What If Without the Conformal Prediction Method

XIONG Zhixi
Department of Mathematics
Hong Kong University of Science and Technology
Hong Kong, China

Abstract

The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

In scientific, engineering, and everyday decision-making, obtaining a prediction is insufficient without also understanding its reliability through effective uncertainty quantification (UQ). Whether in autonomous driving, medical diagnosis, or financial risk assessment, relying solely on point estimates poses significant risks, as models are always imperfect and the world exhibits inherent noise. This challenge is formalized in classical decision theory, where optimal decisions require not only an expected outcome but a complete characterization of uncertainty. Formally, consider a decision problem where we observe data $x \in \mathcal{X}$ and must choose an action $a \in \mathcal{A}$. The consequence of this action depends on an unknown state $y \in \mathcal{Y}$, and is quantified by a loss function $L : \mathcal{A} \times \mathcal{Y} \to \mathbb{R}$. In the Bayesian decision theory [Berger, 1985], the optimal decision minimizes the *posterior expected loss*:

$$a^*(x) = \arg\min_{a \in \mathcal{A}} \mathbb{E}_{y \sim p(y|x)} \left[L(a, y)\right] = \arg\min_{a \in \mathcal{A}} \int_{\mathcal{Y}} L(a, y)\, p(y|x)\, dy, \tag{1}$$

where $p(y|x)$ is the posterior distribution of y given x. As shown in (1), the optimal action a^* depends not merely on a point estimate (such as the posterior mean), but on the entire posterior distribution. A poor characterization of $p(y|x)$ can lead to suboptimal decisions even with an accurate predictive model. Thus, robust and interpretable decision-making fundamentally requires well-calibrated uncertainty estimates. Consequently, a central challenge in modern machine learning, particularly for complex black-box models like deep neural networks, is to provide rigorous and reliable uncertainty measures for predictions. Without such measures, deploying these models in high-stakes domains remains risky.

To address this challenge, a variety of UQ methods have been developed, each with its own set of limitations that restrict practical applicability. Parametric models, for instance, often rely on strong distributional assumptions (e.g., normality) that are rarely justified in complex, real-world data settings [Wasserman, 2004]. Asymptotic theories, while providing theoretical guarantees under ideal conditions, require sample sizes that are often unattainable in practice, and their convergence may not be guaranteed for finite or moderate datasets [Vovk et al., 2005]. Within the Bayesian paradigm, the choice of a prior distribution can be subjective and difficult to justify. Furthermore, computing the posterior distribution is often intractable for high-dimensional or non-conjugate models, which often necessitates the use of approximate inference schemes that may themselves introduce errors [Murphy, 2012]. Among more recent machine learning approaches, ensemble methods (e.g., deep ensembles [Lakshminarayanan et al., 2017]) can yield well-calibrated uncertainty estimates but at a prohibitive computational cost, as they require training and maintaining multiple models, which is a significant burden for large datasets and complex architectures. Techniques like Monte

Carlo Dropout [Gal and Ghahramani, 2016] offer a more lightweight alternative by approximating Bayesian inference within neural networks, yet they impose specific architectural constraints (e.g., dropout layers) and require careful tuning of key hyperparameters, most notably the dropout rate, as well as the overall training schedule. Lastly, generative models (e.g., diffusion [Ho et al., 2020] or flow matching [Lipman et al., 2022]) can model complex data distributions but are notoriously expensive to train and may be challenging to scale to high-dimensional problems. In summary, existing methods are often constrained by strong assumptions, high computational demands, or a lack of finite-sample guarantees, highlighting the need for a framework that is both model-agnostic and theoretically rigorous.

The conformal prediction (CP) framework emerges as a compelling solution to these limitations, offering a model-agnostic approach to uncertainty quantification with finite-sample, distribution-free guarantees [Gammerman et al., 1998, Vovk et al., 2005]. Its core mechanism relies on the concept of *nonconformity scores*. For any new input, CP evaluates how nonconforming each potential output appears relative to a set of labeled reference (or calibration) data. By calibrating a threshold based on the empirical distribution of these scores, CP constructs a prediction set (for classification) or a prediction interval (for regression) that is guaranteed to contain the true outcome with a user-specified probability (e.g., 90%), under the weak assumption of data exchangeability. This procedure essentially uses past empirical errors to quantify future uncertainty, requiring no strong distributional assumptions, no modifications to the underlying model, and no asymptotic approximations.

Originally introduced by Gammerman, Vovk, and Vapnik in 1998 for Support Vector Machines [Hearst et al., 1998], Conformal Prediction has transcended its initial scope to become a pervasive framework in modern reliable AI. While theoretically grounded, its value is most evident in its rapidly expanding real-world impact across diverse high-stakes domains. In the natural sciences, CP is now pivotal for guiding protein design sequences [Fannjiang et al., 2022]. In the realm of robotics and embodied AI, it facilitates safe planning in dynamic environments [Lindemann et al., 2023] and has been successfully applied to align uncertainty in Large Language Models (LLMs) for robotic interaction [Ren et al., 2023]. The framework's reliability is equally critical in healthcare, where it has been deployed for estimating diagnostic uncertainty in cancer pathology [Olsson et al., 2022] and predicting disease courses for conditions such as multiple sclerosis [Sreenivasan et al., 2025]. Additionally, CP has found utility in earth observation [Singh et al., 2024], further demonstrating its versatility and robustness in securing trust for safety-critical applications. Collectively, these advancements illustrate the transformative potential of CP in bridging the gap between complex, black-box algorithms and the rigorous safety standards demanded by real-world applications.

2 Methods

This section formalizes the CP framework. We begin with the general transductive formulation, proceed to the computationally efficient Split Conformal Prediction, and conclude by discussing advanced variants designed to address limitations regarding adaptivity, distribution shifts, and conditional coverage.

2.1 General Conformal Prediction Framework

Consider a regression or classification problem where we observe a sequence of data points $Z_1, \ldots, Z_n$, where $Z_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$. We assume these data points are *exchangeable*, meaning their joint distribution is invariant under any permutation. Given a new test input X_{n+1}, our goal is to construct a prediction set $C(X_{n+1}) \subseteq \mathcal{Y}$ that covers the unknown true label Y_{n+1} with a user-specified probability $1 - \alpha$, where $\alpha \in (0, 1)$ is the miscoverage rate.

The general conformal prediction framework, often referred to as Full or Transductive CP [Vovk et al., 2005], relies on a *nonconformity measure* $S : \mathcal{Z} \to \mathbb{R}$. This function assigns a score $s_i = S(Z_i)$ representing how "unusual" a data point Z_i is relative to the others. For a candidate label $y \in \mathcal{Y}$ paired with X_{n+1}, we form a hypothetical dataset including $Z_{n+1} = (X_{n+1}, y)$. We then compute nonconformity scores for all $i \in \{1, \ldots, n+1\}$. A p-value for the candidate y is derived by comparing its score to the scores of the existing data:

$$\pi(y) = \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{I}\left(s_i \geq s_{n+1}\right), \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The prediction set is constructed by including all candidates y that appear statistically plausible:

$$C_{\text{Full}}(X_{n+1}) = \{y \in \mathcal{Y} \mid \pi(y) > \alpha\}\,. \tag{3}$$

Under the exchangeability assumption, this set satisfies the marginal validity property:

$$\mathbb{P}(Y_{n+1} \in C_{\text{Full}}(X_{n+1})) \geq 1 - \alpha, \tag{4}$$

for any sample size n and any underlying distribution. The structure of the set $C_{\text{Full}}(X_{n+1})$ depends on the nature of the output space $\mathcal{Y}$:

Classification. When $\mathcal{Y}$ is a finite set of discrete labels (e.g., $\{1, \ldots, K\}$), we can explicitly calculate $\pi(y)$ for each possible class. The prediction set is simply the subset of labels for which the p-value exceeds α.

Regression. When $\mathcal{Y} = \mathbb{R}$, iterating through all possible values of y is computationally impossible. However, for standard nonconformity measures (such as the absolute error $|y - \hat{\mu}(X_{n+1})|$), the function $\pi(y)$ is generally quasi-concave with respect to y. Consequently, the set defined in (3) typically forms a continuous interval (or a union of intervals). In practice, this interval is computed by inverting the nonconformity score function to find the boundaries of y that satisfy the condition $\pi(y) > \alpha$.

2.2 Split Conformal Prediction

While theoretically robust, Full CP is computationally prohibitive for complex models (e.g., neural networks) because it requires retraining the underlying model for every candidate y to compute the scores s_i properly. To address this computational bottleneck, *Split Conformal Prediction* (SCP), also known as Inductive CP, is the most widely adopted variant in modern machine learning [Papadopoulos et al., 2002, Lei et al., 2018].

In SCP, the available data is partitioned into two disjoint subsets: a proper training set $\mathcal{D}_{\text{train}}$ and a calibration set $\mathcal{D}_{\text{cal}} = \{(X_i, Y_i)\}_{i=1}^n$. A predictive model is trained solely on $\mathcal{D}_{\text{train}}$. We then compute nonconformity scores on $\mathcal{D}_{\text{cal}}$ using this fixed model. The construction of the prediction set depends on the task type:

Classification. For a classification task with discrete labels $\mathcal{Y} = \{1, \ldots, K\}$, the model $\hat{f} : \mathcal{X} \to [0, 1]^K$ typically outputs a probability distribution over classes, where $\hat{f}(x)_k$ denotes the estimated probability of class k. A common nonconformity score is the complement of the probability assigned to the true class:

$$s_i = 1 - \hat{f}(X_i)_{Y_i}, \quad \forall i \in \mathcal{D}_{\text{cal}}. \tag{5}$$

Let $\hat{q}$ be the $\lceil (n + 1)(1 - \alpha) \rceil / n$ empirical quantile of the calibration scores $\{s_1, \ldots, s_n\}$. The prediction set includes all classes with a predicted probability above the calibrated threshold:

$$C_{\text{SCP}}(X_{n+1}) = \left\{ k \in \mathcal{Y} \mid \hat{f}(X_{n+1})_k \geq 1 - \hat{q} \right\}. \tag{6}$$

This formulation, often referred to as Least Ambiguous Set-valued Classifiers [Sadinle et al., 2019], guarantees that the true label is included in the set with probability at least $1 - \alpha$.

Regression. For a regression task ($\mathcal{Y} = \mathbb{R}$), let $\hat{\mu} : \mathcal{X} \to \mathbb{R}$ be the trained model. A standard choice for the nonconformity score is the absolute residual:

$$s_i = |Y_i - \hat{\mu}(X_i)|, \quad \forall i \in \mathcal{D}_{\text{cal}}. \tag{7}$$

Similarly, let $\hat{q}$ be the $\lceil (n+1)(1-\alpha) \rceil / n$ empirical quantile of these regression scores. The prediction set for a new input X_{n+1} is constructed as a fixed-width interval:

$$C_{\text{SCP}}(X_{n+1}) = [\hat{\mu}(X_{n+1}) - \hat{q}, \quad \hat{\mu}(X_{n+1}) + \hat{q}]\,. \tag{8}$$

In both cases, SCP maintains the finite-sample validity guarantee provided the calibration and test data are exchangeable. Because the model is trained only once, SCP is computationally efficient and model-agnostic.

2.3 Advanced Variants addressing Limitations

While SCP provides rigorous marginal coverage, the formulation in (8) constructs intervals of fixed width $2\hat{q}$ for all inputs. This assumes homoscedasticity, which is rarely true in real-world data where uncertainty often varies with the input X (heteroscedasticity). To address this and other limitations, several key variants have been developed.

Locally Adaptive Conformal Prediction. Standard SCP often results in inefficient sets (too wide for easy inputs, too narrow for hard ones). To mitigate this, *Conformalized Quantile Regression* (CQR) [Romano et al., 2019] was proposed to construct intervals that adapt their width to the local difficulty of the input. The CQR procedure consists of two main steps: quantile regression training and conformal calibration.

First, using the training set $\mathcal{D}_{\text{train}}$, we train a regression model $\hat{f}$ to estimate two conditional quantiles: the lower quantile at level $\gamma_{\text{lo}} = \alpha/2$ and the upper quantile at level $\gamma_{\text{hi}} = 1 - \alpha/2$. The model outputs $\hat{q}_{\text{lo}}(x)$ and $\hat{q}_{\text{hi}}(x)$, which are optimized by minimizing the pinball loss (or quantile loss) [Koenker and Hallock, 2001]:

$$\mathcal{L}(y, \hat{y}, \gamma) = \max\left(\gamma(y - \hat{y}), (\gamma - 1)(y - \hat{y})\right). \tag{9}$$

The total objective minimizes the average loss over both quantiles: $\sum_{i \in \mathcal{D}_{\text{train}}} [\mathcal{L}(y_i, \hat{q}_{\text{lo}}(x_i), \gamma_{\text{lo}}) + \mathcal{L}(y_i, \hat{q}_{\text{hi}}(x_i), \gamma_{\text{hi}})]$.

Second, although these raw quantile estimates provide a heuristic interval $[\hat{q}_{\text{lo}}(X), \hat{q}_{\text{hi}}(X)]$, they do not guarantee finite-sample coverage. CQR acts as a "wrapper" to rigorously calibrate them. We compute nonconformity scores on the calibration set $\mathcal{D}_{\text{cal}}$ as:

$$s_i = \max\left(\hat{q}_{\text{lo}}(X_i) - Y_i,\ Y_i - \hat{q}_{\text{hi}}(X_i)\right). \tag{10}$$

Intuitively, s_i measures the signed distance from the true label Y_i to the nearest boundary of the predicted interval. If Y_i falls inside the interval, s_i is negative; if it falls outside, s_i is positive.

Let $\hat{q}$ be the $\lceil (n+1)(1-\alpha) \rceil / n$ empirical quantile of these scores $\{s_1, \dots, s_n\}$. The final conformalized prediction interval for a new input X_{n+1} is constructed by expanding (or shrinking) the raw quantile estimates by $\hat{q}$:

$$C_{\text{CQR}}(X_{n+1}) = [\hat{q}_{\text{lo}}(X_{n+1}) - \hat{q}, \quad \hat{q}_{\text{hi}}(X_{n+1}) + \hat{q}]. \tag{11}$$

This procedure ensures valid coverage while allowing the interval width $\hat{q}_{\text{hi}}(X) - \hat{q}_{\text{lo}}(X) + 2\hat{q}$ to vary dynamically based on the input X, significantly improving informativeness in heteroscedastic settings.

Covariate Shift and Weighted Conformal Prediction. The standard exchangeability assumption is violated under distribution shift, where the test distribution $P_{\text{test}}(X)$ differs from the training distribution $P_{\text{train}}(X)$. Under such *covariate shift*, standard CP loses its coverage guarantee. Tibshirani et al. [2019] proposed *Weighted Conformal Prediction* (WCP), which adjusts the quantile calculation using likelihood ratios. For each calibration point $X_i \in \mathcal{D}_{\text{cal}}$, we assign a weight proportional to the density ratio:

$$w(X_i) = \frac{dP_{\text{test}}(X_i)}{dP_{\text{train}}(X_i)} = \frac{p_{\text{test}}(X_i)}{p_{\text{train}}(X_i)}, \tag{12}$$

where p_{test} and p_{train} denote the probability density functions (or probability mass functions) of the test and training distributions, respectively. Intuitively, $w(X_i)$ quantifies how much more likely the input X_i is to appear in the test environment compared to the training environment.

Given these weights, WCP modifies the standard calibration procedure by replacing the uniform empirical distribution with a weighted counterpart. Specifically, after computing nonconformity scores $\{s_i\}_{i=1}^n$ on the calibration set, we assign a normalized probability mass to each point. For a new test input X_{n+1}, the calibration points are weighted by $w(X_i)$, while the test point itself is assigned a fixed weight of 1 (reflecting its draw from the target distribution). The resulting probability weights are defined as:

$$p_i(X_{n+1}) = \frac{w(X_i)}{\sum_{j=1}^n w(X_j) + 1}, \quad p_{n+1}(X_{n+1}) = \frac{1}{\sum_{j=1}^n w(X_j) + 1}. \tag{13}$$

These probabilities are then used to construct the weighted empirical cumulative distribution function of the scores:

$$\hat{F}(s) = \sum_{i=1}^{n} p_i(X_{n+1})\,\mathbb{I}(s_i \leq s) + p_{n+1}(X_{n+1})\,\mathbb{I}(\infty \leq s). \tag{14}$$

The calibrated threshold $\hat{q}$ is determined as the smallest score s such that $\hat{F}(s) \geq 1 - \alpha$. Consequently, the final prediction set is formed as $C_{\text{WCP}}(X_{n+1}) = [\hat{\mu}(X_{n+1}) - \hat{q},\ \hat{\mu}(X_{n+1}) + \hat{q}]$. By incorporating the likelihood ratio into the quantile estimation, this method ensures that the coverage guarantee holds with respect to the target distribution P_{test}.

Conditional Coverage and Mondrian CP. SCP guarantees *marginal* coverage (on average over all inputs), but may systematically under-cover for specific subgroups of the data. To address the need for *conditional* coverage (i.e., validity within specific classes or groups), *Mondrian Conformal Prediction* [Vovk et al., 2005] partitions the calibration data into categories (or ”bins”) and performs calibration separately within each bin. This ensures that the coverage guarantee holds independently for each defined subgroup, preventing the model from sacrificing accuracy on minority classes to satisfy the global average.

3 Headings: first level

First level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 12. One line space before the first level heading and 1/2 line space after the first level heading.

3.1 Headings: second level

Second level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 10. One line space before the second level heading and 1/2 line space after the second level heading.

3.1.1 Headings: third level

Third level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 10. One line space before the third level heading and 1/2 line space after the third level heading.

4 Citations, figures, tables, references

These instructions apply to everyone, regardless of the formatter being used.

4.1 Citations within the text

Citations within the text should be numbered consecutively. The corresponding number is to appear enclosed in square brackets, such as [1] or [2]-[5]. The corresponding references are to be listed in the same order at the end of the paper, in the **References** section. (Note: the standard BIBTEX style `unsrt` produces this.) As to the format of the references themselves, any style is acceptable as long as it is used consistently.

As submission is double blind, refer to your own published work in the third person. That is, use “In the previous work of Jones et al. [4]”, not “In our previous work [4]”. If you cite your other papers that are not widely available (e.g. a journal paper under review), use anonymous author names in the citation, e.g. an author of the form “A. Anonymous”.

Table 1: Sample table title

PART	DESCRIPTION
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

4.2 Footnotes

Indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).[2]

4.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction; art work should not be hand-drawn. The figure number and caption always appear after the figure. Place one line space before the figure caption, and one line space after the figure. The figure caption is lower case (except for first word and proper nouns); figures are numbered consecutively.

Make sure the figure caption does not get separated from the figure. Leave sufficient space to avoid splitting the figure and figure caption.

You may use color figures. However, it is best for the figure captions and the paper body to make sense if the paper is printed either in black/white or in color.

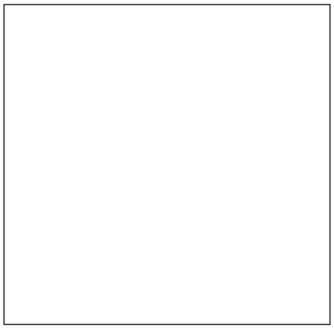

Figure 1: Sample figure caption.

4.4 Tables

All tables must be centered, neat, clean and legible. Do not use hand-drawn tables. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

5 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

[1] Sample of the first footnote

[2] Sample of the second footnote

6 Preparing PostScript or PDF files

Please prepare PostScript or PDF files with paper size "US Letter", and not, for example, "A4". The -t letter option on dvips will produce US Letter files.

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`
- LaTeX users:
  - Consider directly generating PDF files using `pdflatex` (especially if you are a MiKTeX user). PDF figures must be substituted for EPS figures, however.
  - Otherwise, please generate your PostScript and PDF files with the following commands:

    ```
    dvips mypaper.dvi -t letter -Ppdf -G0 -o mypaper.ps
    ps2pdf mypaper.ps mypaper.pdf
    ```

    Check that the PDF files only contains Type 1 fonts.
  - xfig "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
  - The `\bbold` package almost always uses bitmap fonts. You can try the equivalent AMS Fonts with command

    ```
    \usepackage[psamsfonts]{amssymb}
    ```

    or use the following workaround for reals, natural and complex:

    ```
    \newcommand{\RR}{I\!\!R} %real numbers
    \newcommand{\Nat}{I\!\!N} %natural numbers
    \newcommand{\CC}{I\!\!\!\!C} %complex numbers
    ```
  - Sometimes the problematic fonts are used in figures included in LaTeX files. The ghostscript program `eps2eps` is the simplest way to clean such figures. For black and white figures, slightly better results can be achieved with program `potrace`.
- MSWord and Windows users (via PDF file):
  - Install the Microsoft Save as PDF Office 2007 Add-in from `http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=4d951911-3e7e-4ae6-b059-a2e79ed87041`
  - Select "Save or Publish to PDF" from the Office or File menu
- MSWord and Mac OS X users (via PDF file):
  - From the print menu, click the PDF drop-down box, and select "Save as PDF..."
- MSWord and Windows users (via PS file):
  - To create a new printer on your computer, install the AdobePS printer driver and the Adobe Distiller PPD file from `http://www.adobe.com/support/downloads/detail.jsp?ftpID=204` *Note:* You must reboot your PC after installing the AdobePS driver for it to take effect.
  - To produce the ps file, select "Print" from the MS app, choose the installed AdobePS printer, click on "Properties", click on "Advanced."
  - Set "TrueType Font" to be "Download as Softfont"
  - Open the "PostScript Options" folder
  - Select "PostScript Output Option" to be "Optimize for Portability"
  - Select "TrueType Font Download Option" to be "Outline"

- Select “Send PostScript Error Handler” to be “No”
- Click “OK” three times, print your file.
- Now, use Adobe Acrobat Distiller or ps2pdf to create a PDF file from the PS file. In Acrobat, check the option “Embed all fonts” if applicable.

If your file contains Type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

6.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the graphicx package. Always specify the figure width as a multiple of the line width as in the example below using .eps graphics

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.eps}
```

or

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

for .pdf graphics. See section 4.4 in the graphics bundle documentation (`http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to ‘small’ (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

References

JO Berger. Statistical decision theory and bayesian analysis. *Springer series in statistics Show all parts in this series*, 1985.

Clara Fannjiang, Stephen Bates, Anastasios N Angelopoulos, Jennifer Listgarten, and Michael I Jordan. Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43):e2204569119, 2022.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI’98, page 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4): 143–156, 2001.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, volume 30, 2017.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8): 5116–5123, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

Henrik Olsson, Kimmo Kartasalo, Nita Mulliqi, Marco Capuccini, Pekka Ruusuvuori, Hemamali Samaratunga, Brett Delahunt, Cecilia Lindskog, Emiel AM Janssen, Anders Blilie, et al. Estimating diagnostic uncertainty in artificial intelligence assisted pathology using conformal prediction. *Nature communications*, 13(1):7761, 2022.

Harris Papadopoulos, Kyriakos Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*, pages 345–356. Springer, 2002.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234, 2019.

Geethen Singh, Glenn Moncrieff, Zander Venter, Kerry Cawse-Nicholson, Jasper Slingsby, and Tamara B Robinson. Uncertainty quantification for probabilistic machine learning in earth observation using conformal prediction. *Scientific Reports*, 14(1):16166, 2024.

Akshai Parakkal Sreenivasan, Aina Vaivade, Yassine Noui, Payam Emami Khoonsari, Joachim Burman, Ola Spjuth, and Kim Kultima. Conformal prediction enables disease course prediction and allows individualized diagnostic uncertainty in multiple sclerosis. *npj Digital Medicine*, 8(1):224, 2025.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Larry Wasserman. All of statistics. *Springer Texts in Statistics*, 2004.