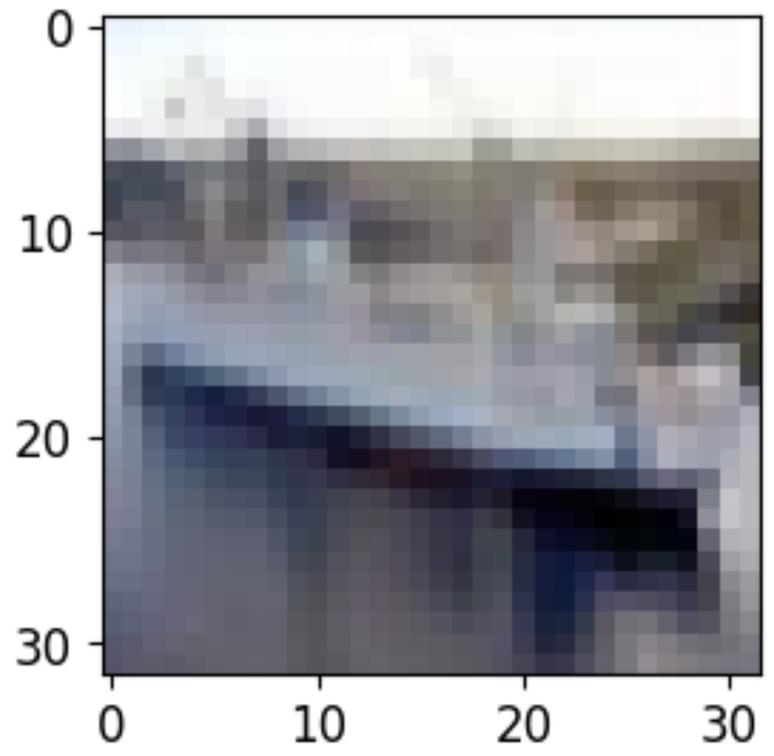
{plane, ship, truck}



{frog, deer}



{bird, plane}



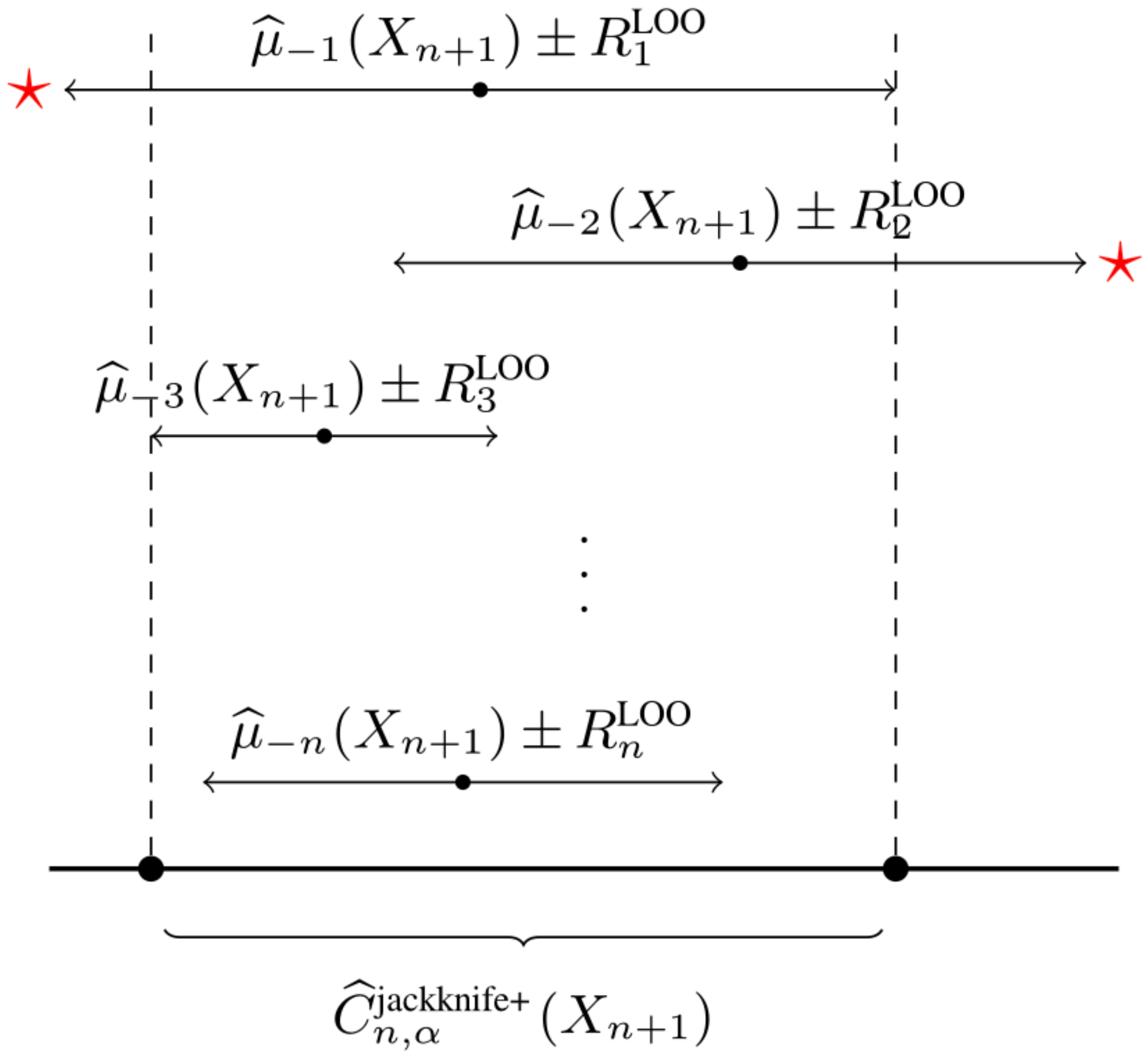
{ship}

CONFORMAL PREDICTION FOR REGRESSION PROBLEMS

- We can use the leave-one-out (LOO) residual as the heuristic notion of uncertainty (higher score, less confident)

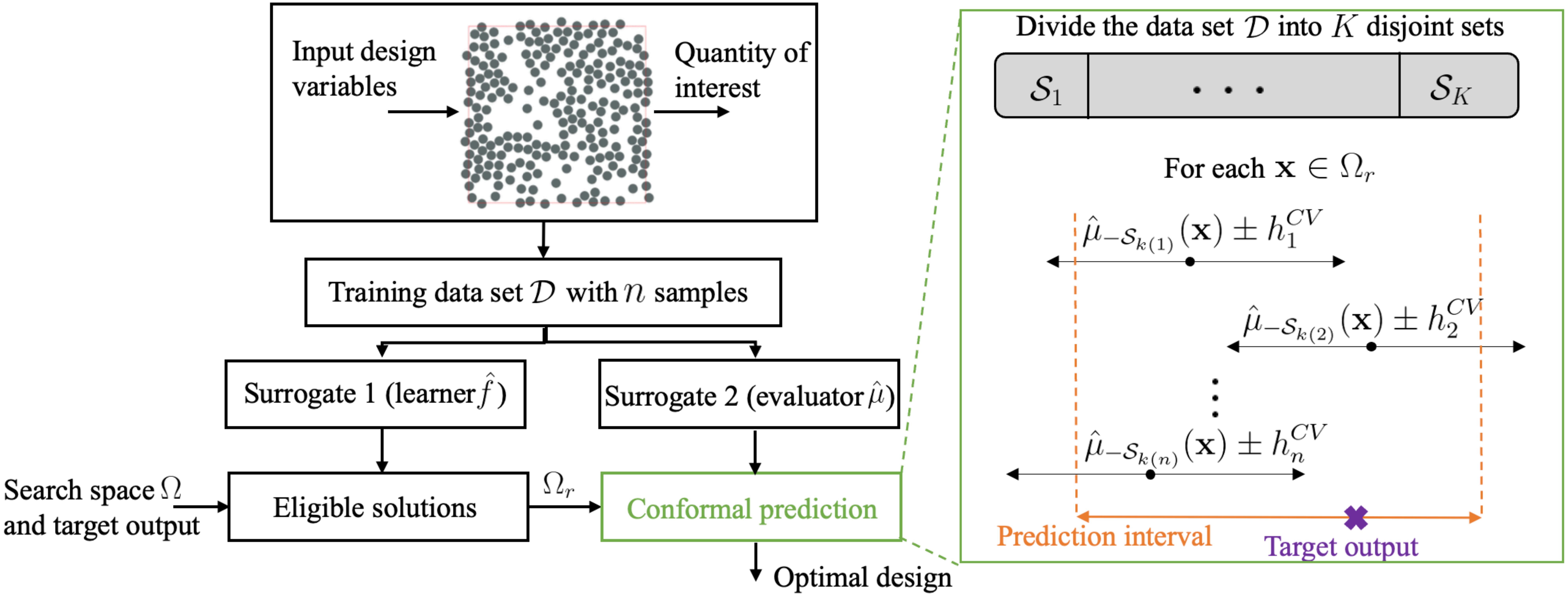
$$s_i = s(x_i, y_i) = R_i^{LOO} = |\, y_i - \hat{\mu}_{-i}(x_i) \,|\, , i = 1, \ldots, n$$

- Barber, R. F., Candes, E. J., Ramdas, A., & Tibshirani, R. J. (2021). Predictive inference with the jackknife+. The Annals of Statistics, 49(1): 486-507



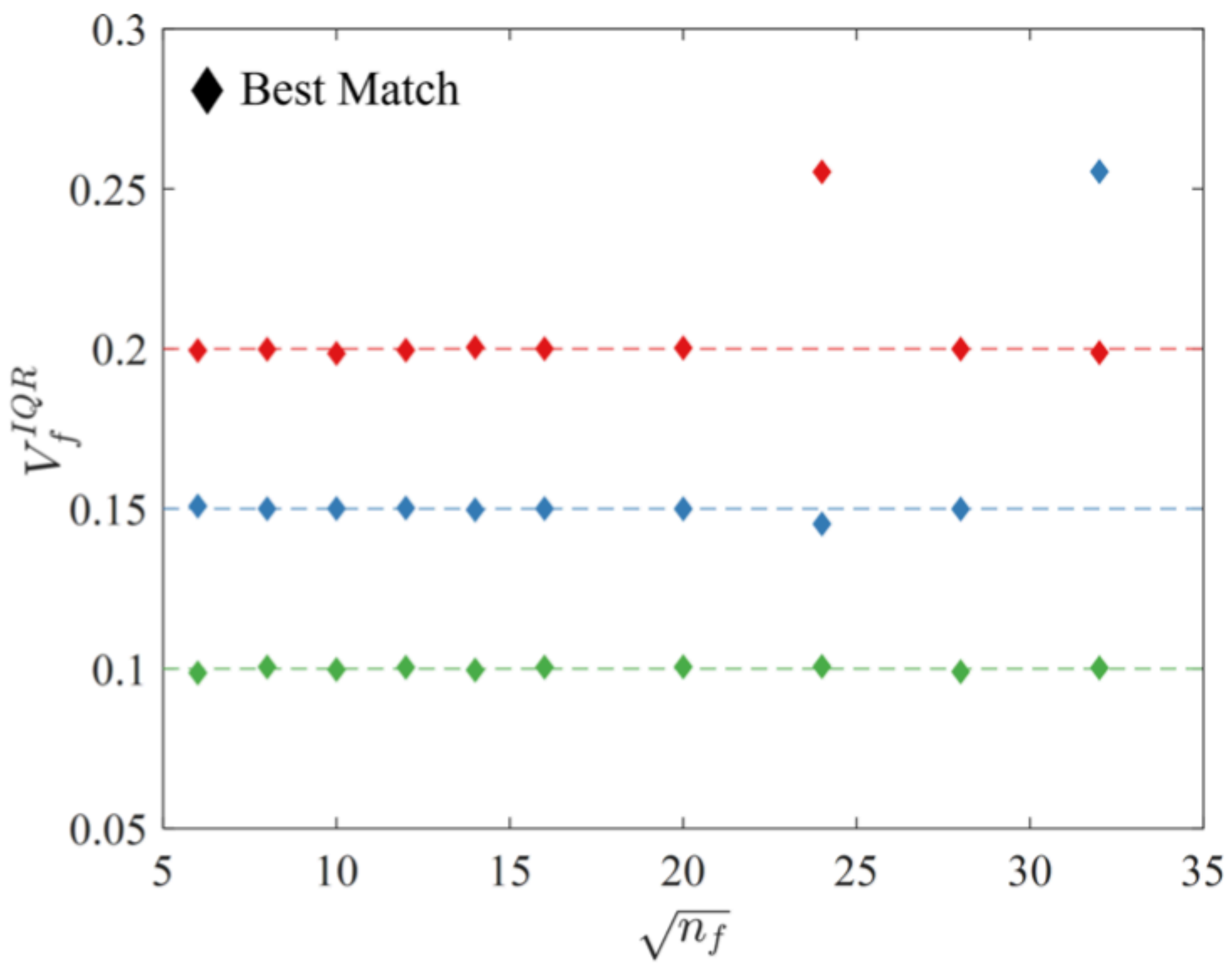
CONFORMAL PREDICTION FOR INVERSE PROBLEMS

- You have a desired output and you want to find the input features that will produce that output
  - Material science: Determine the optimal fiber arrangement

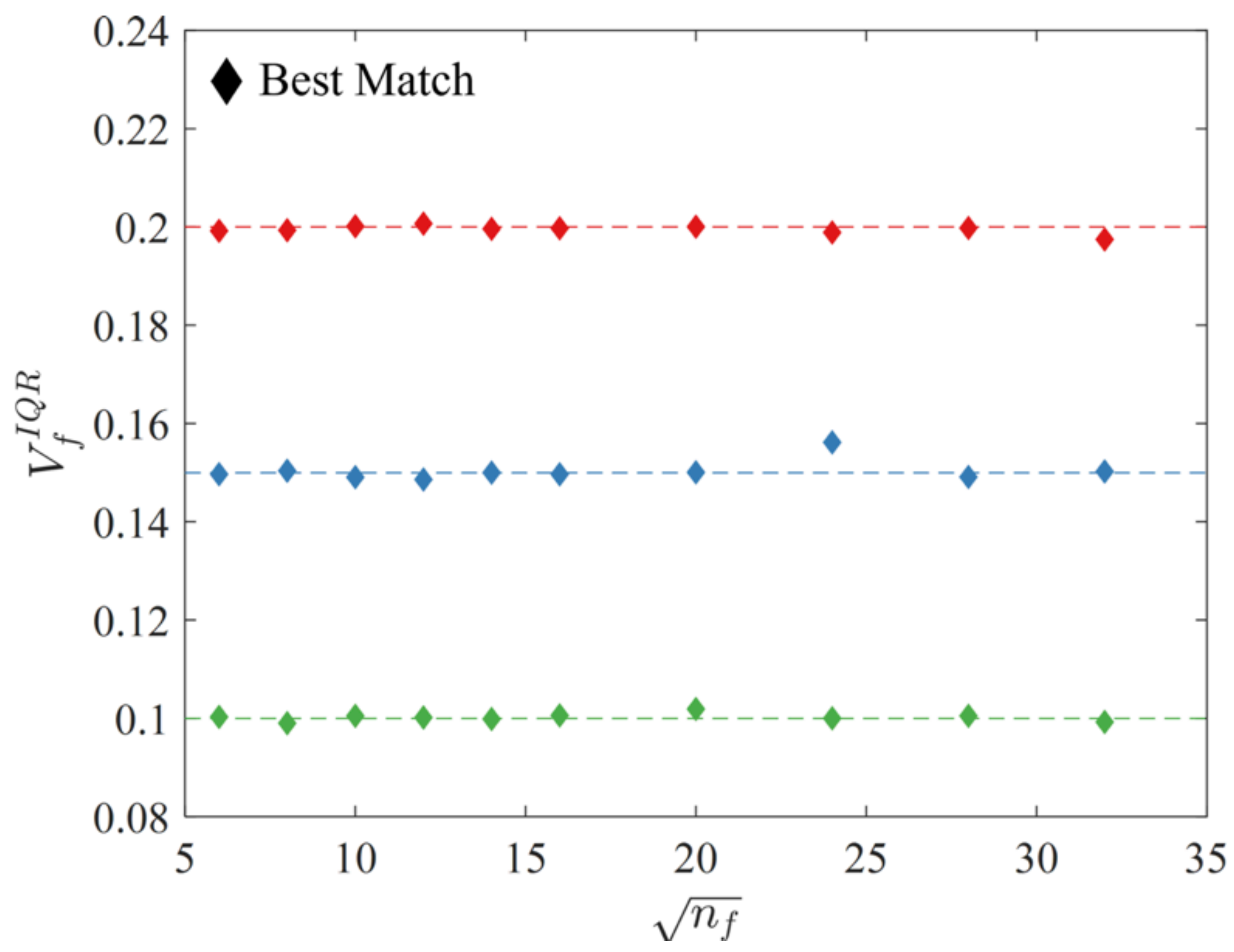


CONFORMAL PREDICTION FOR INVERSE PROBLEMS

- Pourkamali, F., Husseini, J. F., Pineda, E. J., Bednarcyk, B. A., & Stapleton, S. E. (2024). Two-Stage Surrogate Modeling for Data-Driven Design Optimization with Application to Composite Microstructure Generation. Engineering Applications of Artificial Intelligence.



Conformal prediction



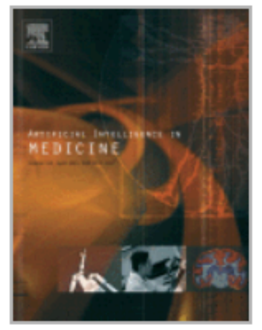
CONFORMAL PREDICTION IN ACADEMIA

- Angelopoulos, A. N., & Bates, S. (2023). Conformal prediction: A gentle introduction. Foundations and Trends in Machine Learning, 16(4), 494-591
- Barber, R. F., Candes, E. J., Ramdas, A., & Tibshirani, R. J. (2023). Conformal prediction beyond exchangeability. The Annals of Statistics, 51(2), 816-845
- Applications:
  - Medical image analysis
  - Time series prediction
  - Robot Learning
  - Natural Language Processing

ELSEVIER

Artificial Intelligence in Medicine

Volume 150, April 2024, 102830



Trustworthy clinical AI solutions: A unified review of uncertainty quantification in Deep Learning models for medical image analysis

Benjamin Lambert [a c], Florence Forbes [b], Senan Doyle [c], Harmonie Dehaene [c], Michel Dojat [a]

CONFORMAL PREDICTION BEYOND ACADEMIA

- MAPIE

MAPIE - Model Agnostic Prediction Interval Estimator

MAPIE is an open-source Python library for quantifying uncertainties and controlling the risks of machine learning models. It is a scikit-learn-contrib project that allows you to:

- Easily **compute conformal prediction intervals** (or prediction sets) with controlled (or guaranteed) marginal coverage rate for regression [3,4,8], classification (binary and multi-class) [5-7] and time series [9].
- Easily **control risks** of more complex tasks such as multi-label classification, semantic segmentation in computer vision (probabilistic guarantees on recall, precision, ...) [10-12].
- Easily **wrap any model (scikit-learn, tensorflow, pytorch, ...) with, if needed, a scikit-learn-compatible wrapper** for the purposes just mentioned.

CONFORMAL PREDICTION BEYOND ACADEMIA

- Amazon Web Services - Fortuna

AWS Machine Learning Blog

Introducing Fortuna: A library for uncertainty quantification

by Gianluca Detommaso, Alberto Gasparin, Cedric Archambeau, Michele Donini, Matthias Seeger, and Andrew Gordon Wilson | on 16 DEC 2022 | in Amazon Machine Learning, Artificial Intelligence, Foundational (100) | Permalink | Comments | Share

Proper estimation of predictive uncertainty is fundamental in applications that involve critical decisions. Uncertainty can be used to assess the reliability of model predictions, trigger human intervention, or decide whether a model can be safely deployed in the wild.

We introduce Fortuna, an open-source library for uncertainty quantification. Fortuna provides calibration methods, such as conformal prediction, that can be applied to any trained neural network to obtain calibrated uncertainty estimates. The library further supports a number of Bayesian inference methods that can be applied to deep neural networks written in Flax. The library makes it easy to run benchmarks and will enable practitioners to build robust and reliable AI solutions by taking advantage of advanced uncertainty quantification techniques.