

Project Submission Pro-Forma

Student name: Yantong.LU

Student ID: u5528854

I wish the dissertation to be considered for the course (select one only):

- ☐ MSc in Cyber Security Engineering
- ☐ MSc in Cyber Security Management
- ☐ MSc in e-Business Management
- ☐ MSc in Engineering Business Management
- ☐ MSc in Healthcare Operational Management
- ☐ MSc in Innovation & Entrepreneurship
- ☐ MSc in Intelligent Manufacturing Systems
- ☐ MSc in International Trade, Strategy & Operations
- ☐ MSc in Programme & Project Management
- ☐ MSc in Supply Chain & Logistics Management
- ☐ MSc in Sustainable Automotive Engineering
- ☒ MSc in Smart, Connected and Autonomous Vehicles

I confirm that I have included in my dissertation:

- ☒ An abstract of the work completed
- ☒ A declaration of my contribution to the work and its suitability for the degree
- ☒ A table of contents
- ☒ A list of figures & tables (if applicable)
- ☒ A glossary of terms (where appropriate)
- ☒ A clear statement of my project objectives
- ☒ A full reference list (the [Harvard referencing style is recommended for WMG](#))
- ☒ An appendix containing email confirmation of ethical approval or waiver

If receiving ethical approval, the ethical approval number for this research is:

WMG-R_204Hy1Z3xjjghWN

☒ I consent to ongoing storage of this dissertation and potential access by third parties (e.g. for staff/student training purposes)

Signed: 

Date: 2024/09/02



Project Title:

Object Detection with Uncertainty Quantification

Yantong.LU

Dissertation submitted in partial fulfilment for the Degree of Master of Science in
Smart, Connected and Autonomous Vehicles

WMG
University of Warwick

Submitted September 2, 2024

Declaration

I have read and understood the rules on cheating, plagiarism and appropriate referencing as outlined in my handbook and I declare that the work contained in this assignment is my own, unless otherwise acknowledged.

No substantial part of the work submitted here has also been submitted by me in other assessments for this or previous degree courses, and I acknowledge that if this has been done an appropriate reduction in the mark I might otherwise have received will be made.

Project definition for my degree (as copied from <http://www2.warwick.ac.uk/fac/sci/wmg/globalcontent/general/project/requirement/>)

The project must meet the following criteria:

1. be based on an aspect of smart connected and autonomous vehicles;
2. be aligned with either an industry supporting body or a current research topic;
3. be of a technical nature, or relate to strategic direction of SCAVs;
4. contextualise social, ethical, economical, political and environmental aspects.

My project relates to this definition in the following way:

The title of this project is 'Object Detection with Uncertainty Quantification'. The object detection model is crucial for accurately identifying objects around the autonomous driving vehicles. This research employs an efficient and straightforward uncertainty quantification method: conformal prediction for the 'Pedestrian Detection' task. It is used on the autonomous driving dataset BDD100K to assess the uncertainty of predicted bounding boxes by the object detection model. Additionally, it generates conformalized bounding boxes to ensure strict coverage of the actual object's bounding box, with a pre-set coverage requirement. By this post-processing method, the model's coverage rate of real objects is improved, thereby enhancing the safety of autonomous driving. Thus, this research is highly relevant to the project requirements of the course.

Conformal Prediction for Spatial Uncertainty in Object Detection

Abstract

Generating predicted bounding boxes is one of the key tasks in object detection. Along with this comes the need to measure the uncertainty of whether these predicted bounding boxes actually contain the true object boxes, referred to as Spatial Uncertainty. Although existing object detection models infer the confidence scores of bounding box occurrences, these probability values are often uncalibrated and sometimes ‘unreliable’. Therefore, we need to employ additional uncertainty quantification methods to more accurately measure the uncertainty of the model's results. Conformal prediction, as one such method, is simple and efficient in computation and can be directly applied to any predictive model without needing modifications to the internal structure of the model. By calibrating and computing the errors between prediction results and ground truth, it provides a rigorous statistical guarantee of the model's coverage of correct samples.

This study extensively builds on the work of de Grancey et al., applying conformal prediction techniques to generate conformalized bounding boxes based on the predicted bounding boxes, which are designed to cover the true object boxes in the image with a predefined probability level. Beyond replicating de Grancey's original conformalization technique and its *observed Coverage* metrics, this paper introduces novel methods including *Coordinate-Adaptation*, *Area-Difference*, and *Image-wise Binary search*, along with a new metric, *Expansion*, which assesses the fit of the conformalized box to the predicted box. Experimental results indicate that while the *observed Coverage* of *Coordinate-Adaptation* and *Image-wise Binary search* methods achieve the preset coverage rates, while *Area-Difference* not. In terms of the *Expansion* metric, except for the *Image-wise Binary Search* which produces significantly larger boxes, the conformalized boxes from other methods closely fit the predicted bounding boxes. Overall, compared to the pretrained models, the experiments on fine-tuned models show higher coverage and lower *Expansion* values for the conformalization methods, emphasizing the importance of model precision for effective conformal prediction.

Keywords: object detection, spatial uncertainty, conformal prediction

Content

Project Submission Pro-Forma	1
Declaration.....	ii
Abstract.....	iii
Content.....	iv
List of tables.....	v
List of figures.....	v
1 Introduction.....	1
2 Literature Review.....	5
2.1 Object detection.....	5
2.1.1 Object Detection Methods.....	5
2.1.2 Dataset related to the experiment	10
2.1.3 The input and output of Object Detection	11
2.2 Conformal Prediction.....	11
2.2.1 Conformal Prediction Algorithm Process.....	12
2.2.2 Extended forms of conformal prediction	13
3 Implementing Conformal Prediction in Object Detection	15
3.1 Designing Nonconformity Scores Across Different Dimensions	15
3.1.1 Coordinate-Wise.....	15
3.1.2 Box-Wise.....	16
3.1.3 Image -Wise.....	18
3.2 Calculating the Conformal Quantile.....	18
3.3 Generating Prediction Sets.....	19
3.4 Evaluation Metrics.....	23
4 Experiment.....	25
4.1 Research Questions	25
4.2 Experimental Setup	25
4.3 Results	27
4.4 Analysis for Research Question in Coverage and Expansion.....	30
4.5 Comprehensive Analysis	35
5 Conclusion	37
Reference	38
Appendices.....	42
Appendix A: Evidence of completing ALL required ethics training.	42
Appendix B: a confirmation email of ethical approval for project.....	44

List of tables

Table 1 Evaluation of observed coverage on \mathcal{D}_{test}	27
Table 2 Baseline work of de Grancey et al. Observed Coverage on \mathcal{D}_{test}	28
Table 3 Evaluation of Expansion on \mathcal{D}_{test} for conformalization methods	28

List of figures

Figure 1 The example of conformal prediction.....	2
Figure 2 Conformalization example	3
Figure 3 Road map of object detection.	6
Figure 4 Architecture of YOLO Detection Model.....	8
Figure 5 Statistics of different object types	11
Figure 6 Possible scenarios of deviation between true box and predicted boxes	16
Figure 7 Examples of conformalized boxes	21
Figure 8 the workflow of experiment	26
Figure 9 Images of conformalized box.....	29

1 Introduction

Zou[1] emphasizes that object detection is a critical task within the autonomous driving domain. The ability of object detection models deployed in car cameras to accurately recognize humans and objects in road environments is directly linked to the safety of drivers and other road participants. In recent years, many state-of-the-art object detection models have been proposed, achieving impressive performances across various datasets.

However, an issue that cannot be overlooked is that the predictions and accuracy of these models can be “unreliable”. In other words, despite some models providing so-called “confidence scores” to measure and guarantee this uncertainty of model outputs, various factors can lead to an overestimation or underestimation of these confidence scores. These factors include confidence scores not being calibrated in the test set, changes in data distribution when models are applied to new datasets, and models being overly confident or not confident enough about their predictions. Clearly, in these cases, confidence scores are not a reasonable measure of a model's true “uncertainty”[2].

Specifically, the two core tasks in object detection: object classification and generating prediction bounding boxes, also lack a reasonable measure of uncertainty. The reasons are: (1) Confidence scores for class categorization are often not calibrated. (2) There is a lack of uncertainty estimation for the deviation of bounding boxes[3].

Therefore, additional methods for measuring uncertainty are needed to provide "reliable" uncertainty measurements for object detection model outputs.

Uncertainty Quantification Methods for Object Detection

Although it is impossible to eliminate uncertainties, researchers have developed various methods to measure them. For example, Monte Carlo (MC) dropout[4] is efficient, low-cost method for uncertainty quantification. Deep Bayesian Active Learning (DBAL) combine active learning (AL) framework and Bayesian deep learning (DL) techniques to handle high-dimensional data challenges[5]. Deep ensemble[6] reduce the risk of overfitting across different data sets by combining predictions from various models. In recent years, ‘Probabilistic Object Detection’[7] has integrated methods such as (MC) dropout, Bayesian Neural Networks (BNNs) and Ensembles methods with foundational object detection models to reduce both *Semantic Uncertainty* (the uncertainty of whether object detection models classify objects into specific categories) and *Spatial Uncertainty* (the uncertainty regarding whether the bounding boxes generated by object detection models accurately locate and cover the objects).

The methods mentioned above have been applied to object detection and have achieved certain success, but there are still some issues: (1) Most importantly, these methods do not provide statistical guarantees about the estimated uncertainties[2]. (2) Some of these methods have additional requirements for the data distribution, such as

being independent and identically distributed or satisfying prior probabilities[8]. (3) These methods involve modifications to the model structure and algorithms, making them overly complex to apply.

However, Angelopoulos et al.[9] discovered a statistical method called 'Conformal Prediction', which can serve as a post-processing method to measure and provide probabilistic guarantees for the semantic uncertainty of model outputs. To put it simply, as shown in Figure 1, suppose we set an acceptable error rate at α , and feed a set of images containing squirrels to an animal recognition model, the model typically outputs the categories of the identified objects along with corresponding confidence levels. Conformal prediction, then, generates an additional prediction set for each model prediction, which collectively covers the true categories in the ground truth with a coverage rate of $1 - \alpha$. The fox squirrel in the picture is an example of a correct category successfully included within the prediction set. This simple example can be seen as conformal prediction measuring Semantic Uncertainty while strictly ensuring the coverage process.



Figure 1 The example of conformal prediction

Furthermore, in 2022, de Grancey et al.[2] noted that while there has been substantial research on the application of conformal prediction for Semantic Uncertainty in object detection model categories, there is a notable lack of research on applying conformal prediction to the Spatial Uncertainty of bounding boxes. They first proposed and designed specific conformalization methods using 'Conformal Prediction' as a post-processing method to measure and provide probabilistic guarantees for the Spatial Uncertainty of the bounding boxes predicted by the Yolov3 model.

Specifically, de Grancey et al. use the YOLO model pretrained on the COCO dataset and focus only on the bounding box of the 'person' category in the dataset.

The process of their work involves: (1) They design different nonconformity score function to calibrate the errors between predicted boxes and true boxes across three dimensions: coordinate, the entire bounding box, and the image containing the bounding box. (2) Then, for new inputs, using these errors, they infer additional conformalized bounding boxes based on the predicted boxes to ensure coverage of the actual true box with a predefined coverage containing the true boxes (the specific process will be mentioned in subsequent sections).

Conformal Prediction is highly efficient and can be applied to the outputs of any predictive model, providing probabilistic guarantees for model uncertainty, which holds significant research value. Meanwhile, research on the Spatial Uncertainty of bounding boxes is just beginning, representing a novel field.

Therefore, this project primarily builds on the "pedestrian detection" work of de Grancey et al., trying to apply conformal prediction to the predicted bounding boxes of a baseline model's output, ensuring that the newly generated conformalized boxes cover the actual bounding boxes with a predefined probability. As illustrated in Figure 2, the blue boxes represent the true boxes for the "person" category within the dataset, the yellow boxes are the predicted boxes derived from the YOLOv3 model inference, and the purple boxes, the focus of this study, are the new conformalized boxes generated through the conformal prediction process by adding margins to the predicted boxes. Across the entire BDD100K dataset, this paper's *Coordinate-Adaptation* method ensures that all conformalized boxes cover $1 - \alpha$, which equates to 90% of the true boxes.

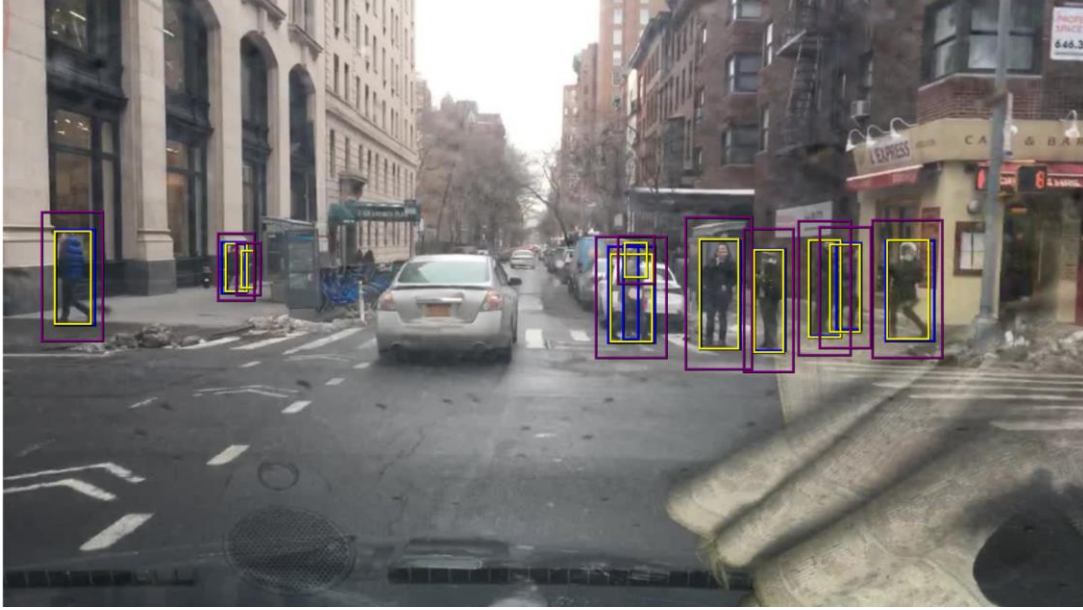


Figure 2 Conformalization example (*Coordinate-Adaptation*, risk level $\alpha = 0.1$) on a BDD100k image with Ground Truth, Inference and Conformalized boxes.

The main contributions of this paper are as follows:

(a) To investigate the impact of a fine-tuned model on the process of conformal prediction, unlike the original work that only uses a pretrained YOLOv3 model, this paper fine-tunes a new YOLOv3 model on the BDD100K dataset for subsequent comparative experiments.

(b) This study not only replicates all methods from de Grancey's research but also theoretically designs and experiments with new methods of conformalization on the dataset: *Coordinate-Adaptation*: This method, by incorporating the width and height of

the predicted bounding box into the nonconformity score function step, ensures that the generated conformalized boxes better fit the predicted boxes.

(c) Another new conformalization method: *Area-Difference*: By considering the four coordinates of the bounding box together based on area, this method addresses, to some extent, the issues of a reduction in the actual coverage rate of the conformalized boxes over the real boxes.

(d) In the implementation of image-level conformalization, a *Binary-Search* algorithm is used to find the value that satisfies the minimal margin, supplementing the parts of de Grancey et al.'s study that were not detailed.

(e) Introduce a new evaluation metric, *Expansion*, which effectively measures whether different conformalization techniques generate conformalized boxes that closely fit the predicted boxes.

The paper is structured as follows:

Section 2 provides a systematic literature review of common models, datasets, input-output processes in object detection, and the conformal prediction algorithm, including its variants.

Section 3 delves into a detailed explanation of the methodologies employed in the experimental part of this paper, starts with the standard conformal prediction algorithm process, from designing the nonconformity score function, to calculating the quantile, and finally explaining in detail how to generate the prediction set (i.e., conformalized box) based on the predicted bounding box, how conformal prediction achieves coverage assurance, and new methods: *Coordinate-Adaptation*, *Area-Difference*, *Binary-Search* are introduced, then a new evaluation metric, *Expansion*, is proposed.

Section 4 presents the research questions, set up for experiments, and finally analyses the results of comparative experiments of all conformalization techniques.

Section 5 summarizes the limitations of this study and the potential future applications of using conformal prediction to measure spatial uncertainty.

2 Literature Review

This chapter is divided into two sections: (1) Object Detection, which systematically reviews the datasets, input-output mechanisms, and common conformal prediction and its variant methods involved in this study; (2) Conformal Prediction, including the concept of CP, its advantages, basic algorithmic processes, and common variants. Due to the inherent advantages of conformal prediction, which does not require modifications to the internal structure, it can theoretically be directly applied to all target detection algorithms.

2.1 Object detection

2.1.1 Object Detection Methods

The core task of object detection includes correctly classifying objects within images and generating corresponding predicted bounding boxes[10]. In Section 2.1, following the classification methods for object detection models in references [11] and [12] this document categorizes object detection models into two types: traditional and deep learning methods. Deep learning is further divided into two-stage and one-stage detectors, as shown in the Figure3.

Traditional object detection methods

Before the era of deep learning revolutionized object detection, early computer scientists lack efficient image representation and computation methods. They have to design complex image processing and feature extraction processes.

Viola-Jones detectors (VJ detector, 2001)[13]: Proposed for real-time face detection, the Viola-Jones detector operates faster than most algorithms at the time while ensuring a certain level of accuracy. Its operational principle, which appears straightforward today, involves a sliding window that traverses the image to detect the presence of a face within the window.

Histogram of Oriented Gradients (HOG, 2005) Detector[14]: HOG starts by dividing the image into many small grid cells, calculating the histogram of gradient directions within each cell to capture pixel variations within the area, such as the direction of edges. After calculating the features of each small cell, HOG combines some adjacent cells into "blocks" and normalizes these blocks to reduce the impact of feature variations caused by shifts and lighting.

Deformable Part-based Model (DPM, 2008)[15]: DPM adopts a "divide and conquer" detection philosophy. Specifically, the model learns to decompose an object into multiple parts using a "star-model"; it then assesses the presence of these different parts to determine the presence of the entire object. This method allows the model to better adapt object appearances' variations.

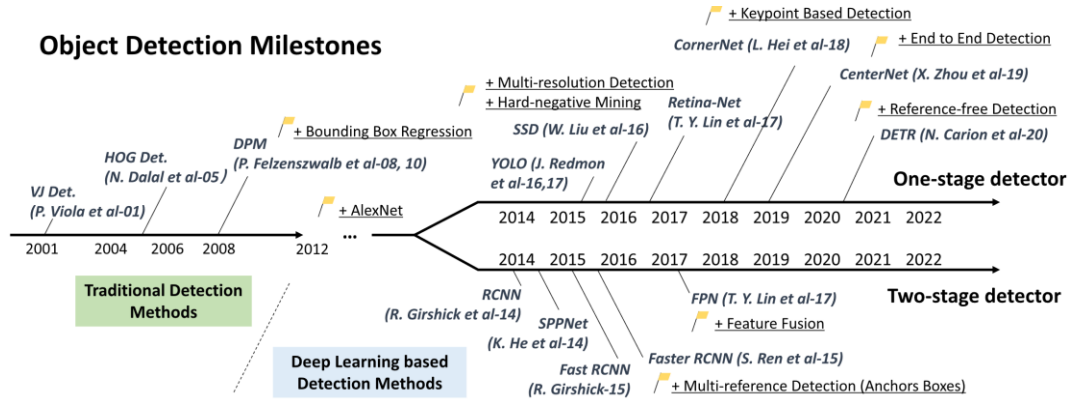


Figure 3 Road map of object detection. [12]

Deep learning-based methods

With the advent of Convolutional Neural Networks (CNNs) in 2012 [16], which are capable of learning to extract high-level image features, computer vision scientists began to explore the integration of CNNs into object detection. In 2014, Girshick et al. [17] pioneered Region-based Convolutional Neural Networks (R-CNN), marking the entry of object detection into the era of deep learning. During this period, object detectors can be categorized into "two-stage detectors" and "one-stage detectors"[11].

Two-Stage Detectors do not directly detect objects across the entire image; instead, they initially extract regions likely containing objects, known as the region proposal step. For each candidate region, the detector further determines the type of object and refines its bounding box, enhancing the accuracy of the detection. one-stage detectors.

One-Stage Detectors eliminate the separate region proposal step, conducting object detection and bounding box predictions directly across the whole image, which simplifies the process. These are faster and more efficient, often used in real-time detection scenarios such as video surveillance and autonomous driving.

A. Two Stage Detector

Region-based Convolutional Neural Networks (RCNN): R-CNN[18] begins with a Selective Search algorithm to generate a set of object proposal candidates, also known as candidate boxes. Each candidate box is resized to a fixed dimension and then fed into a pretrained convolutional neural network (e.g., AlexNet[16]) to extract features. The extracted features are then input into a linear SVM classifier to determine whether the region contains a particular type of object and to identify the object's category. However, as each candidate box requires individual feature computation through the CNN, this results in a significant slowdown in detection speed.

Spatial Pyramid Pooling Networks (SPPNet): To address the efficiency issues of R-CNN, the Spatial Pyramid Pooling Network (SPPNet) [19] was introduced. The core of SPPNet is the SPP layer, which can generate a fixed-length feature representation from

input of any size. This resolves the traditional CNN requirement for fixed-size input. SPPNet needs only a single computation of the feature map on the entire image, significantly improving detection speed. Although SPPNet greatly enhances speed, it has some limitations: the training process is relatively complex and requires multiple stages. Additionally, SPPNet only fine-tunes the fully connected layers while neglecting fine-tuning of the convolutional layers, which limits the model's optimization potential.

Fast R-CNN: In 2015, Fast R-CNN[20] was proposed by Girshick et al. It introduces the ROI pooling layer, which allows for the extraction of fixed-length features from regions of varying sizes from the feature map computed from the whole image. This avoids the repetitive feature computation for each proposal detection, as seen in SPPNet. On the VOC07 dataset, Fast R-CNN increased the mean Average Precision (mAP) from 58.5% in R-CNN to 70.0%, and the detection speed was more than 200 times faster than R-CNN[11]. Despite significant performance improvements, the speed of Fast R-CNN is still limited by the time-consuming proposal detection step.

Faster R-CNN: Faster R-CNN[21] was introduced by Ren et al. in 2015, solving the efficiency bottleneck of proposal detection in Fast R-CNN through the introduction of the Region Proposal Network (RPN). It can generate region proposals nearly cost-free directly on the CNN feature map. On the COCO dataset, it achieved an mAP@0.5 of 42.7%, and on the VOC07 dataset, and mAP of 73.2%, with detection speeds of up to 17 frames per second (fps) using ZF-Net. It is the first deep learning-based object detector to approach real-time performance.

B. One-Stage Detector

Although most two-stage detectors can easily achieve high precision, their complexity and slow inference speeds often limit their use in engineering applications. In contrast, one-stage detectors are widely used in the industry due to their ability to predict in real-time and simple deployment[11].

You Only Look Once

YOLO[22] was proposed by Joseph et al. in 2015 and represents the first one-stage detector of the deep learning era. YOLO employs a single neural network to evaluate the entire image. This network divides the image into multiple grids, each of which predicts bounding boxes and the probability that these boxes contain objects. As YOLO only requires a single forward pass to complete the detection process, it is exceptionally fast. A fast version of YOLO can achieve up to 155 frames per second (fps), and a more accurate version can achieve 45 fps, with a mAP of 63.4% on the VOC07 dataset[11].

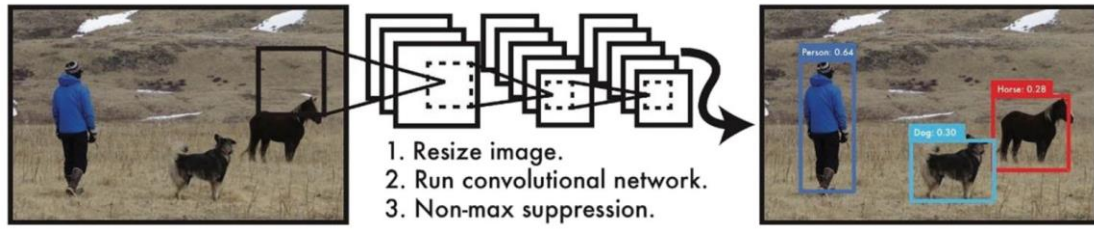


Figure 4 Architecture of YOLO Detection Model [23]

Although YOLO's detection speed is rapid, its localization accuracy is reduced compared to two-stage detectors, particularly for small objects. This is due to YOLO's simple division of the image, which can lead to small objects being overlooked or inaccurately positioned.

YOLOv2 and YOLOv3[23] introduced the Darknet architecture, which added more convolutional and pooling layers to improve the detection capabilities for small objects.

YOLOv4 employs the CSPDarknet53 as its backbone network and incorporates the Spatial Pyramid Pooling (SPP) module, which captures features at different receptive fields, making the model more robust to objects of various scales[24].

YOLOv5 is developed in the PyTorch framework, offering enhanced flexibility and richer community support. It also introduced automatic learning of bounding box anchors and data augmentation techniques to further enhance performance[25].

YOLOv7[26], proposed in 2022, introduced a trainable optimization method known as "bag-of-freebies," outweighing most existing object detectors in both speed and accuracy. YOLOv7 also uses a dynamic label assignment strategy and re-parameterizes the model structure, enhancing computational efficiency during inference while maintaining or even improving detection accuracy.

Given that this paper later employs YOLOv3 as the experimental model, a brief introduction to the methodological workflow of YOLOv3 is provided:

First step: YOLOv3 utilizes the Darknet53 to extract three feature layers at different resolutions. This design enables the network to capture targets at various scales, enhancing its ability to detect small objects and improving detection accuracy and speed.

Second step: YOLOv3 segments the entire image into grids of varying densities. Each grid cell predicts multiple bounding boxes, which are then transformed into actual bounding box coordinates through a decoding process. This decoding process involves mapping the predicted values to the actual grid locations and uses the sigmoid function for activation, constraining the output values within the range $[0,1]$.

Third step: Low probability bounding boxes are filtered out based on confidence scores, and then Non-Maximum Suppression (NMS)[27] is employed to eliminate boxes that significantly overlap, retaining only those most likely to contain targets.

Single Shot MultiBox Detector (SSD) SSD[28] introduced multireference and multiresolution detection techniques. Detection is performed at different layers of the network, utilizing feature maps from various layers to detect objects of different scales. Lower layers, with higher resolution, are suitable for detecting smaller objects, while

higher layers, with larger receptive fields, are better for detecting larger objects. SSD is capable of performing detection at high frame rates, for instance, its faster version can achieve 59 frames per second (fps), and it achieves an $\text{mAP}@0.5$ of 46.5% on the COCO dataset[11].

RetinaNet was proposed by Lin et al. in 2017 [29] to address the core issue where single-stage detectors lag behind two-stage detectors in accuracy due to foreground-background class imbalance. During training, single-stage detectors encounter a large number of background regions that significantly dilute the impact of foreground objects, making the detectors easy to overlooking important, hard-to-classify foreground objects. A new loss function, Focal Loss, was introduced, which modifies the traditional cross-entropy function to assign higher weights to hard-to-classify samples while reducing the impact of easy-to-classify samples. This makes the detector focus more on challenging cases that are prone to errors during training. RetinaNet consists of a unified network architecture, including a backbone network (such as ResNet) and two subnetworks, one for classification and one for bounding box regression. While maintaining high detection speeds, RetinaNet achieved an $\text{mAP}@0.5$ of 59.1% on the COCO dataset, marking a significant advancement for single-stage detectors.

CornerNet[30] employs a novel single-stage detection approach, abandoning traditional anchor boxes and instead directly predicting two key points: the top-left and bottom-right corners of an object's bounding box. These key points are represented on two heatmaps, and an embedding vector is used to match corner points belonging to the same object. Traditional methods place a large number of anchor boxes on images, leading to an imbalance in the ratio of positive to negative samples. Moreover, anchor boxes require the design of multiple parameters (such as quantity, size, and ratio). CornerNet reduces the number of negative samples by directly detecting key points, thereby avoiding complex parameter settings. Additionally, CornerNet introduces a corner pooling layer to better locate the corners of an object's bounding box. With these innovations, CornerNet achieved superior performance on the COCO dataset compared to most contemporary single-stage detectors, achieving an $\text{mAP}@0.5$ of 57.8%, effectively enhancing detection accuracy.

CenterNet[31], introduced in 2019, transforms object detection into a center point prediction problem, using an object's center point to represent the target and simultaneously regressing its dimensions, orientation, and other attributes. This method eliminates the need for anchor boxes and post-processing steps, such as non-maximum suppression (NMS). CenterNet feeds an image into a fully convolutional network, producing a heatmap whose peak positions correspond to the object's center points. The model regresses the target's dimensions and offsets from the center point and uses various loss functions (keypoint loss, offset loss, and dimension loss) to optimize performance. This approach can be applied to multiple tasks, including 3D object detection, pose estimation, optical flow learning, and depth estimation, achieving a

COCO mAP@.5 of 61.1%.

DETR[32] (Detection Transformer), introduced by Facebook in 2020, is an end-to-end object detection method based on the Transformer architecture that views object detection as a set of prediction problem, without using anchor boxes. Utilizing a pretrained ResNet as the backbone to extract features, the Transformer encoder encodes global image features through a multi-layer attention mechanism, while the decoder use learnable object queries for target prediction. The model directly outputs the object categories and bounding boxes without the need for anchor boxes and NMS. DETR achieved a mAP@0.5 of 71.9% on the MSCOCO dataset, outperforming Faster R-CNN in detecting medium and large objects. However, DETR exhibits weaker performance in detecting small objects and faces challenges in learning convergence due to its match loss design. To address the long convergence time and insufficient performance on small objects, Deformable DETR[33] was proposed, significantly enhancing the model's efficiency and detection capabilities for small objects.

2.1.2 Dataset related to the experiment

COCO[34] (Common Objects in Context) In the experiments of this paper, one of the models used is pretrained on the COCO dataset. Here is a brief introduction to the COCO dataset. COCO is a large-scale image recognition dataset used for object detection and image segmentation. It contains over 330,000 images with 91 object categories, 80 of which have more than 5,000 labeled instances each, totaling approximately 2.5 million labeled instances. Each image in COCO has average 7.7 object instances, which is higher than ImageNet (3.0) and PASCAL (2.3), helping in the learning of inter relationships within scenes. The COCO dataset provides precise pixel-level segmentation annotations for each object instance, a feature uncommon in other datasets such as PASCAL VOC and SUN, which helps enhance the model's ability to accurately localize objects. However, we can estimate that due to some categories having significantly more samples than others, the model may overfit to frequent categories during training and perform poorly on rarer categories.

BDD100K Dataset[35]: The actual data object of this paper is the BDD100K Dataset. The BDD100K dataset is a large-scale, widely used open driving video dataset for computer vision research, released by the University of California, Berkeley. The BDD100K dataset includes labels for ten categories: Bus, Light, Sign, Person, Bike, Truck, Motor, Car, Train, and Rider, totaling approximately 1.84 million annotated bounding boxes. The bar chart below displays the quantity of different object categories.

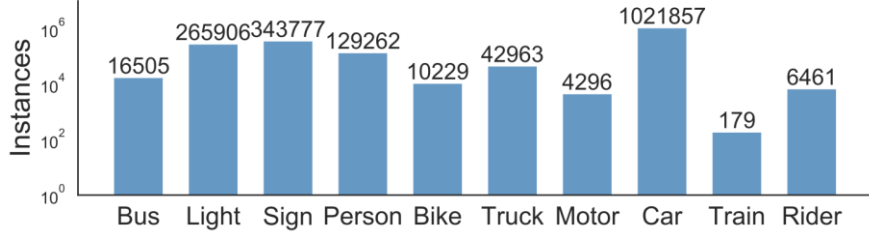


Figure 5 Statistics of different object types

2.1.3 The input and output of Object Detection

In this study, we do not need to make structural changes to the object detection model; instead, we directly use the results output by the model. Furthermore, conformal prediction serves as a post-processing method for the model and does not impose any requirements on the pre-trained model. Thus, this section simply standardizes the model's input and output for ease of understanding and subsequent experiments.

Assume that in our dataset we have an image $X \in \mathbb{R}^{H \times W \times D}$, where H , W , D correspond to the height, width, and depth of the image, respectively. Each image X is associated with a label tuple $L = \{(x_{min}, y_{min}, x_{max}, y_{max}, l)_j\}_j^{O(X)}$. Here, $O(X)$ indicates the actual number of objects present in the image, with $j = 1, \dots, O(X)$. The coordinates $(x_{min}, y_{min}, x_{max}, y_{max}) \in \mathbb{R}$ represent the coordinates of the lower left and upper right corners of the object's bounding box. The category of each object is denoted by l , $l \in \{1, \dots, K\}$.

The image X is input into a pre-trained object detection model \hat{f} , which through inference predicts $\hat{O}(X)$ object. The output for each predicted object is given by $\{(\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max}, \hat{l}, p_{obj}, p_{box})_j\}_j^{\hat{O}(X)}$, where $j = 1, \dots, \hat{O}(X)$.

Here, $\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max}$ specify the coordinates of the predicted bounding box, \hat{l} specifies the predicted category, p_{obj} quantifies the probability associated with the predicted category, p_{box} reflects the model's confidence in the accuracy of the bounding box.

2.2 Conformal Prediction

Conformal Prediction (CP) is a member of uncertainty quantification methods that offers several key advantages. First, it is computationally efficient and does not rely on specific data distribution assumptions, making it robust in situations where the training and test data are not drawn from the same distribution.[36]. CP is flexible because it can be applied to any predictive model, as it works with the model's outputs rather than its internal processes, making it suitable for a wide range of applications from simple models to complex deep learning systems[9]. Moreover, CP provides a probabilistic

guarantee, creating prediction intervals or sets that are likely to contain the true outcome with a specified confidence level, making it valuable for risk assessment and decision-making. Why conformal prediction holds these advantages can be validated through the detailed algorithmic steps and formulas presented below.

2.2.1 Conformal Prediction Algorithm Process

Assume we have three datasets: a training set \mathcal{D}_{train} , used for training the foundational model, a calibration set \mathcal{D}_{cal} used for calculating and calibrating the discrepancies between the model outputs and the ground truth, and an unlabeled test set \mathcal{D}_{test} that the model has not previously encountered. For these datasets, this paper employs the most used split CP algorithm, with the process detailed as follows:

Algorithm 1 Split conformal prediction

1: **Input:** data $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$, prediction algorithm \mathcal{A} , tolerated miscoverage $\alpha \in (0,1)$

2: **Output:** Prediction set $\hat{\mathcal{C}}(X_{n+1})$ for test sample $(X_{n+1}, Y_{n+1}) \in \mathcal{D}_{test}$

3: **Procedure:**

4: Split data \mathcal{D} into two non-intersecting subsets: a training set \mathcal{D}_{train} and calibration set \mathcal{D}_{cal}

5: Train a machine learning model on the training set:

$$\hat{f}(\cdot) \leftarrow \mathcal{A}(\mathcal{D}_{train})$$

6: Define a score function $s : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, and the learned model $\hat{f}(\cdot)$ is evaluated by s on \mathcal{D}_{cal} , resulting in a set of nonconformity scores

$$S = \{s(\hat{f}(X_i), Y_i)\}_{i=1}^n = \{s_i\}_{i=1}^n$$

Where score s_i represents a notion of dissimilarity (nonconformity) between the prediction $\hat{f}(X_i)$ and the ground truth Y_i , we calculate the nonconformity S on all data points of \mathcal{D}_{cal} , n is the size of \mathcal{D}_{cal}

7: Define a conformal quantile \hat{q} as:

$[(n+1)(1-\alpha)/n]$ -th quantile of S , which means $[(n+1)(1-\alpha)/n]$ -th largest value among the nonconformity S

Under exchangeability of $\mathcal{D}_{cal} \cup \{(X_{n+1}, Y_{n+1})\}$, the conformal quantile \hat{q} is a finite sample-corrected quantile ensuring target coverage $(1-\alpha)$ by its construction

8: Given a new input $(X_{n+1}, Y_{n+1}) \in \mathcal{D}_{test}$, instead of simply outputting a $\hat{f}(X_{n+1})$, a conformal prediction set $\hat{\mathcal{C}}(X_{n+1})$ for X_{n+1} can be defined as

$$\hat{\mathcal{C}}(X_{n+1}) = \{y \in \mathcal{Y} : s(\hat{f}(X_{n+1}), y) \leq \hat{q}\}$$

In the specific steps of the conformal prediction algorithm, we only require basic conditions for the datasets (including training, calibration, and test sets): the data in the

calibration and test sets need to be exchangeable. This is in contrast to the i.i.d. (independent and identically distributed) condition, which requires not only that the random variables are independent but also that each variable shares the same probability distribution. Under the assumption of exchangeability, there can be some dependence between the random variables in the dataset, with the overall probability distribution of the sequence remaining unchanged regardless of the order of the variables, representing a weaker requirement.

It is noteworthy that there are no associative requirements among the training, calibration, and test sets within the dataset requirements. Theoretically, the training set data can differ from the distributions of the latter two sets; however, a model trained on a specific data distribution may perform poorly on a completely different distribution, hence the significant errors during the subsequent calibration phase.

According to the proofs by references[8], [9], if the dataset requirements are met and the above algorithmic steps are strictly followed, we can ensure a probabilistic guarantee that:

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}(X_{n+1})\right) \geq 1 - \alpha$$

To put it in simple, for a new data point X_{n+1} , the probability that the prediction set $\hat{C}(X_{n+1})$ contains the true label Y_{n+1} will be at least $1 - \alpha$. In other words, if we repeatedly conduct the conformal prediction independently, the probability of an erroneous prediction will not exceed α . The probabilistic guarantee of this paper is valid based on exchangeability, indicating that it does not require the independence of random variables, thereby broadening the applicability of the theorem and relaxing the conditions.

Due to the inherent advantages of conformal prediction, there is no need to modify the internal structure of the model; it only requires the application of the model's output. Theoretically, conformal prediction can be applied to any predictive model.

2.2.2 Extended forms of conformal prediction

In **group-balanced conformal prediction**, the aim is to ensure that prediction intervals are fair and consistent across different groups identified within the data. This method modifies the traditional conformal prediction technique to prevent disparities in error rates that can occur in varied demographic groups. This is especially crucial in fields like healthcare, where equitable predictions are essential.

The foundation of this method is to ensure that for any test instance from a particular group g , the prediction intervals correctly cover the true value with a probability of at least $1 - \alpha$ where α is the acceptable error rate. This is mathematically represented as:

$$P(Y_{test} \in C(X_{test}) | X_{test,1} = g) \geq 1 - \alpha$$

To achieve this, the conformal scores are calculated separately for each group using a specific scoring function. These scores are then used to determine the conformal

quantile $\hat{q}^{(g)}$ for each group g . This quantile calculation is crucial as it sets the threshold specific to each group's characteristics and is defined by

$$\hat{q}^{(g)} = \text{Quantile}(\{s_1^{(g)}, \dots, s_{n(g)}^{(g)}\}; \frac{(n(g) + 1)(1 - \alpha)}{n(g)})$$

Where $n(g)$ is the count of samples in that group. Finally, for a new observation x , the prediction set $C(x)$ is formed using the quantile of the group to which x belongs. The set includes all potential outcomes y , that are below this quantile threshold, ensuring that the predictions remain within the specified error margins for each group. This is formulated as:

$$C(x) = \{y: s(x, y) \leq \hat{q}^{(x_1)}\}$$

Where x_1 is the group identifier of x

Through these steps, group-balanced conformal prediction not only meets statistical coverage requirements but also ensures that predictions are fair and balanced across all groups, tackling a significant challenge in sensitive and diverse application areas.

Conformal risk control is another form of conformal prediction which extends traditional CP methods by not only managing miscoverage but also controlling a broader range of error types using various loss functions. Traditionally, conformal prediction aims to ensure that the miscoverage probability, $P(Y_{test} \notin C(X_{test})) \leq \alpha$, remains below a user-defined threshold. This approach primarily focuses on the presence or absence of the true value within the predicted set.

The core of conformal risk control is to determine an optimal λ that balances the coverage and the size of the prediction set, ensuring the expected loss does not exceed a specified error rate α . The loss function used, ℓ , measures the quality of the prediction relative to this threshold, ensuring that:

$$E[\ell(C_{\hat{\lambda}}(X_{test}), Y_{test})] \leq \alpha$$

where the loss function ℓ shrinks with increasing size of C

$$P(\ell(C_{\hat{\lambda}}(X_{test}), Y_{test}) \leq \alpha)$$

To find the appropriate λ , we define an infimum over possible values of λ that keeps the empirical risk within a desired range, adjusted slightly for a conservative estimate:

$$\lambda = \inf \{ \lambda: \hat{R}(\lambda) \leq \alpha - \frac{B - \alpha}{n} \}$$

Where $\hat{R}(\lambda)$ is the average of the losses over the calibration dataset, capturing the empirical risk.

3 Implementing Conformal Prediction in Object Detection

This chapter relies on the algorithmic workflow of the general conformal prediction method discussed in section 2.2, explaining how to apply conformal prediction as a post-processing technique to the bounding boxes predicted by object detection models, thereby ensuring that the newly generated conformalized bounding boxes cover the true bounding boxes with a probability $1 - \alpha$.

Specifically, this construction process consists of three steps: (1) designing a nonconformity score function, (2) computing the conformal quantile, and (3) generating the prediction set (i.e., how to use the computed quantile to allocate margins to the predicted bounding boxes).

We will follow the CP methods of de Grancey et al.[2], employing a detailed approach that builds bounding boxes from three different perspectives: Coordinate-Wise, Box-Wise, and Image-Wise, to construct the three steps mentioned above.

These three perspectives all treat bounding boxes as independent data points. The primary distinction lies in the design of the nonconformity score function: Coordinate-Wise calculates errors for each coordinate of the bounding box independently, Box-Wise treats the four coordinates of a bounding box as a whole to compute errors, and Image-Wise calculates errors across all bounding boxes in an image. In addition to the methods described in the original text, this paper innovates in the Box-Wise and Image-Wise conformalization by introducing new methods: *Coordinate-Adaptation* and *Area-difference* in box-wise, and *Binary-search* in image-wise.

Finally, in addition to the original baseline evaluation metric *Coverage*, we designed a new metric: *Expansion* to evaluate the extension of the conformalized boxes compared to the predicted boxes for different methods.

3.1 Designing Nonconformity Scores Across Different

Dimensions

3.1.1 Coordinate-Wise

Assume $k = 1, \dots, n$ where k denotes the index of the k -th ground-truth bounding box in the calibration dataset \mathcal{D}_{cal} . As illustrated in Figure 6, $Y^k = (x_{min}, y_{min}, x_{max}, y_{max})$ represents the coordinates of the lower-left and upper-right corners of the k -th bounding box. $\hat{Y}^k = (\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max})$ represents the coordinates of the bounding box predicted by the model.

Referencing [2], the nonconformity score of for Coordinate-Wise can simply be obtained by calculating the differences between the coordinates of the true box and the predicted box, as given in Equation (1). Figure 6 merely demonstrates two possible scenarios: where the predicted box completely covers the true box, and where the predicted box deviates from the true box. The values in the formula can be positive or

negative, representing the degree and direction of deviation.

$$\begin{aligned} R_{x_{min}}^k &= \hat{x}_{min}^k - x_{min}^k & R_{y_{min}}^k &= \hat{y}_{min}^k - y_{min}^k \\ R_{x_{max}}^k &= x_{max}^k - \hat{x}_{max}^k & R_{y_{max}}^k &= y_{max}^k - \hat{y}_{max}^k \end{aligned} \quad (1)$$

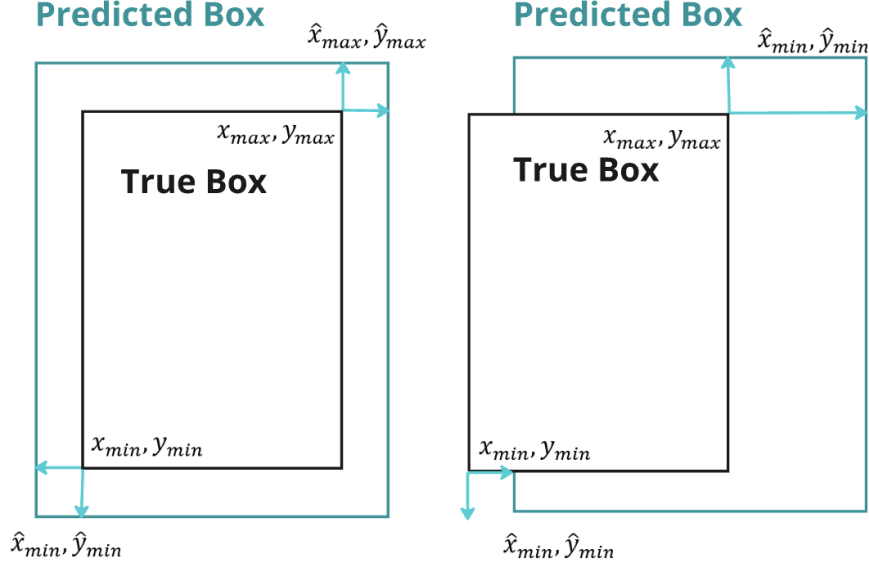


Figure 6 Possible scenarios of deviation between true box and predicted boxes

3.1.2 Box-Wise

At the box-wise level, we consider the bounding box as a whole and calculate the error. Therefore, taking inspiration from the Coordinate-Wise approach, the most intuitive design for a nonconformity score function is to take the overall difference between the two coordinate points, defined as the "*coordinate-difference*" method:

$$R_{box}^k = (\hat{x}_{min}^k - x_{min}^k, \hat{y}_{min}^k - y_{min}^k, x_{max}^k - \hat{x}_{max}^k, y_{max}^k - \hat{y}_{max}^k) \quad (2)$$

However, this method may introduce some challenges. Observations from the BDD100K dataset reveal significant size variations among the bounding boxes of different objects; larger objects have larger ground-truth and predicted boxes, leading to more significant differences. The calculated errors are therefore larger for bigger objects and smaller for smaller objects, which may cause the score function R_{box}^k and the subsequent conformal prediction calculations to be disproportionately influenced by larger objects, potentially neglecting smaller ones. As an experimental approach, I propose a new nonconformity score as shown in Equation (3), termed the "*coordinate-adaptation*" method:

$$R_{box}^k = \left(\frac{\hat{x}_{min}^k - x_{min}^k}{\hat{w}^k}, \frac{\hat{y}_{min}^k - y_{min}^k}{\hat{h}^k}, \frac{x_{max}^k - \hat{x}_{max}^k}{\hat{w}^k}, \frac{y_{max}^k - \hat{y}_{max}^k}{\hat{h}^k} \right) \quad (3)$$

where \hat{w}^k and \hat{h}^k represent the width and height of the predicted box, respectively.

This normalization reduces the influence of larger objects and enhances the impact of smaller ones, thereby improving the method's robustness and attempting to make the subsequent bounding boxes better fit the actual object sizes.

In validating the box-wise "*Coordinate-difference*" method, if we view the conformalized boxes entirely cover the true boxes as a successful coverage. de Grancey et al. found that under a 95% coverage requirement ($1 - \alpha$), only 86% of the real boxes were covered by the conformalized boxes (this deviation between actual and theoretical coverage rates was also confirmed in my experiments). de Grancey et al. noted that the failure to meet the coverage expectations is due to the independent processing of each bounding box coordinate in Equation (2), where each coordinate's individual coverage guarantee does not imply that the conformalized boxes can ensure overall coverage. To maintain the risk level α while covering all four coordinates, they adopted the Bonferroni correction[37] at the box level, constructing the prediction sets with an error coverage rate of $\alpha/4$ for each coordinate. This stricter coverage requirement was proven to ensure that the actual coverage rate met the set expectations in their experiments.

Inspired by the methods of de Grancey et al. and considering the issues arising from not considering all four coordinates together, a stricter control over the error rate α is necessary. If I do not use the Bonferroni correction but intuitively consider the areas of the predicted and true boxes, can the formula ensure actual coverage by the four coordinates? Therefore, I designed the nonconformity score as shown in Equation (4), which I named the "*area difference*" method, considering all four coordinates at the box level for subsequent experiments.

Initially, the areas of the predicted box \hat{A}^k and the ground truth box A^k can be defined as:

$$\hat{A}^k = (\hat{x}_{max}^k - \hat{x}_{min}^k) \times (\hat{y}_{max}^k - \hat{y}_{min}^k)$$

$$A^k = (x_{max}^k - x_{min}^k) \times (y_{max}^k - y_{min}^k)$$

Intuitively, the nonconformity score can be represented by the difference in areas:

$$R_{box}^k = \hat{A}^k - A^k$$

Similar to the previous method, to make the area difference robust to bounding boxes of different sizes, we apply scaling on it. Specifically, we normalize the area difference by the area of the predicted box, defining the final nonconformity score function as Equation (4):

$$R_{box}^k = \frac{\hat{A}^k - A^k}{\hat{A}^k} \quad (4)$$

3.1.3 Image -Wise

After implementing and experimenting with the box-wise *coordinate-difference* method, de Grancey et al. found two issues: First, the model failed to detect certain pedestrians (false positives), resulting in no predicted bounding boxes and thus no subsequent conformalization process; this reveals a lack of control mechanisms for such samples. Second, they observed variability in the coverage rate of the box-wise coordinate-difference method (coverage rate: the proportion of conformalized boxes that completely cover the true boxes out of all true boxes in the test set, which will be elaborated in Chapter 4 when discussing evaluation metrics). In specific images, the proportion of correctly covered true bounding boxes could be significantly lower than $1 - \alpha$, because $1 - \alpha$ is the average coverage rate of all conformalized boxes over the predicted boxes in the test set.

To address these issues, de Grancey et al. proposed a new method that guarantees coverage at the image level rather than at the individual object level. The specific goal is that during inference, at least $1 - \alpha$ of the images should have $1 - \beta$ of their true bounding boxes correctly covered by the conformalized boxes. If $\alpha = 0.1$, $\beta = 0.25$ are set, it means that at least 75% of the true boxes in at least 90% of the images in the dataset need to be covered by the conformalized boxes.

Thus, the nonconformity score $s_\beta(\hat{B}, B)$ is calculated by comparing the entire set of predicted bounding boxes \hat{B} and the set of all true boxes B in each image. The score is defined by finding a minimal boundary $r \geq 0$ that, when added to the four coordinate values of all predicted boxes, ensures that at least $1 - \beta$ of the true boxes are correctly covered by the conformalized boxes.

3.2 Calculating the Conformal Quantile

Following the general algorithm workflow for conformal prediction mentioned in section 2.2, after the nonconformity score functions have been computed for data points in the calibration set, we obtain an array $R_{all} = \{R^1, \dots, R^k\}$, that includes the "deviations" between all predicted and actual bounding boxes. Upon sorting this array, we can determine the conformal quantile:

(1) For the *Coordinate-Wise* nonconformity score, defined as Equation (5):

$$q_{1-\alpha} = \lceil (n_c + 1)(1 - \alpha) \rceil - th \text{ element of } R_{all} \quad (5)$$

Here, quantiles are calculated separately for each of the four differences mentioned in equation (1), where n_c represents the number of coordinate values of bounding boxes in the calibration set.

(2) For the box-wise nonconformity score without Bonferroni correction (Methods include *Coordinate-Difference w/o Bonferroni*, *Coordinate-Adaptation w/o Bonferroni*,

Area-Difference):

$$q_{1-\alpha} = \lceil (n_{c/box} + 1)(1 - \alpha) \rceil - th \text{ element of } R_{all} \quad (6)$$

(3) For box-wise methods with Bonferroni correction applied (*Coordinate-Difference with Bonferroni*, *Coordinate-Adaptation with Bonferroni*):

$$q_{1-\frac{\alpha}{4}} = \lceil (n_{box} + 1) \left(1 - \frac{\alpha}{4}\right) \rceil - th \text{ element of } R_{all}^c, c \in \{x_{min}, y_{min}, x_{max}, y_{max}\} \quad (7)$$

In these calculations, α is the pre-set error rate, and $1 - \alpha$ or $1 - \frac{\alpha}{4}$ is the desired

coverage rate. n_{box} represents the number of bounding boxes in the calibration set.

(4) The computation of the quantile and the generation of the prediction set for the Image-Wise method are integrated into a unified algorithm, which will be thoroughly explained at the end of section 3.3.

The quantile calculated in the this chapter can essentially be referred to as the margin[38] applied to the predicted boxes. In the algorithm for generating conformalized boxes within the conformal prediction framework, it is evident that conformalized boxes are exclusively derived from predicted boxes, implying a one-to-one correspondence. If the model fails to successfully detect an object, it is almost impossible for the conformalized boxes to cover the actual object, unless this object overlaps with another object that has a predicted box. The image-wise method mentioned at the end of next chapter can, to a certain extent, enable conformalized boxes to cover objects not recognized by the model.

3.3 Generating Prediction Sets

In the algorithm workflow described in section 2.2, this step is known as generating a prediction set that conforms to a coverage rate of $1 - \alpha$. In the context of this paper's work, addressing the spatial uncertainty of bounding boxes, this essentially involves generating conformalized bounding boxes based on the previously calculated quantile from the predicted boxes, thus ensuring a larger bounding box that intuitively provides a better guarantee of complete coverage.

Coordinate-wise: For each new coordinate in $x_{min}, y_{min}, x_{max}, y_{max}$, given the model's predicted coordinates $\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max}$, the prediction set is defined as:

$$\begin{aligned} \hat{C}_\alpha(x_{min}) &= [\hat{x}_{min} - q_\alpha^{x_{min}}, +\infty) & \hat{C}_\alpha(y_{min}) &= [\hat{y}_{min} - q_\alpha^{y_{min}}, +\infty) \\ \hat{C}_\alpha(x_{max}) &= (-\infty, \hat{x}_{max} + q_\alpha^{x_{max}}] & \hat{C}_\alpha(y_{max}) &= (-\infty, \hat{y}_{max} + q_\alpha^{y_{max}}] \end{aligned} \quad (8)$$

Box-wise: For new data points X_{n+1} , the model predicts the coordinates

$$(\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max})$$

For *Coordinate-Difference w/o Bonferroni Method*:

$$\hat{C}_\alpha(X_{n+1}) = \{\hat{x}_{min} - q_\alpha^{x_{min}}, \hat{y}_{min} - q_\alpha^{y_{min}}, \hat{x}_{max} + q_\alpha^{x_{max}}, \hat{y}_{max} + q_\alpha^{y_{max}}\} \quad (9)$$

For *Coordinate-Difference with Bonferroni Method*:

$$\hat{C}_\alpha(X_{n+1}) = \left\{ \hat{x}_{min} - q_{1-\frac{\alpha}{4}}^{x_{min}}, \hat{y}_{min} - q_{1-\frac{\alpha}{4}}^{y_{min}}, \hat{x}_{max} + q_{1-\frac{\alpha}{4}}^{x_{max}}, \hat{y}_{max} + q_{1-\frac{\alpha}{4}}^{y_{max}} \right\} \quad (10)$$

These two box-wise generated conformalized boxes are generally as shown in Figure 7.a., The black box in the middle represents the Prediction box of the model output, the outer colored box is the conformalized box, and the arrows represent the quantiles calculated in the previous section, i.e. the applied margins.

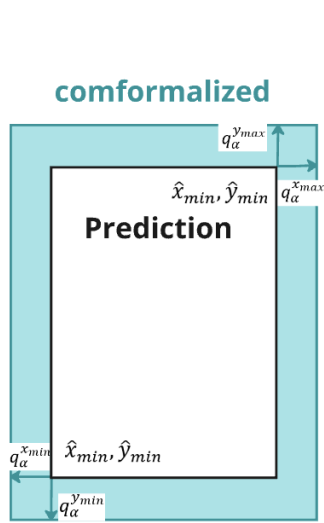
For *Coordinate-Adaptation Method*:

$$\hat{C}_\alpha(X_{n+1}) = \left\{ \begin{aligned} &\hat{x}_{min} - \hat{w} \cdot q_{1-\frac{\alpha}{4}}^{x_{min}}, \hat{y}_{min} - \hat{h} \cdot q_{1-\frac{\alpha}{4}}^{y_{min}}, \\ &\hat{x}_{max} + \hat{w} \cdot q_{1-\frac{\alpha}{4}}^{x_{max}}, \hat{y}_{max} + \hat{h} \cdot q_{1-\frac{\alpha}{4}}^{y_{max}} \end{aligned} \right\} \quad (11)$$

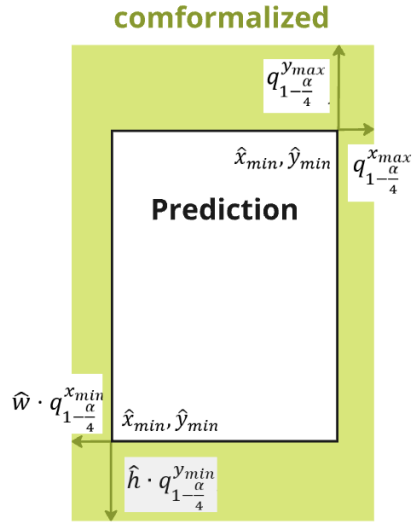
It's noteworthy that, to correspond with Equation (3), the generated prediction set, or bounding box, adjusts itself based on the width and height of its own predicted box, as depicted in Figure 7.b.

(a) Coordinate-Difference

(b) Coordinate-Adaptation



(c) Area-Difference



(d) Binary-Search

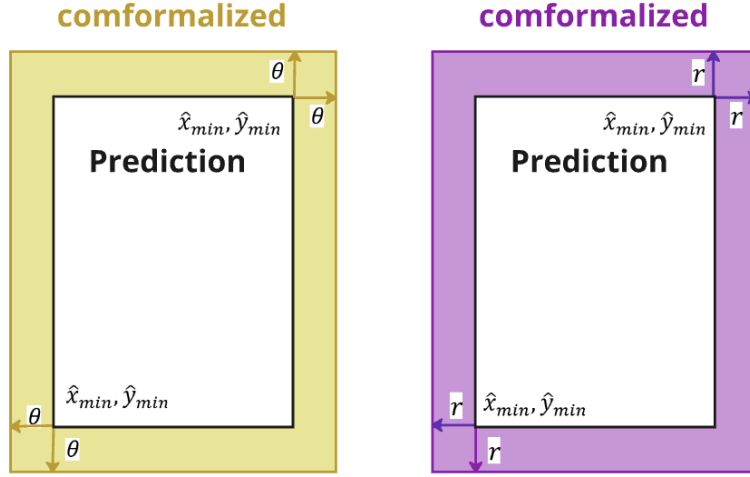


Figure 7 Examples of conformalized boxes generated by different conformalization methods

For *Area-Difference Method*, there is no need to use Bonferroni correction due to joint considerations for four coordinates. For new data points X_{n+1} , the model predicts $(\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max})$. In the previous chapter, when calculating the conformal quantile, $q_{1-\alpha}$ is actually a specific area difference. My design method is to distribute this difference evenly to the predicted bounding box coordinates $(\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max})$. This involves finding a margin θ , such that the difference in area between the conformalized boxes and the predicted boxes' area \hat{A} , divided by the area \hat{A} used in the quantile calculation, ultimately equals $q_{1-\alpha}$, as shown in Figure 7.c.

The area of the conformalized boxes can be represented by $(\hat{x}_{max} - \hat{x}_{min} + 2\theta)(\hat{y}_{max} - \hat{y}_{min} + 2\theta)$. Similarly, to correspond with Equation (4), in the error calculation, we also divide by \hat{A} to make the values equal. Therefore, the calculation formula can be expressed as:

$$\frac{[(\hat{x}_{max} - \hat{x}_{min} + 2\theta)(\hat{y}_{max} - \hat{y}_{min} + 2\theta) - \hat{A}]}{\hat{A}} = q_{1-\alpha}$$

From this, we solve for the margin θ , which can be computed with an efficiency of $O(1)$. Ultimately, the prediction set can be defined as:

$$\hat{C}_\alpha(X_{n+1}) = \{\hat{x}_{min} - \theta, \hat{y}_{min} - \theta, \hat{x}_{max} + \theta, \hat{y}_{max} + \theta\} \quad (12)$$

Image-Wise: For the image-wise conformalization method, since de Grancey only briefly mentioned the nonconformity score function without providing any formulas or experimental procedures, the following section aims to supplement these theoretical details. Specifically, it addresses how to find the smallest boundary $r \geq 0$ such that in the dataset, for $1 - \alpha$ of the images, at least $1 - \beta$ of the true boxes are correctly covered by the conformalized box expanded by the margin r , as is shown in Figure 7.d.

I propose Algorithm 2: Binary-search Image-wise Conformal Prediction as follows:

Algorithm 2 *Binary-search Image-wise Conformal Prediction*

- 1.**Input:** Image dataset $D \in X \times Y$, prediction algorithm A , tolerated miscoverage $\alpha \in (0,1)$, and non-coverage rate per image $\beta \in (0,1)$.
 - 2.**Output:** Prediction set $\hat{C}(X_{n+1})$ for each test image $(X_{n+1}, Y_{n+1}) \in D_{test}$, ensuring that at least $1 - \beta$ of the true boxes are covered with a margin $r \geq 0$ for at least $1 - \alpha$ proportion of the images.
 - 3.**Procedure:**
 4. Divide the dataset D into two non-intersecting subsets: a calibration set D_{cal} and a test set D_{test} .
 5. **Initialize** binary search parameters:
 $left = 0$
 $right = 1000$
Maximum iterations N for binary search
 6. **While** $N > 0$:
 $\lambda = (left + right)/2$ (midpoint of current left and right)
 Generate conformalized boxes for each X_{n+1} in D_{cal} :
 $\hat{C}(X_{n+1}) = \{\hat{x}_{min} - \lambda, \hat{y}_{min} - \lambda, \hat{x}_{max} + \lambda, \hat{y}_{max} + \lambda\}$
 Compute the coverage: Evaluate the proportion of images where at least $1 - \beta$ of the true boxes are covered by $\hat{C}(X_{n+1})$
 If this proportion $\geq 1 - \alpha$, set $right = \lambda$ (reduce λ to decrease coverage).
 Else, set $left = \lambda$ (increase λ to enhance coverage)
 $N = N - 1$
 7. **Finalize** λ as the optimal margin r after N iterations.
 8. For a new input image $(X_{n+1}, Y_{n+1}) \in D_{test}$, output the prediction set:
 $\hat{C}(X_{n+1}) = \{\hat{x}_{min} - r, \hat{y}_{min} - r, \hat{x}_{max} + r, \hat{y}_{max} + r\}$
-

Compared to Algorithm 1, the Binary-search Image-wise Conformal Prediction omits the training part since we use a pretrained YOLOv3. Additionally, we abstract the nonconformity score function $s_\beta(\hat{B}, B)$ and the process of finding the quantile into a binary search, updating boundaries, and seeking the minimal boundary r . Given that the algorithm strictly ensures coverage rates, this method can be considered a variant of conformal prediction (while also guaranteeing coverage rates).

Actually, this algorithm is a procedure involves initially setting a conformalized boxes $\hat{C}(X_{n+1}) = \{\hat{x}_{min} - \lambda, \hat{y}_{min} - \lambda, \hat{x}_{max} + \lambda, \hat{y}_{max} + \lambda\}$ and then iteratively adjusting this bounding box to continually test the coverage rate.

3.4 Evaluation Metrics

To assess and compare the performance of various conformal techniques, we employ two types of metrics in our experiments. The first metric, *Observed Coverage*, follows the one used by de Grancey et al., and measures the proportion of new conformalized bounding boxes that completely cover the actual ground truth bounding boxes across the entire test set. Specifically, if there are n images in the dataset with $n_{truebox}$ true bounding boxes, for each image data point X_i , $i = 1, \dots, n$, Y_i represents the true boxes for the i -th image and $\hat{C}(X_i)$ are conformalized boxes, the formula is as follows:

$$observed\ coverage = \frac{\sum_{i=1}^n Y_i \in \hat{C}(X_i)}{n_{truebox}}$$

This *Observed Coverage* is closely related to the pre-set conformal parameter, the error rate α (e.g., we might set $\alpha = 0.05$, which means we expect 95% of the true targets to be correctly covered by the conformalized bounding boxes). For this metric, we aim for *Observed Coverage* to closely approximate the preset $1 - \alpha$. If the *Observed Coverage* is below $1 - \alpha$, it clearly does not meet our aim of strict probabilistic coverage. If the *Observed Coverage* exceeds $1 - \alpha$, while it does ensure strict coverage, there is a possibility that the new conformalized boxes are significantly larger than the original predicted boxes, which could increase the coverage rate but contradict our goal of precisely locating the actual objects. Therefore, we hope for the conformalized boxes to fit the original model's predicted boxes as closely as possible while ensuring a certain *Observed Coverage* rate. To address this, I propose a new evaluation metric called “*Expansion*”.

The *Expansion* metric can be defined as the difference in area between the conformalized boxes (generated after conformal prediction) and the predicted boxes, divided by the area of the predicted boxes. Theoretically, this ratio ranges from 0 to infinity but is limited by the size of the image (the largest conformalized box cannot exceed the image itself). It intuitively represents the size relationship between the conformalized boxes and the predicted boxes. Assuming the dataset contains n images, for each image data point X_i , $i = 1, \dots, n$. with $j = 1, \dots, n_{box}$ predicted and corresponding conformalized boxes, then *Expansion* can be defined as:

$$E = \frac{\sum_{i=1}^n \sum_j^{n_{box}} A(C(X_i))_j - \sum_{i=1}^n \sum_j^{n_{box}} A(f(X_i))_j}{\sum_{i=1}^n \sum_j^{n_{box}} A(f(X_i))_j}$$

Where $\sum_j^{n_{box}} A(C(X_i))_j$ represents the total area of all conformalized boxes in the i -th image, and $\sum_{i=1}^n \sum_j^{n_{box}} A(f(X_i))_j$ represents the total area of all predicted boxes in the i -th image. By calculating E , we can quantify the relative expansion of each

conformalization method. If the value is large, it indicates that the conformalized boxes are much larger than the true boxes, suggesting a trade-off of precision for probabilistic assurance, which is not desirable; if E is close to 0, it indicates that the conformalized boxes match or are more compact to the true boxes, which would be a good performance on the Expansion metric.

It's worth noting that both *Expansion* and *Coverage* are calculated as "average expansion" and "average coverage," for instance, while some images may have many true boxes not covered by conformalized boxes, the *Coverage* can be compensated by the number of successful covers in other images.

4 Experiment

The previous chapter provided a detailed explanation of the algorithmic process of applying conformal prediction to predicted boxes and introduced the evaluation metrics *Coverage* and *Expansion*. In this chapter, we will specifically explain how to preprocess and construct the dataset, fine-tune the model on this dataset, design experiments based on research questions, and analyze and evaluate our results using both coverage and expansion metrics, ultimately addressing these research questions.

4.1 Research Questions

We will explore five research questions (RQs), as follows:

RQ1: What impact does a fine-tuned model have on the results of conformal prediction compared to a pretrained model?

RQ2: Does *Coordinate-Adaptation* method provide better coverage and more accurately fit conformalized boxes to predicted boxes compared to the baseline *Coordinate-Difference* method?

RQ3: Can the *Area-difference* method designed in this paper capture the interdependence of coordinates, ensure box-wise coverage, and potentially replace the Bonferroni correction method used in the baseline?

RQ4: How do the *Observed Coverage* and *Expansion* performance of the *Image-wise Binary-Search* method designed in this paper measure up?

RQ5: Can *Expansion* metric effectively compare different nonconformity score functions that generate conformalized boxes? What is the relationship between this metric and another metric, *Observed Coverage*?

4.2 Experimental Setup

To facilitate comparisons with the experimental results of de Grancey et al., the choice and processing of the dataset were made consistent. Our baseline model is YOLOv3, pretrained on the COCO training dataset. Our actual dataset is BDD100K, with its training set considered as our calibration set D_{cal} , and its validation set as our test set D_{test} .

To simplify the task, we filtered out other categories from BDD100K, retaining only those containing "person" and "rider". Consequently, the calibration and test sets were reduced to 22,213 and 3,220 images, respectively. Given that the original study use only a pretrained YOLO model, to explore whether the model's inherent performance impacts the final coverage rate of conformal prediction, we fine-tuned the YOLOv3 model across all layers for 100 epochs at a learning rate of 0.001.

We set the objectness threshold of the YOLOv3 to 0.3 and the Intersection over Union (IoU) threshold to 0.3. Through experimentation, the Average Precision (AP) of

the pretrained YOLOv3 model and the fine-tuned model were 0.33 and 0.38, respectively. Since an image may contain many objects, referring to [2], we use the Hungarian algorithm to match the predicted bounding box of the model output with the actual bounding box of the object based on $\text{IoU} = 0.3$.

On the selection of conformal prediction parameters, in order to compare with de Grancey's baseline work, three values were chosen for the $1 - \alpha$ coverage rate: 0.7, 0.9, and 0.95. Both pretrained and fine-tuned models were used to replicate and experiment with all conformalization methods discussed in Chapter 3, from Coordinate-Wise to Box-Wise, and finally to Image-Wise. Each method was applied on the calibration set D_{cal} of BDD100K using the nonconformity score function to compute errors, sort them, and calculate quantiles. Then, inference was performed on the unseen test set D_{test} of BDD100K, and the quantiles previously calculated were applied to the model's predicted boxes. Subsequently, the Coverage and Expansion metrics from Section 3.4 were used to evaluate the different conformalization methods. (Detailed design processes and computational methods are thoroughly explained in Chapter 3.)

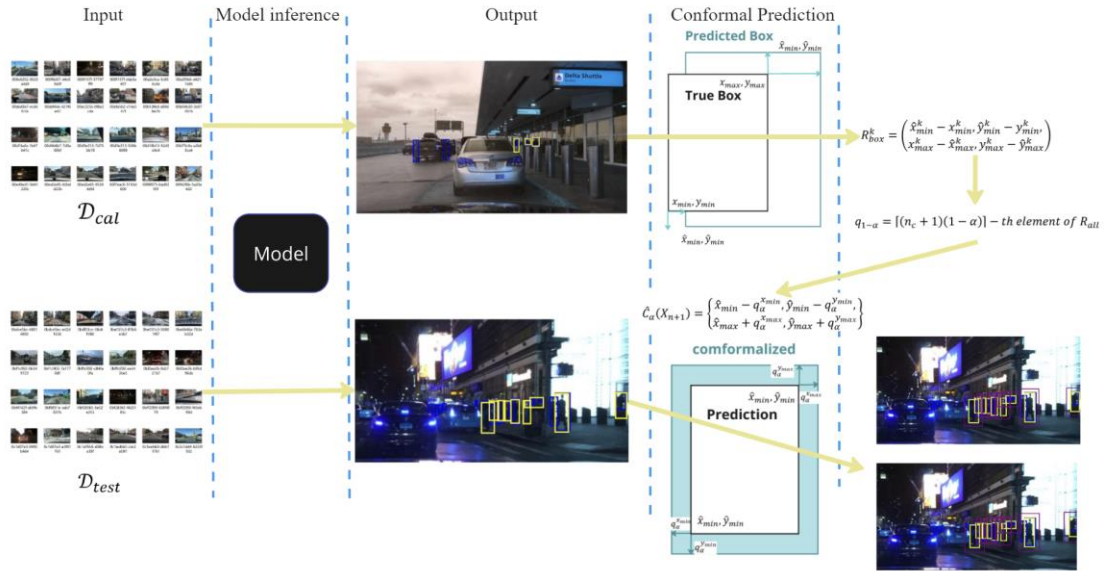


Figure 8 the workflow of experiment

After setting up the experiment, the next step is to initiate the experimental process. The general experimental workflow, as illustrated in the Figure8, involves dividing the BDD100K dataset into a calibration set and a test set. The steps are as follows:

1. First, data from the calibration set is processed through the YOLO model, resulting in model-predicted bounding boxes (yellow) and corresponding matched ground truth boxes (blue).
2. Next, by employing a designed nonconformity score function, we compute the discrepancies between the predicted bounding boxes and the actual ground truth boxes.
3. These discrepancies are then sorted, and based on the coverage requirement, a specific error, our quantile, is selected.

4. Using this quantile, we determine the margins for generating conformalized boxes according to different conformalization methods.

5. The test set images are then input into the model for inference, similarly yielding predicted bounding boxes (yellow).

6. The method for creating conformalized boxes, determined in step four, is applied to the current predicted bounding boxes to generate conformalized boxes.

7. Performance Metrics Calculation: Finally, the coverage and Expansion metrics for different methods and models are calculated on the test set.

4.3 Results

The experimental results regarding *Observed Coverage* on \mathcal{D}_{test} under specific coverage rates of 0.7, 0.9, and 0.95 for all conformalization methods are summarized in Table 1. For ease of comparison and to partially contrast with the same methods in de Grancey et al.'s baseline, their results are displayed in Table 2. In the tables, all values that did not meet the expected coverage standards are marked in *red*, such as 'Coordinate-difference without Bonferroni', which is abbreviated in the tables as 'Coordinate-difference w/o Bonferroni'.

For ease of discussion, if referencing results from the table in analysis, the format used would be '0.70/0.73(0.7)' indicating that under the specific coverage of 0.7, the Observed Coverage for the pretrained/fine-tuned models are 0.7 and 0.73 respectively.

It can be observed from the table below that there are two types of coverage rates: ***Specific Coverage***, which corresponds to the theoretical coverage rate associated with the predefined error rate α . This is the coverage rate we expect the model to achieve. ***Observed Coverage*** is the actual coverage rate obtained after calibration, and this metric can be influenced by many factors. For instance, variations in the conformalization method might introduce flaws, or the precision issues of the model itself, as mentioned in Section 3.2. Objects that are not recognized naturally do not have conformalized boxes to cover them, resulting in a reduced actual coverage rate.

Table 1 Evaluation of observed coverage on \mathcal{D}_{test}

Method (<i>this paper</i>)	Specific coverage ($1 - \alpha$)	0.7	0.9	0.95
		Observed Coverage (pretrained/fine-tuned)		
Coordinate-Wise	x_{min}	0.68/0.74	0.89/0.91	0.94/0.96
	x_{max}	0.71/0.74	0.93/0.93	0.95/0.96
	y_{min}	0.70/0.70	0.88/0.91	0.95/0.96
	y_{max}	0.70/0.73	0.92/0.93	0.90/0.95
Box-Wise	Coordinate-difference w/o Bonferroni	0.41/0.39	0.70/0.71	0.78/0.80
	Coordinate-difference with Bonferroni	0.75/0.76	0.92/0.91	0.96/0.96
	Coordinate-adaptation	0.45/0.55	0.81/0.80	0.89/0.90

	w/o Bonferroni			
	<i>Coordinate-adaptation</i> with Bonferroni	0.73/0.78	0.91/0.93	0.95/0.97
	<i>Area-difference</i>	0.55/0.63	0.80/0.83	0.88/0.93
Image-Wise	<i>Binary-search</i>	0.78/0.79	0.93/0.94	0.95/0.96

Table 2 Baseline work of de Grancey et al. Observed Coverage on \mathcal{D}_{test}

Method (<i>de Grancey</i>)	Specific coverage ($1 - \alpha$)	0.7	0.9	0.95
		Observed Coverage (Only Pretrained)		
Coordinate-Wise	x_{min}	0.76	0.91	0.96
	x_{max}	0.78	0.91	0.96
	y_{min}	0.70	0.92	0.95
	y_{max}	0.71	0.91	0.95
Box-Wise	<i>Coordinate-difference</i> w/o Bonferroni	0.35	0.73	0.86
	<i>Coordinate-difference</i> with Bonferroni	0.79	0.92	0.96

Given that the results in Table 1 indicate significant discrepancies for *Observed Coverage* and specific coverage on the *Coordinate-Difference* and *Coordinate-Adaptation* methods without Bonferroni correction (Coordinate-difference w/o Bonferroni 0.41/0.39 (0.7) 0.70/0.71 (0.9) 0.78/0.80 (0.95), Coordinate-adaptation w/o Bonferroni 0.45/0.55 (0.7) 0.81/0.80 (0.9) 0.89/0.90 (0.95)), it is evident that methods without coverage assurance, do not meet the *Specific Coverage* requirements. Such methods that do **not** satisfy the coverage requirements are **bolded** in font in the table 3 below. Furthermore, since De Grancey's paper does not include the Expansion metric, we will compare methods amongst themselves. However, it's important to note that the Expansion metric, being specific to bounding boxes, cannot be applied to measure the coordinate-based methods. The final comparison of Expansion is therefore shown below:

Table 3 Evaluation of Expansion on \mathcal{D}_{test} for conformalization methods

Method	Specific coverage ($1 - \alpha$)	0.7	0.9	0.95
		Expansion (pretrained/fine-tuned)		
Box-Wise	<i>Coordinate-difference</i> w/o Bonferroni (<i>de Grancey</i>)	0.15/0.17	0.32/0.27	0.38/0.32
	<i>Coordinate-difference</i> with Bonferroni (<i>de Grancey</i>)	0.37/0.28	0.45/0.35	0.49/0.41

	<i>Coordinate-adaptation w/o Bonferroni (this paper)</i>	0.39/0.21	0.49/0.31	0.61/0.42
	<i>Coordinate-adaptation with Bonferroni (this paper)</i>	0.48/0.29	0.53/0.36	0.67/0.48
	<i>Area-difference (this paper)</i>	0.52/0.44	0.58/0.50	0.78/0.70
Image-Wise	<i>Binary-search (this paper)</i>	12.25/8.54	15.64/10.94	22.64/16.84

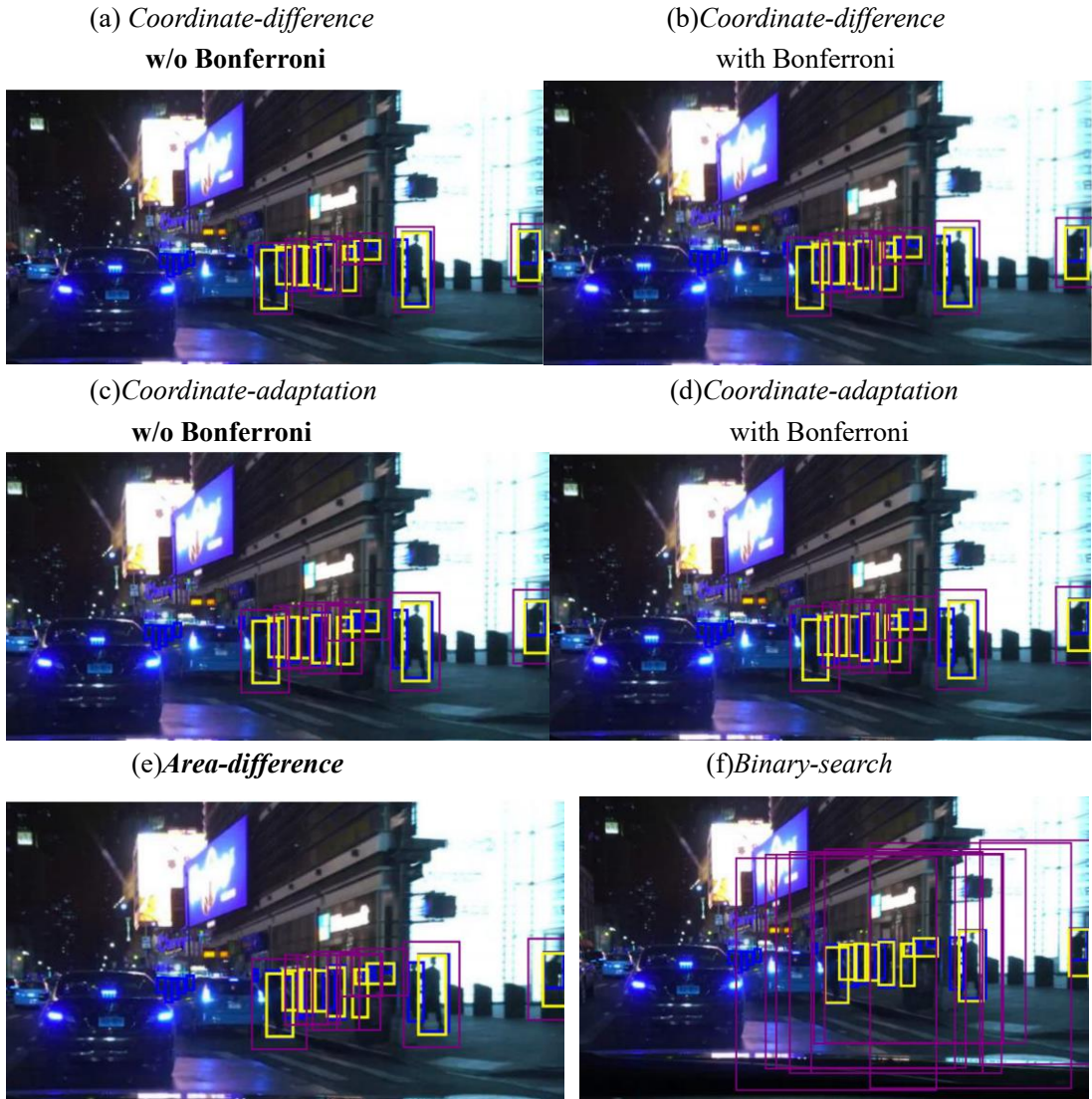


Figure 9 Images of conformalized box
(including 6 methods, risk level $\alpha = 0.1$, pretrained model) on a BDD100k image
with Ground Truth, Inference and Conformalized boxes.

4.4 Analysis for Research Question in Coverage and Expansion

RQ1: What impact does a fine-tuned model have on the results of conformal prediction compared to a pretrained model?

For *Observed Coverage*:

(1) Taking the methods *Coordinate-adaptation w/o Bonferroni* (0.45/0.55, 0.81/0.80, 0.89/0.90) and *Coordinate-adaptation with Bonferroni* (0.73/0.78, 0.91/0.93, 0.95/0.97) as examples: The fine-tuned models exhibit the same or higher observed coverage rates under the same preset coverage rates compared to the pretrained models. This is because the fine-tuned models have higher precision, capable of recognizing objects that the pretrained models could not, thereby producing more conformalized boxes during the conformal prediction process, which can cover more real boxes.

(2) For *Coordinate-adaptation w/o Bonferroni*, the experimental results show (0.45/0.55, 0.81/0.80, 0.89/0.90). At a 0.7 specific coverage rate, there is a 0.1 difference in coverage rates between the pretrained and fine-tuned models. However, at preset coverage rates of 0.9 and 0.95, the difference in coverage rates is negligible, and at 0.9, the coverage rate of the pretrained model is even higher by 0.01. This could be due to the conformalized boxes becoming larger as the coverage assurance increases, covering some objects that were not originally recognized, thus reducing the impact of model precision on the observed coverage rate.

For *Expansion*:

Considering *Coordinate-difference with Bonferroni* (de Grancey) (0.37/0.28, 0.45/0.35, 0.49/0.41), we observe that in most cases, when the model's precision increases, the required Expansion decreases, and the generated conformalized boxes fit more closely to the predicted boxes. This is explainable because the fine-tuned model can more precisely locate each object, resulting in predicted boxes that closely match the actual boxes. Consequently, the calculated error is smaller, meaning that the conformalized box's margin is smaller, naturally leading to less expansion.

RQ2: Does *Coordinate-Adaptation* method provide better coverage and more accurately fit conformalized boxes to predicted boxes compared to the baseline *Coordinate-Difference* method?

For *Observed Coverage*:

The *Coordinate-Adaptation* method was developed to optimize the process of nonconformity score function and predicted boxes generated in the *Coordinate-Difference* method by incorporating the width and height of the predicted boxes, with the aim of providing better attention to smaller objects and thus generating more closely fitting predicted boxes.

Comparing experimental results in the tables, the *Coordinate-Difference w/o Bonferroni* results were 0.41/0.39, 0.70/0.71, 0.78/0.80, whereas the *Coordinate-Adaptation w/o Bonferroni* were 0.45/0.55, 0.81/0.80, 0.89/0.90. Without Bonferroni correction, neither method met the expected probabilistic guarantees, but *Coordinate-*

Adaptation exhibited an average increase in coverage of about 0.1, showing better performance. However, with Bonferroni correction applied, the results for *Coordinate-Difference* (0.75/0.76, 0.92/0.91, 0.96/0.96) and *Coordinate-Adaptation* (0.73/0.78, 0.91/0.93, 0.95/0.97) both ensuring strict probabilistic coverage with almost no differences.

For *Expansion*:

Firstly, it's important to note that comparing the *Expansion* of the two methods is necessary upon their coverage being meeting the requirement. If the *Expansion* is very low and conforms more closely to the predicted boxes, but the coverage rate is significantly below the requirement, the comparison is meaningless. Therefore, we only compare the two methods with Bonferroni correction applied. According to Table 3, the results for *Coordinate-Difference* with Bonferroni (de Grancey) are 0.37/0.28, 0.45/0.35, 0.49/0.41, whereas for *Coordinate-Adaptation* with Bonferroni (this paper) the results are 0.48/0.29, 0.53/0.36, 0.67/0.48. It is evident that at any preset coverage rate, *Coordinate-Difference* with Bonferroni (de Grancey) performs better, requiring only 0.45/0.35 *Expansion* at a 0.9 coverage rate, which closely aligns with the predicted boxes.

Of course, the *Expansion* for *Coordinate-Adaptation* is not significantly high, ranking second among all methods guaranteed by coverage, with only a 0.1-0.2 increase in expansion compared to *Coordinate-Difference*. Given that this involves area calculations, As figure 9.b and 9.d show, there is almost no difference between the conformalized boxes of the two methods. The conformalized boxes generated by *Coordinate-Adaptation* are only slightly larger.

RQ3: Can the *Area-difference* method designed in this paper capture the interdependence of coordinates, ensure box-wise coverage, and potentially replace the Bonferroni correction method used in the baseline?

For *Observed Coverage*:

In Section 3, the *Area-difference* method was designed to ensure box-level coverage by linking the four coordinates via area, rather than through Bonferroni correction. It is noticeable that, although this method did not meet the preset probabilistic requirements for both pretrained and fine-tuned models—with results being 0.55/0.63, 0.80/0.83, 0.88/0.93—it still performed closer to the coverage requirements compared to *Coordinate-difference w/o Bonferroni* (0.41/0.39, 0.70/0.71, 0.78/0.80) and *Coordinate-adaptation w/o Bonferroni* (0.45/0.55, 0.81/0.80, 0.89/0.90). This suggests that the *Area-difference* method can to some extent capture the interdependence of coordinates, though this connection is still not as strict as methods with Bonferroni correction applied.

For *Expansion*:

Results in the table for *Area-difference (this paper)* are 0.52/0.44, 0.58/0.50, 0.78/0.70. Given that *Area-difference* almost ensures coverage, switching to a higher precision model could potentially meet the coverage requirement, thus making

observations of its expansion meaningful. It is noted that while the *Expansion* is not exaggerated, it still exceeds 0.5 and does not achieve compactness. Among all methods that nearly meet or satisfy coverage requirements, it ranks third, ahead of the image-wise method. Observing Figure 9.e, Area-difference does succeed in covering most of the true box and have relatively compact conformalized boxes.

RQ4: How do the *Observed Coverage* and *Expansion* performance of the Image-wise Binary-Search method designed in this paper measure up?

For *Observed Coverage*:

Data from Table 1 shows that the *Image-Wise Binary-Search* achieves *Coverage* rates of 0.78/0.79, 0.93/0.94, and 0.95/0.96. Compared to other methods such as Coordinate-difference with Bonferroni (0.75/0.76, 0.92/0.91, 0.96/0.96) and Coordinate-adaptation with Bonferroni (0.73/0.78, 0.91/0.93, 0.95/0.97), it is evident that Binary-Search shows better performance in coverage, outweighing all other methods at the same preset coverage levels. Observing Figure 9.f, although the conformalized boxes are significantly expanded, they indeed cover objects that the model failed to recognize (the blue true boxes in the middle of figure 9), indirectly addressing a common issue with other methods that lack measures for unrecognized objects, thereby enhancing the actual coverage rate.

For *Expansion*:

The Image-Wise Binary-Search (de Grancey) displays *Expansion* of 12.25/8.54, 15.64/10.94, and 22.64/16.84. These results indicate a substantial expansion, nearly ten times the size of the predicted boxes, with the expansion increasing as the coverage requirement rises, expanding from 12 to 22. Even at the most relaxed coverage requirement of 0.7, the fine-tuned model exhibits nearly an 8-times *Expansion*. This suggests that to ensure coverage, some boxes must expand significantly.

Recalling the image-wise binary method, we have two parameters, α and β with settings such as $\alpha = 0.1$, and $\beta = 0.25$ in de Grancey's approach. This implies that for at least 90% of the images in the dataset, 75% of the true boxes need to be covered by the conformalized box. De Grancey et al. intended to address two issues left by previous methods: (1) Uncontrolled Frequencies: For pedestrians not detected by the model, there is no control over their occurrence rates. (2) Coverage Variability: As evidenced by previous tables, the proportion of true bounding boxes correctly covered in a specific image might be zero because the $1 - \alpha$ coverage rate is the average of all conformalized boxes covering predicted boxes across the test set.

Upon examining Figure 9.f, it can be seen that in the center of the image, there are also blue true boxes representing "pedestrians." Due to nighttime conditions and car headlights, or inherent limitations of our base model YOLO, these pedestrians were not detected. My image-wise method requires that 75% (assume $\beta = 0.25$) of the true boxes in this image must be covered. To meet this requirement, the conformalized box of the nearest detected pedestrian must be continuously expanded until it covers the knowledge frames of these few pedestrians, fulfilling the statistical demands for

coverage.

Therefore, the constraints posed by de Grancey's method are quite strict. Sometimes, to meet the pre-set *Coverage*, more true boxes must be included in an image. If there is a true box far from the predicted box at the image's edge, or if the model fails to detect an object entirely, other predicted boxes might need to expand a lot to cover this true box to satisfy the probability of covering true boxes. This situation causes the *Expansion* metric to explode, particularly in image-wise cases. Unlike box-wise, where coverage failures in difficult images can be compensated by other images as long as the average is maintained, image-wise strictly requires at least $1 - \alpha$ of images to correctly cover $1 - \beta$ of real boxes, leading to significant expansions. Furthermore, because fine-tuned models can detect more objects, they require less Expansion.

RQ5: Can *Expansion* metric effectively compare different nonconformity score functions that generate conformalized boxes? What is the relationship between this metric and another metric, *Observed Coverage*?

The *Expansion* metric serves as an effective indicator for measuring the size of conformalized boxes relative to predicted boxes. For example, using the methods of *Coordinate-difference* with Bonferroni (0.75/0.76, 0.92/0.91, 0.96/0.96) and *Coordinate-adaptation* with Bonferroni (0.73/0.78, 0.91/0.93, 0.95/0.97), both meet the predefined coverage requirements. And then observe figures 9.b and 9.d, it is challenging to determine which type of conformalized box performs better, as they are very close in performance. However, going through the results shown in Table 3, the *Coordinate-difference* with Bonferroni method (0.37/0.28, 0.45/0.35, 0.49/0.41), and the *Coordinate-adaptation* with Bonferroni, in this paper (0.48/0.29, 0.53/0.36, 0.67/0.48), showing that after calculating the expansion, slight differences are noticeable, with *Coordinate-difference with Bonferroni* generally having boxes that conform more closely to the predicted frames.

Furthermore, for methods with excessive *Expansion*, such as *Image-wise Binary-search* Method (12.25/8.54, 15.64/10.94, 22.64/16.84), we can intuitively perceive the changes in numbers, providing a good reference for subsequent analysis.

Regarding the connection between the metrics of *Expansion* and *Observed Coverage*, the former complements the latter. For a method that does not meet the specific coverage requirements, such as *Coordinate-difference without Bonferroni* (0.15/0.17, 0.32/0.27, 0.38/0.32), even if its conformalized boxes align very well with the predicted boxes, it is meaningless since ensuring specific *Coverage* is a critical aspect that distinguishes conformal prediction from other uncertainty quantification methods.

Therefore, evaluating the performance of *Expansion* requires prior consideration of the *Observed Coverage*. Moreover, as shown in table 3 with the example of *Coordinate-difference with Bonferroni* (0.37/0.28, 0.45/0.35, 0.49/0.41), as the specific coverage requirements increase, *Expansion* shows a positive correlation with increasing specific *Coverage*. This happens because as we impose stricter conditions, without changing the model accuracy, to cover more predicted boxes, the conformalized boxes must be

enlarged. This can also be explained by the quantile formula derived in chapter three; if the coverage requirement increases from 90% to 95%, the quantile we seek moves from the 90th percentile error to the 95th percentile error. Since errors are sorted in the list, the errors increase towards the end, hence the larger the margin applied to the conformalized boxes.

Beyond these research questions (RQ), we have additional findings:

Finding 1: The necessity of Bonferroni correction in the baseline.

By comparing the experimental results of this paper on *Coordinate-difference*, along with baseline experiments on *Coordinate-difference* without Bonferroni correction, it is evident that under specific coverage requirements of 0.7, 0.9, and 0.95, the *Coordinate-difference* method in my paper in a pretrained model setting only achieves *Observed Coverage* rates of 0.41, 0.70, and 0.78 respectively. In a fine-tuned model, the *Observed Coverage* are only slightly better at 0.39, 0.71, and 0.80, similar to the baseline results of 0.35, 0.73, and 0.86. This reconfirms the issue identified in the baseline: in the box-wise *Coordinate-difference* method, when considering each of the four coordinates independently, the actual coverage rate is very low. Therefore, we need Bonferroni correction to evenly distribute the original error rate limit α , across the four coordinates, allocating $\alpha/4$ to each, ultimately ensuring compliance with the specific coverage requirements.

Finding 2: If we compare the overall *Observed Coverage* of the same method in my approach to the baseline: Under the same method, the *Observed Coverage* shows that my pretrained model performs worst, while my fine-tuned model performs comparably or better than the baseline. The detailed analysis is as follows.

When comparing the results from Table 1 with those in Table 2 for the same conformalization techniques, such as the *Coordinate-wise method* at the x_{min} row, my results are (0.68/0.74, 0.89/0.91, 0.94/0.96) compared to the baseline results of 0.76, 0.91, 0.96. We observe that the *Observed Coverage* on my pretrained model is lower than the baseline, while the fine-tuned model's *Observed Coverage* is close to or matches the baseline.

Similarly, for the '*Coordinate-difference with Bonferroni*' method, my results are (0.73/0.78 0.91/0.93 0.95/0.97), whereas the baseline shows 0.79, 0.92, 0.96. Here, the *Observed Coverage* of the pretrained model is lower than the baseline, but after fine-tuning, the model's performance is close to or better than the baseline. Considering all data from the tables, in terms of *Observed Coverage*, pretrained model in this paper is worse than the one in baseline's work, and the *Observed Coverage* of the fine-tuned model is close to or higher than the baseline.

Several reasons could account for these phenomena:

First, the baseline paper does not mention how their pretrained YOLOv3 model on COCO performed on BDD100K. Hence, I suppose that the pretrained YOLOv3 used in my study is less accurate than the baseline. This would explain why it fails to recognize some very small 'people', or 'people' in dark conditions, or 'people' whose colors closely match the background. These challenging recognition conditions can lead

to a failure to detect certain categories of 'people' with the pretrained on COCO model (as also evidenced by the lower AP of the pretrained model in my experiments). Since the model does not recognize these, naturally, no predicted boxes are generated, and thus, no conformal prediction process can be conducted. Consequently, these true boxes are entirely missed by the model, leading to a decreased coverage rate. However, after fine-tuning, the model's ability to detect previously unrecognized objects is enhanced, slightly increasing the coverage rate to meet the preset coverage requirements.

This point reconfirms the conclusion from RQ1: the precision of the model significantly impacts the coverage rate, where a less accurate model results in fewer recognized objects, thereby reducing the coverage rate of the actual objects.

4.5 Comprehensive Analysis

This paper builds on the "Pedestrian Detection" research by de Grancey et al., attempting to apply conformal prediction to the YOLOv3 model's predicted bounding boxes to ensure that the newly generated conformalized bounding boxes cover the actual bounding boxes with a preset probability. The study used both fine-tuned and pretrained models and explored the effects and performance of different conformal prediction methods based on five research questions (RQ1-RQ5).

From **RQ1**: The impact of pretrained/fine-tuned models on conformal prediction, it is evident that the fine-tuned model demonstrates a higher *Observed Coverage* at the same preset coverage rates, attributed to its higher precision and recognition rate, which can detect objects that the pretrained model failed to identify, thus generating more conformalized bounding boxes. Moreover, as the model's precision increases, the required *Expansion* decreases, making the conformalized bounding boxes more closely fit the predicted boxes. Furthermore, it is observed that conformal prediction cannot replace the training or fine-tuning of object detection models; an average conformalization method using a fine-tuned model might even achieve better coverage and lower expansion than a more capable conformalization method using a pretrained model. This discrepancy will be larger as the precision of the model decreases, making it challenging for conformalization itself to compensate.

From **RQ2-RQ4**, all the methods proposed in this paper indicate:

The *Coordinate-Adaptation* method, designed to include the width and height of the prediction box in calculating errors and generating conformalized boxes, aims for better coverage and minimal expansion. This method almost matches Coordinate-Difference in terms of coverage assurance but has slightly higher expansion, yet it performs well and closely fits the predicted boxes.

The *Area-Difference* method, by considering all four coordinates together, attempts to replace the strict Bonferroni method of the original paper. Although it did not meet the preset specific coverage requirements, its performance significantly improved compared to uncorrected methods, indicating a degree of correlation found.

Image-wise Binary-Search method achieves the best *Observed Coverage* through significant expansion and can cover many objects that the original model failed to

recognize; this outcome is not desirable because the expansion of the conformalized boxes at this stage is too extensive, rendering them devoid of practical research value. The reason lies in the inherent flaws in the method designed by de Grancey et al., where overly strict probability guarantees result in conformalized boxes that are excessively large, aimed to cover more real object boxes than is pragmatically necessary.

From **RQ5: Comparison and correlation of the expansiveness metric**, it is known that generally, expansion increases with higher coverage requirements, but excessive expansion is of no practical research significance.

5 Conclusion

This paper explores the use of conformal prediction techniques in pedestrian detection tasks, ensuring that newly generated conformalized boxes can cover real objects with strict probabilistic guarantees. In addition to replicating the methods of de Grancey et al., this paper introduces novel approaches such as *Coordinate-Adaptation*, *Area-Difference*, and *Image-wise Binary-Search* to address issues mentioned in the baseline and introduces a new evaluation metric, *Expansion*, which effectively measures the performance of different conformalization methods in generating conformalized boxes. Among these, the *Coordinate-Adaptation* method demonstrates qualified effectiveness with lower *Expansion* while maintaining the predefined coverage rate. Although *Area-Difference* does not strictly meet the specific coverage rate, it indicates to some extent that area can capture the associations of bounding coordinates. *Image-wise Binary-Search* achieves the highest *Observed Coverage* but due to the overly strict design of de Grancey et al., it has very high *Expansion*, rendering it impractical for real-world applications.

The discrepancy between results from fine-tuned and pretrained models shows that model precision significantly impacts *Observed Coverage*. Conformal prediction cannot replace the model training and fine-tuning process; objects which are not recognized by model and hence without predicted bounding boxes. It is difficult to cover unrecognized objects using traditional conformal prediction methods, leading to reduced coverage. Improved score functions can only slightly enhance coverage and model accuracy is the decisive factor.

For objects that are not detected, how to cover them with conformalized boxes remains a challenge. The *image-wise Binary-Search* method introduced in this paper serves as an exploratory attempt. Although the method's *Expansion* is excessively high, it successfully identifies undetected objects. Future research could explore relaxing the overly strict probabilistic requirements of de Grancey et al. to reduce the method's *Expansion*.

Furthermore, designing the nonconformity score function is absolutely a core task to conformal prediction, as it pertains to the handling of all known information. Current methods are designed from an intuitive graphical perspective; hence, future research could explore deriving this function from internal model outputs or incorporating probabilistic computations.

This paper only selected coverage parameters $(1 - \alpha)$ of 0.7, 0.9, and 0.95 for computation, with an IoU threshold of 0.3 for object detectors. These parameter choices may appear limited, thus future researchers could systematically experiment with different parameter settings.

Reference

- [1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, ‘Object Detection in 20 Years: A Survey’, *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, 2023, doi: 10.1109/JPROC.2023.3238524.
- [2] F. de Grancey, J.-L. Adam, L. Alecu, S. Gerchinovitz, F. Mamalet, and D. Vigouroux, ‘Object Detection with Probabilistic Guarantees: A Conformal Prediction Approach’, in *Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops*, M. Trapp, E. Schoitsch, J. Guiochet, and F. Bitsch, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 316–329. doi: 10.1007/978-3-031-14862-0_23.
- [3] Y. Gal, ‘Uncertainty in deep learning’, 2016.
- [4] Y. Gal and Z. Ghahramani, ‘Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning’, in *Proceedings of The 33rd International Conference on Machine Learning*, PMLR, Jun. 2016, pp. 1050–1059. Accessed: Feb. 23, 2024. [Online]. Available: <https://proceedings.mlr.press/v48/gal16.html>
- [5] Y. Gal, R. Islam, and Z. Ghahramani, ‘Deep Bayesian Active Learning with Image Data’, in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 1183–1192. Accessed: Feb. 23, 2024. [Online]. Available: <https://proceedings.mlr.press/v70/gal17a.html>
- [6] R. Hu, Q. Huang, S. Chang, H. Wang, and J. He, ‘The MBPEP: a deep ensemble pruning algorithm providing high quality uncertainty prediction’, *Appl. Intell.*, vol. 49, no. 8, pp. 2942–2955, Aug. 2019, doi: 10.1007/s10489-019-01421-8.
- [7] D. Hall *et al.*, ‘Probabilistic Object Detection: Definition and Evaluation’, presented at the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1031–1040. Accessed: Feb. 23, 2024. [Online]. Available: https://openaccess.thecvf.com/content_WACV_2020/html/Hall_Probabilistic_Object_Detection_Definition_and_Evaluation_WACV_2020_paper.html
- [8] R. J. Tibshirani, R. F. Barber, E. J. Candes, and A. Ramdas, ‘Conformal Prediction Under Covariate Shift’, Cornell University Library, arXiv.org, Ithaca, 2020. doi: 10.48550/arxiv.1904.06019.
- [9] A. N. Angelopoulos and S. Bates, ‘A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification’, Cornell University Library, arXiv.org, Ithaca, 2022. doi: 10.48550/arxiv.2107.07511.
- [10] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, ‘Object Detection With Deep Learning: A Review’, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [11] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, ‘Object Detection in 20 Years: A Survey’, *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023, doi: 10.1109/JPROC.2023.3238524.
- [12] L. Jiao *et al.*, ‘A Survey of Deep Learning-Based Object Detection’, *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

- [13]P. Viola and M. Jones, ‘Rapid object detection using a boosted cascade of simple features’, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Dec. 2001, p. I–I. doi: 10.1109/CVPR.2001.990517.
- [14]N. Dalal and B. Triggs, ‘Histograms of oriented gradients for human detection’, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, Jun. 2005, pp. 886–893 vol. 1. doi: 10.1109/CVPR.2005.177.
- [15]P. Felzenszwalb, D. McAllester, and D. Ramanan, ‘A discriminatively trained, multiscale, deformable part model’, in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8. doi: 10.1109/CVPR.2008.4587597.
- [16]A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘ImageNet Classification with Deep Convolutional Neural Networks’, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Feb. 23, 2024. [Online]. Available: <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- [17]R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation’, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587. Accessed: Aug. 29, 2024. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html
- [18]R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘Region-Based Convolutional Networks for Accurate Object Detection and Segmentation’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.
- [19]K. He, X. Zhang, S. Ren, and J. Sun, ‘Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.
- [20]R. Girshick, ‘Fast R-CNN’, presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. Accessed: Feb. 23, 2024. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html
- [21]S. Ren, K. He, R. Girshick, and J. Sun, ‘Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks’, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2015. Accessed: Feb. 23, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

- [22]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘You Only Look Once: Unified, Real-Time Object Detection’, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788. Accessed: Feb. 23, 2024. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html
- [23]J. Redmon and A. Farhadi, ‘YOLOv3: An Incremental Improvement’, Apr. 08, 2018, *arXiv*: arXiv:1804.02767. doi: 10.48550/arXiv.1804.02767.
- [24]A. Bochkovskiy, W. Chien-Yao, and M. L. Hong-Yuan, ‘YOLOv4: Optimal Speed and Accuracy of Object Detection’, Cornell University Library, arXiv.org, Ithaca, 2020. doi: 10.48550/arxiv.2004.10934.
- [25]U. Nepal and H. Eslamiat, ‘Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs’, *Sensors*, vol. 22, no. 2, Art. no. 2, Jan. 2022, doi: 10.3390/s22020464.
- [26]C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, ‘YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors’, presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7464–7475. Accessed: Feb. 23, 2024. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023/html/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.html
- [27]A. Neubeck and L. Van Gool, ‘Efficient Non-Maximum Suppression’, in *18th International Conference on Pattern Recognition (ICPR’06)*, Aug. 2006, pp. 850–855. doi: 10.1109/ICPR.2006.479.
- [28]W. Liu *et al.*, ‘SSD: Single Shot MultiBox Detector’, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
- [29]T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, ‘Focal Loss for Dense Object Detection’, presented at the Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988. Accessed: Feb. 23, 2024. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html
- [30]H. Law and J. Deng, ‘CornerNet: Detecting Objects as Paired Keypoints’, presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 734–750. Accessed: Sep. 01, 2024. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Hei_Law_CornerNet_Detecting_Objects_ECCV_2018_paper.html
- [31]X. Zhou, D. Wang, and P. Krähenbühl, ‘Objects as Points’, Apr. 25, 2019, *arXiv*: arXiv:1904.07850. doi: 10.48550/arXiv.1904.07850.
- [32]N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, ‘End-

- to-End Object Detection with Transformers’, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 213–229. doi: 10.1007/978-3-030-58452-8_13.
- [33] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, ‘Deformable DETR: Deformable Transformers for End-to-End Object Detection’, Mar. 17, 2021, *arXiv:arXiv:2010.04159*. doi: 10.48550/arXiv.2010.04159.
- [34] L. Tsung-Yi *et al.*, ‘Microsoft COCO: Common Objects in Context’, Cornell University Library, arXiv.org, Ithaca, 2015. doi: 10.48550/arxiv.1405.0312.
- [35] F. Yu *et al.*, ‘BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning’, presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2636–2645. Accessed: Feb. 24, 2024. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Yu_BDD100K_A_Diverse_Driving_Dataset_for_Heterogeneous_Multitask_Learning_CVPR_2020_paper.html
- [36] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, ‘Conformal prediction beyond exchangeability’, Cornell University Library, arXiv.org, Ithaca, 2022. doi: 10.48550/arxiv.2202.13415.
- [37] J. M. Bland and D. G. Altman, ‘Multiple significance tests: the Bonferroni method’, *BMJ*, vol. 310, no. 6973, p. 170, Jan. 1995, doi: 10.1136/bmj.310.6973.170.
- [38] A. Timans, C.-N. Straehle, K. Sakmann, and E. Nalisnick, ‘Adaptive Bounding Box Uncertainties via Two-Step Conformal Prediction’, Mar. 11, 2024, *arXiv:arXiv:2403.07263*. doi: 10.48550/arXiv.2403.07263.

Appendices

Appendix A: Evidence of completing ALL required ethics training.

Evidence 1: Epigeum Training



Evidence 2: Information Security Smart Training

Course completed

Do not reply to this email (via Moodle@Warwick) <no-reply-moodle@warwick.ac.uk>

Thu 5/16/2024 7:55 PM

To: LU, YANTONG (PGT) <Yantong.Lu@warwick.ac.uk>

Congratulations!

You have completed the course [Information Security Smart](#).


Evidence 3: WMG Student Ethics Training

Congratulations! You just earned a badge!

moodle.warwick (via Moodle@Warwick) <no-reply-moodle@warwick.ac.uk>

Thu 5/16/2024 12:52 PM

To: LU, YANTONG (PGT) <Yantong.Lu@warwick.ac.uk>

 1 attachments (80 KB)

23-24_Research_Ethics.png;

You have been awarded the badge "23-24 Research Ethics"!

More information about this badge can be found on the [23-24 Research Ethics](#) badge information page.

You can manage and download the badge from your [Manage badges](#) page.

Appendix B: a confirmation email of ethical approval for project

2024/9/2 14:37

Mail - LU, YANTONG (PGT) - Outlook

Ethical approval is granted (WMG Taught Student Research)

Qualtrics Survey Software <noreply@qemailserver.com>

Mon 9/2/2024 10:38 AM

To: LU, YANTONG (PGT) <Yantong.Lu@warwick.ac.uk>

Date: **September 2, 2024**

Student: **YANTONG LU**

Student ID number: **5528854**

Project title: **Object Detection with Uncertainty Quantification**

Your ethical approval number is **WMG-R_204Hy1Z3xjjghWN**

Your supervisor **Xingyu Zhao** has **granted ethical approval for your project**.

This means you have consent to conduct your research.

You now have the appropriate approval in place to begin your data collection. It is advisable to note the actual dates of data collection in the final project submission to evidence that your data was collected after the date of ethical approval (as stated in this email).

You are reminded that you must now adhere to the answers and detail given in the completed ethics form. If anything changes in your research such that any of your answers change to the form for which you received **ethical approval** for, then you must contact your supervisor to check if you need to reapply for or update your **ethical approval** before you proceed with data collection.

When you submit your project, please write **your ethical approval number against the ethical approval field** on the cover page of the submission and include a copy of this email in the Appendices of the submission.

Kind regards,

WMG Projects Team

[Download as PDF](#)

2024/9/2 14:37

Mail - LU, YANTONG (PGT) - Outlook

<i>URL to view Results</i>	[Click Here]
---------------------------------------	------------------------------

Response Summary:

Supervisor Delegated Approval (SDA) Form for WMG Taught Student Projects

This form is NOT for supervisors of cyber security students

Important information:

- This form is for taught student projects ONLY. EngD/PhD/Postgraduate Research students must receive ethical approval from BSREC
- Approval can be delegated to taught student projects that are LOW RISK (click [here](#) for information)
- On submission of this form, a notification email will be sent by Qualtrics. The email confirmation of approval/waiver must be included in the student's project submission

What is this form about?

All supervisors/coaches must grant or waive ethical approval for their student's research project BEFORE the student carries out data collection. There are four possible outcomes a supervisor can delegate to a student's ethics form: approve, waive, defer, and not approve.

This form is a legally binding agreement, therefore, when you sign and submit the form you are acknowledging your role as a supervisor in advising the student to maintain high professional standards of ethical conduct as set down in UK legislation. This includes maintaining the ethical principles and guidance provided by WMG and the University of Warwick.

When is the deadline to submit this form?

Supervisors/coaches must submit this form as soon as possible to enable the student to complete their research project and dissertation with sufficient time prior to the deadline. If it is not possible to grant or waive ethical approval, then it is important the supervisor/coach:

- Provide feedback to the student so necessary changes can be

<https://outlook.office.com/mail/inbox/id/AAQkAGM0MjQ4ZmQzLWZkOGQlNDQ4NC1hMzEwLTU4NzFkODEwNjgxMQAQAONH59Mutm9NhikQP...> 3/8

- Provides feedback to the student so necessary changes can be made to the ethics form to delegate approval/waiver
- Contacts the WMG Projects Team to inform them of any delays to the student's project completion
- Completes this form selecting the option "Ethical approval is not granted" and provides an explanation for this selection **OR** "Ethical approval requires deferral" and provides an explanation for this selection

Who needs to fill in this form?

The supervisor/coach is responsible for completing this form for each of their project students. A student will NOT be able to complete this form as it is for *supervisor* delegated approval of student research.

What information will be asked?

The form seeks information about the supervisor/coach, the project student, the student ethics form, and confirmations related to ethical approval.

To complete this form, please have the following information ready:

- Student's ID number and email address
- A PDF of the student's ethics form

To generate a PDF of the student's ethics form, print the email from Qualtrics containing the student's responses to the students ethics form and 'save as PDF'.

How many times can a supervisor submit this form?

This form can be submitted as many times as needed. The student will need to include the email confirmation of ethical approval/waiver in the Appendices of their dissertation. *Note this email confirmation of ethical approval/waiver will be sent to the supervisor and student after successfully submitting this form.*

How much time will it take to complete this form?

It is estimated that the form will take no more than 15 minutes to complete, including uploading a PDF of the student's ethics form.

How will the data collected in this form be used?

The data from this form will be shared with the Projects Team, Course Leaders, Discipline Group Leads, Programme Administrators, and Ethics Committees.

Will my data be safe?

Data will be securely stored on Warwick platforms.

What if I and/or my student has not received the email notification for ethical approval/waiver following submission of this form?

Please wait at least five minutes for the email notification to appear. You may need to check your Spam/Junk folder if the email does not appear in your inbox. A supervisor/coach is responsible for ensuring the student receives an email confirmation of ethical approval/waiver. This means a supervisor/coach may need to forward the email notification they have received from Qualtrics directly to the student.

Typically, if an email notification from Qualtrics is not received, it is due to an incorrect email address being provided when completing the form. You may need to resubmit the form if there is an error. If after resubmitting the form and correcting email addresses a notification email from Qualtrics does not appear, please contact the Projects Team.

What if I have a question about this form?

- If you have a question about the project requirements of the student's course, contact the Course Leader
- If you have a question about research ethics, contact WMG-FT-SPA@warwick.ac.uk

Q1.2. What programme is your project student enrolled on?

- Full-time Postgraduate Taught programme (e.g. FT MSc)

Q1.3. Is this your first year supervising a taught student project at WMG?

- Yes

Section 1. Supervisor/Coach Information

All questions must be answered with the correct information to ensure you receive an email notification of ethical approval/waiver. You can locate information about your staff ID and Warwick email address on your Tabula profile.

Q2.2. What is your name?

Xingyu Zhao

Q2.3. What is your university ID number? *This can be found on your Tabula profile.*

2272010

Q2.4. What is your **Warwick** email address? *Please check this is filled in correctly so you can receive an email notification following submission of this form.*

xingyu.zhao@warwick.ac.uk

As you have selected that this is **your first year supervising a taught student project at WMG**, please be aware that ALL supervisors/coaches must complete the following mandatory training in order to delegate ethical approval/waiver to taught student research:

- [Epigeum](#)
- [Information Security Essentials](#)
- [WMG Supervisor Training on Research Ethics](#)

To support you in your role as a supervisor/coach, please be aware of the following resources:

- WMG guidance for staff on taught student projects
 - [Guidance on Ethical Approval of Student Projects](#)
 - [Policy for Ethical Approval for Taught Student Projects](#)
- University of Warwick guidance on research projects
 - [Research Compliance](#)
 - [Research Integrity](#)

Section 2. Student Information

All questions must be answered with the correct information to ensure the student receives an email notification of ethical approval/waiver.

Q3.2. What is the student's name? *Please check this is filled in correctly by copying the information from the student's Tabula profile.*

YANTONG LU

Q3.3. What is the student's Warwick ID number? *Please check this is filled in correctly by copying the information from the student's Tabula profile.*

5528854

Q3.4. What is the student's Warwickemail address? *Please check this is filled in correctly by copying the information from the student's Tabula profile.*

Yantong.Lu@warwick.ac.uk

Q3.5. What is the student's project title? *This information can be copied from the student's ethics form.*

Object Detection with Uncertainty Quantification

Q3.6. What full-time master's course is the student enrolled on? *If the course is not listed, return to the first page of this form to select the correct programme.*

- MSc Smart, Connected and Autonomous Vehicles

Section 3. Confirmations of Research Ethics and Upload of Student's Ethics Form

All statements must be responded to in order to complete this form.

Q4.2. I have checked the student's ethics form and can confirm that:

- 1. I am satisfied that the student has completed ALL mandatory training as required by the University: **WMG Ethics Training, Epigeum, and Information Security Smart**
- 2. I have discussed the importance of ethical approval with the student and am satisfied that they understand the importance of maintaining high ethical standards when conducting research.
- 3 - FT MSc. I have discussed with the student that data collection must take place whilst the student is resident in the UK.
- 4. I have discussed with the student that one of the primary conditions of being granted ethical approval/waiver is the requirement to work with their supervisor to develop their research methods, approach and tools.

Q4.3. Please indicate if any of the following cases has been mentioned in regard to the student's ethics form:

- None of the above

Q4.5. Please upload a PDF record of the student's responses to the ethics form that you are granting/waiving ethical approval for. *Note if the student has created multiple versions of the ethics form, it is imperative that only the approved version is uploaded here.*
[\[Click here\]](#)

Q4.6. Please enter the version number of the student's ethics form that you are approving. *This is to ensure the correct ethics form is being approved in case the student has submitted multiple ethics forms.*

1

Q4.7. By submitting this form, I confirm:

- Ethical approval has been **granted** for this research

Embedded Data:

N/A