Adaptive Bounding Box Uncertainty via Conformal Prediction

Alexander Timans*
UvA-Bosch Delta Lab
University of Amsterdam

Christoph-Nikolas Straehle
Bosch Center for Artifical Intelligence
Robert Bosch GmbH

Kaspar Sakmann
Bosch Center for Artifical Intelligence
Robert Bosch GmbH

Eric Nalisnick
UvA-Bosch Delta Lab
University of Amsterdam

Abstract

We quantify the uncertainty in multi-object bounding box predictions via conformal prediction. Using novel ensemble and quantile regression formulations, we are able to achieve per-class prediction intervals with guaranteed coverage that are adaptive to object size. We validate our approaches on real-world datasets (COCO, Cityscapes, BDD100k) for 2D bounding box localization, and achieve the desired coverage targets with sensibly tight intervals.

1. Introduction

Quantifying a model's predictive uncertainty is essential for success in safety-critical applications such as autonomous driving [23] and mobile robotics [22]. Yet one obstacle to principled uncertainty quantification (UQ) for computer vision is the pervasive use of deep neural networks — which are often unamenable to traditional techniques for UQ. The framework of *Conformal Prediction* (CP) [27, 32] enables a form of distribution-free UQ that is agnostic to the predictive model's structure, rendering it well-suited for black-box models such as neural networks.

In this work, we propose a CP framework designed to quantify predictive uncertainties in multi-object detection tasks. CP allows us to produce *post-hoc*, distribution-free prediction intervals equipped with a coverage guarantee for the bounding boxes of new objects (of known classes). Specifically, we provide users with the following statement of assurance: *"The conformal prediction interval covers the object's true bounding box with probability* $(1-\alpha)$ *for any known object class"*, where α is an acceptable margin of error. In the context of autonomous driving, such a guarantee can, for example, help certify collision avoidance by steering clear of the outer interval limit. We ensure our prediction intervals are adaptive to object sizes by employing strategies based on ensembling and quantile regression.

In the experiments, we apply our approaches to a range of classes on three large-scale object detection datasets. We obtain adaptive and informative intervals that adhere to the above guarantee, and also improve upon prior work [1, 11].

2. Background

We now provide some background on the desired conformal coverage guarantee, formalize the multi-object detection setting and relate CP to our setting.

2.1. Conformal prediction

We consider the setting of split conformal prediction [24], where we perform a distinct single split between training data $\mathcal{D}_{train}$ and calibration data $\{(X_i, Y_i)\}_{i=1}^{n} \sim P_{XY}$. If we follow the general conformal procedure as outlined in Algorithm 1, we can provide a coverage guarantee for a new test sample $(X_{n+1}, Y_{n+1}) \sim P_{XY}$ in terms of a prediction set $\hat{C}(X_{n+1})$, where a finite-sample, distribution-free guarantee is given over the event of $\hat{C}(X_{n+1})$ containing Y_{n+1}.

That is, assuming the samples $\{(X_i, Y_i)\}_{i=1}^{n+1}$ are exchangeable[1], we have that

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

for some tolerated miscoverage rate $\alpha \in (0, 1)$. The provided guarantee is *marginally* valid since it holds on average across all sample draws from P_{XY}. This is in contrast to the ideal scenario of *conditionally* valid coverage per input sample X_{n+1}, which has been shown to be impossible to achieve without imposing further assumptions [14, 31].

*Corresponding author: a.r.timans@uva.nl.

[1] this can be considered a relaxed *i.i.d.* assumption on the data

As an in-between notion of conditionality, *class-conditional* validity can be achieved by conformalizing only across samples per distinct class (see e.g. [7]), yielding the following guarantee:

$$\mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1}) | Y_{n+1} = y) \geq 1 - \alpha \quad \forall y \in \mathcal{Y}, \quad (2)$$

where $\mathcal{Y} = \{1, \ldots, K\}$ are distinct class labels. The guarantee we aim to provide relates most strongly to Equation 2.

2.2. Object detection

We now formalize our multi-object detection setting. Consider an input image $X \in \mathbb{R}^{H \times W \times D}$, where H, W and D correspond to image height, width and channel depth. For each input image, we also receive a set of tuples (x_0, y_0, x_1, y_1, l), where $(x_0, y_0, x_1, y_1) \in \mathbb{R}^4$ are the coordinates indicating an object's bounding box location within the image, and $l \in \{1, \ldots, K\}$ represents the object's class label. Each tuple parameterizes an object, with a total of $O(X)$ ground truth objects located in the image. For image X we thus have responses $\{(x_0, y_0, x_1, y_1, l)_j\}_{j=1}^{O(X)}$. Note that the object detection model predicts $\hat{f}(X) = \hat{O}(X)$ objects, where $O(X)$ and $\hat{O}(X)$ don't necessarily match. We consider every object as an individual sample for conformalization, i.e., the same input X will produce multiple samples matching the different tuple responses $(x_0, y_0, x_1, y_1, l)_j$, $j = 1, \ldots, O(X)$.

2.3. Conformal prediction for object detection

We apply CP to the bounding boxes on a per-coordinate basis, in our 2D case denoted (x_0, y_0, x_1, y_1). However, let us consider the generalization to arbitrary coordinates c^k, $k = 1, \ldots, m$. Since we apply CP to real-valued coordinates, we face a regression task and our sets $\hat{C}(X_{n+1}) \in \mathbb{R}$ take the form of prediction intervals (PIs). Given our multi-object detection setting, we consider a *class-conditional* guarantee to be most meaningful. Intuitively, it would not be sensible to, for example, use coordinates for detected objects of class 'car' to inform our prediction interval construction for objects of class 'person', as we would for a general *marginal* guarantee. Rather, we conformalize within each group of objects belonging to a common class.

Given the above abstraction to m coordinates and a fixed class label l for any grouped objects, an individual sample response can be interpreted as a realized random variable of the m coordinates only, i.e., we define $Y_i := (c_i^1, \ldots, c_i^k, \ldots, c_i^m) \in \mathbb{R}^m$. The *class-conditional* coverage guarantee we strive for in Equation 2 is then reinterpreted in our context as

$$\mathbb{P}\Big(\bigcap_{k=1}^{m} (c_{n+1}^k \in \hat{C}^k(X_{n+1})) | l_{n+1} = y\Big) \geq 1 - \alpha \quad \forall y \in \mathcal{Y}, \quad (3)$$

where components are indexed accordingly for a specific coordinate dimension k, e.g., $\hat{C}^k(X_{n+1})$. Conformalizing each coordinate separately gives rise to multiple testing issues as described in subsection B.2, which we address with our own correction scheme.

Finally, it is crucial to highlight that the construction of $\hat{C}^k(X_{n+1})$ necessitates a correct class label prediction in order to satisfy validity. That is, we require $\hat{l}_{n+1} = l_{n+1}$ and the provided guarantee in Equation 3 thus only holds strictly for true positive object detections, an important limitation also noted by [1, 11].

Prior work. Conformalized PIs for bounding boxes have been recently considered in [1, 11] (see also Appendix A). Given our theoretical formulation, we identify the following limitations in their approaches, which we address:

1. CP is applied to a single class only, thus providing the most trivial form of the *class-conditional* guarantee in Equation 3.
2. Constructed PIs are one-sided intervals (an outer limit), and more informative two-sided versions are not considered.
3. To correct for multiple testing, [11] apply Bonferroni and [1] additionally consider a $\max(\cdot)$ operation. Both employed corrections do not efficiently account for correlation structure between box coordinates (see also subsection 3.1).

3. Methods

A key modelling decision in CP is the choice of scoring function $s(\cdot)$ for computing conformity scores (see Algorithm 1). We experiment with three different choices of scoring function and subsequent PI construction.

Standard conformal (`StdConf`). We firstly consider the simple case of using regression residuals as scores [27], i.e., $s(\hat{f}(x), y) = |\hat{f}(x) - y|$. The resulting conformal PIs are constructed as $\hat{C}(X_{n+1}) = [\hat{f}(X_{n+1}) - \hat{q},\ \hat{f}(X_{n+1}) + \hat{q}]$, where $\hat{q}$ denotes the computed conformal quantile. This approach is a straight-forward alteration of [11] for two-sided PIs. However, it only provides non-adaptive, fixed-width PIs around coordinates.

Ensemble conformal (`EnsConf`). In order to construct adaptive PIs, we next consider using normalized residual scores [19] of the form $s(\hat{f}(x), y) = |\hat{f}(x) - y| / \hat{\sigma}(x)$, where $\hat{\sigma}(\cdot)$ is some choice for a heuristic uncertainty estimate. The resulting conformal PIs are constructed as $\hat{C}(X_{n+1}) = [\hat{f}(X_{n+1}) - \hat{\sigma}(X_{n+1})\,\hat{q},\ \hat{f}(X_{n+1}) + \hat{\sigma}(X_{n+1})\,\hat{q}]$. By incorporating a notion of uncertainty, the constructed PIs can be scaled individually in each coordinate and thus be adaptive in their magnitudes. We employ an ensemble of object detectors and quantify $\hat{\sigma}(\cdot)$ as the standard deviation in model predictions [18], while $\hat{f}(\cdot)$ is obtained using weighted box fusion [28].

Conformal quantile regression (`CQRConf`)**.** As another adaptive method, we accommodate Conformal Quantile Regression (CQR) [25] for our setting. We first modify our object detection model and train it with a *quantile loss* function to produce lower and upper quantile predictions $\hat{q}^{low}_{\alpha/2}$ and $\hat{q}^{high}_{1-\alpha/2}$ per bounding box coordinate[2]. Under some regularity conditions, these will converge asymptotically to the true conditional quantiles [8, 15] and achieve target coverage $(1-\alpha)$, suggesting their usefulness also for practical finite-sample cases. We subsequently perform CQR by defining scores $s(\hat{f}(x), y) = \max\{\hat{q}^{low}_{\alpha/2}(x) - y, y - \hat{q}^{high}_{1-\alpha/2}(x)\}$, and constructing conformal PIs as $\hat{C}(X_{n+1}) = [\hat{q}^{low}_{\alpha/2}(X_{n+1}) - \hat{q},\ \hat{q}^{high}_{1-\alpha/2}(X_{n+1}) + \hat{q}]$. For further implementation details see subsection C.1.

3.1. Multiple testing correction

Conformalizing each of m coordinates separately gives rise to multiple testing issues, since interpreting CP from a hypothesis testing view means running m permutation tests on nonconformity in parallel [32], resulting in a guaranteed coverage of at most $(1 - m\alpha)$ (see Equation 7). A naive correction can be achieved using Bonferroni, since choosing $\alpha_B = \alpha/m$ will satisfy target coverage. However, the Bonferroni correction is overly conservative under positive dependency of the individual hypothesis [33], which is reasonable to assume given that all coordinates parametrize an object's bounding box jointly. In fact, [5] show that a set of conformal p-values exhibits positive dependency structure *a priori*[3]. We propose an alternative procedure that is able to exploit correlation among coordinates efficiently for a less conservative correction without loss of power. We operate in the rank domain of coordinate-wise conformal scores and collapse the multiple testing problem to a single hypothesis test via a $\max(\cdot)$ operator. Applying a $\max(\cdot)$ has been considered before for CP [1, 7, 26], but not in conjunction with ranks and as a stand-alone multiple testing correction procedure. We show in subsection B.3 that our procedure `max-rank` satisfies exchangeability and validity, permitting its integration into any CP approach.

4. Experiments

For our experiments, we rely on pre-trained object detection models from `detectron2` [34], primarily based on a Faster R-CNN architecture and trained on `COCO-train` [21]. We consider datasets `COCO-val`, `cityscapes` [10] and `BDD100k-train` [35], which are split into calibration and test data respectively. We conformalize object bounding boxes for a variety of classes (hence our *class-conditional* approach), and focus our results on a coherent set of COCO classes that class labels across the datasets can be mapped to: {`person`, `bicycle`, `car`, `motorcycle`, `bus`, `truck`} (see subsection C.2).

In order to compute conformal scores for each object, we need to establish a pairing between an object's ground truth and predicted bounding boxes. Similarly to [1, 11], we perform Hungarian matching [17] based on an intersection-over-union (IoU) threshold of 0.5. Throughout, we set $\alpha = 0.1$ for a target coverage of 90%.

4.1. Metrics

We validate the predictive performance of our object detection model and assess the key desiderata of CP using respective metrics: validity via empirical coverage, adaptivity via size-stratified coverage, and efficiency via mean prediction interval width.

Predictive performance. We follow standard practice and validate model performance using the metrics from the COCO detection challenge[4], i.e., average precision across multiple IoU thresholds as well as object sizes (see Appendix D).

Validity. We ensure that our procedure satisfies the coverage guarantee by verifying empirical coverage. Let us consider a test dataset $\{(X_j, Y_j)\}_{j=n+1}^{n+n_t}$ of size n_t, then we define empirical coverage as

$$\texttt{cov} = \frac{1}{n_t} \sum_{j=n+1}^{n+n_t} \mathbb{1}\Big[\bigcap_{k=1}^{m} (c^k_j \in \hat{C}^k(X_j))\Big]. \tag{4}$$

Note that `cov` is a random quantity parametrized by a distribution and may deviate from target coverage $(1-\alpha)$ based on factors such as calibration set size n [31].

Adaptivity. To verify if target coverage is achieved by compensating undercoverage on some objects with overcoverage on others, similarly to [2] we verify coverage across different strata, namely object sizes. We follow the COCO detection challenge and verify for three sizes of bounding box surface areas: `cov-small` (area $\leq 32^2$), `cov-med` (area $\in (32^2, 96^2]$) and `cov-large` (area $> 96^2$).

Efficiency. We want constructed conformal PIs to be as narrow as possible while still satisfying ground truth coverage (i.e., remaining valid). We assess their efficiency using mean prediction interval width. If we consider an obtained

[2] Note that these are *not* conformalized quantiles

[3] they are jointly *positive regression dependent on a subset* [6]

[4] `https://cocodataset.org/#detection-eval`

Dataset	Method	calib. size n	cov	cov-small	cov-mid	cov-large	MPIW
COCO-val	StdConf	905	0.8816	0.9970	0.9505	0.7649	55.8229
	EnsConf	914	0.8829	0.8729	0.8622	0.8975	58.1730
	CQRConf	905	0.8862	0.9695	0.9201	0.8091	56.8073
cityscapes	StdConf	3010	0.8909	0.9980	0.9571	0.8358	62.5641
	EnsConf	2947	0.8901	0.8782	0.8829	0.8924	82.1022
	CQRConf	3010	0.8908	0.8716	0.9227	0.8587	65.2987
BDD100k-train	StdConf	53133	0.8988	0.9979	0.9610	0.7473	47.8932
	EnsConf	52278	0.8992	0.8737	0.8849	0.9189	60.4757
	CQRConf	53133	0.8988	0.9654	0.9363	0.7942	50.6458

Table 1. Metrics comparison of tested bounding box conformalization procedures across three datasets using the max-rank correction. Values are means over trials and selected set of classes. cov is expressed as a fraction and should be close to 0.9 (i.e. 90%), while MPIW is expressed in pixels. A trade-off between MPIW (efficiency) and stratified coverage (adaptivity) is apparent.

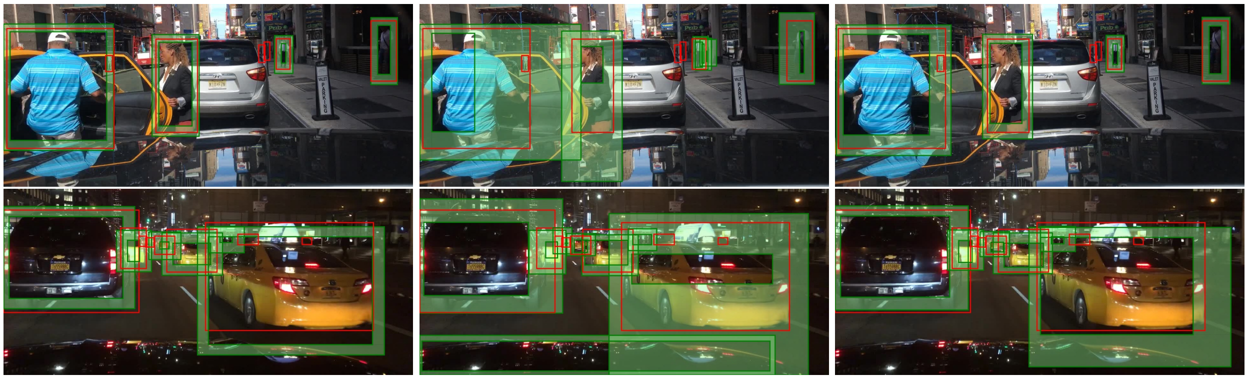

Figure 1. Examples for conformalized bounding boxes on BDD100k-train for classes {person, car}. Left to right by column: StdConf, EnsConf, CQRConf. Ground truth boxes are in red, two-sided conformal prediction interval regions are shaded in green.

PI to be of the form $\hat{C}(X_{n+1}) = [\hat{L}(X_{n+1}), \hat{U}(X_{n+1})]$, then we define the metric as

$$\texttt{MPIW} = \frac{1}{n_t m} \sum_{j=n+1}^{n+n_t} \sum_{k=1}^{m} |\hat{U}^k(X_j) - \hat{L}^k(X_j)|. \quad (5)$$

4.2. Results

Our main comparison of the three conformalization procedures across datasets is displayed in Table 1. We use our max-rank approach for multiple testing correction, and also report the mean size of calibration sets $|\mathcal{D}_{cal}| = n$. Values are averaged across 1000 trials of random calibration and test splits, and across the selected set of classes[5].

We find that empirical coverage satisfies target coverage to the extent that calibration set sizes n allow, with cov for BDD100k-train closest to 90%. Strictly by MPIW, the basic StdConf method seems to perform best. However, efficiency comes at the cost of adaptivity, with overcoverage for small objects compensating undercoverage for larger ones. We find this effect to balance out for CQRConf and in particular EnsConf, as corroborated by results on a per-class basis in Table 6.

This benefit is also apparent visually in Figure 1, where PIs for these methods scale adaptively in individual coordinates as needed. That is, they grow for uncertain box boundaries due to e.g. partial occlusions, and shrink for certain boundaries. We conclude that for detection tasks with equally-sized objects, StdConf provides good performance, while tasks with multiple object sizes can benefit from the adaptivity properties of EnsConf and CQRConf. In comparison to using a Bonferroni correction as seen in Table 4, our max-rank approach produces substantially tighter intervals and suppresses overcoverage tendencies.

Baseline comparison. We run additional experiments comparing our results to [1], a follow-up research on [11]. They propose conformal scores to construct outer, one-sided prediction intervals, and suggest using both Bonferroni and a $\max(\cdot)$ operation for multiple testing correction. We adapt our approaches to produce one-sided PIs, and also compare CP methods using their proposed 'box stretch' metric (stretch). Further details can be found in subsec-

[5] {person, bicycle, car, motorcycle, bus, truck}

tion C.4. Our results in Table 5 show that our combination of CP methods and `max-rank` correction achieves tighter PIs both in terms of `stretch` and `MPIW` while remaining equally valid.

5. Conclusion

We present and evaluate CP methods to generate PIs for bounding boxes with a *class-conditional* coverage guarantee for new samples. Our proposals, which include a novel multiple testing correction subroutine, generate PIs that are adaptive to object size. A notable limitation to the provided guarantee is the condition on correct label prediction, which we aim to address in the future by additional conformalized label sets. We also plan the extension of our methods to 3D bounding boxes and other detection tasks.

Acknowledgements

We thank Metod Jazbec and Mona Schirmer for helpful discussions. AT and EN are generously supported by the Bosch Center for Artificial Intelligence.

References

[1] Léo Andéol, Thomas Fel, Florence De Grancey, and Luca Mossina. Confident Object Detection via Conformal Prediction and Conformal Risk Control: An Application to Railway Signaling. 2023. 1, 2, 3, 4, 7, 10, 11, 12

[2] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty Sets for Image Classifiers using Conformal Prediction, Sept. 2022. 3

[3] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control, Sept. 2022. 7

[4] Anastasios N Angelopoulos, Stephen Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023. 7

[5] Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for Outliers with Conformal p-values, May 2022. 3

[6] Yoav Benjamini and Daniel Yekutieli. The Control of the False Discovery Rate in Multiple Testing under Dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001. 3

[7] Maxime Cauchois, Suyash Gupta, and John C Duchi. Knowing what You Know: Valid and validated confidence sets in multiclass and multilabel prediction. page 42, 2021. 2, 3

[8] Probal Chaudhuri. Global nonparametric estimation of conditional quantile functions and their derivatives. *Journal of multivariate analysis*, 39(2):246–269, 1991. 3

[9] Daniel Commenges. Transformations which preserve exchangeability and application to permutation tests. *Journal of Nonparametric Statistics*, 15(2):171–185, Jan. 2003. 8

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[11] Florence de Grancey, Jean-Luc Adam, Lucian Alecu, Sébastien Gerchinovitz, Franck Mamalet, and David Vigouroux. Object Detection with Probabilistic Guarantees: A Conformal Prediction Approach. In Mario Trapp, Erwin Schoitsch, Jérémie Guiochet, and Friedemann Bitsch, editors, *Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops*, volume 13415, pages 316–329, Cham, 2022. Springer International Publishing. 1, 2, 3, 4, 7

[12] Di Feng, Ali Harakeh, Steven L Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):9961–9980, 2021. 7

[13] Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Conformal Prediction Sets with Limited False Positives, Feb. 2022. 8

[14] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2020. 1

[15] Roger Koenker and Gilbert Bassett. Regression Quantiles. *Econometrica*, 46(1):33, Jan. 1978. 3, 9

[16] Arun Kumar Kuchibhotla. Exchangeability, Conformal Prediction, and Rank Tests, June 2021. 8

[17] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3

[18] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. page 12, 2017. 3, 7

[19] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 113(523):1094–1111, July 2018. 2

[20] Shuo Li, Sangdon Park, Xiayan Ji, Insup Lee, and Osbert Bastani. Towards PAC Multi-Object Detection and Tracking, Apr. 2022. 7

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, Feb. 2015. 3

[22] Björn Lütjens, Michael Everett, and Jonathan P How. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8662–8668. IEEE, 2019. 1

[23] RT McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017. 1

[24] Harris Papadopoulos, Volodya Vovk, and Alex Gammerman. Conformal Prediction with Neural Networks. In *19th*

IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007), volume 2, pages 388–395, Oct. 2007. 1

[25] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized Quantile Regression. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 3, 9

[26] Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent Accelerated Inference via Confident Adaptive Transformers, Sept. 2021. 3

[27] Glenn Shafer and Vladimir Vovk. A Tutorial on Conformal Prediction. page 51, 2008. 1, 2

[28] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, Mar. 2021. 3

[29] Ingo Steinwart and Andreas Christmann. Estimating conditional quantiles with the help of the pinball loss. 2011. 9

[30] Ryan J Tibshirani, Emmanuel J Candès, Rina Foygel Barber, and Aaditya Ramdas. Conformal Prediction Under Covariate Shift. page 11, 2019. 8

[31] Vladimir Vovk. Conditional Validity of Inductive Conformal Predictors. In *Proceedings of the Asian Conference on Machine Learning*, pages 475–490. PMLR, Nov. 2012. 1, 3

[32] Vladimir Vovk, A. Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005. 1, 3, 7

[33] Vladimir Vovk, Bin Wang, and Ruodu Wang. Admissible ways of merging p-values under arbitrary dependence. *The Annals of Statistics*, 50(1):351–375, 2022. 3

[34] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 3

[35] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning, Apr. 2020. 3

Appendix

A. Related work

Most existing methods for quantifying uncertainty in bounding box regression problems rely on adapting to the task at hand well-known methods from the general uncertainty quantification literature such as deep ensembles [18], Monte Carlo dropout or Bayesian learning, and may require substantial modifications to the model architecture (see [12] for a recent survey). However, such uncertainty quantification methods do not provide a theoretical guarantee or validity statement on the quality of obtained uncertainty estimates, specifically on any obtained prediction intervals containing the ground truth bounding box.

Recent attempts at providing such guarantees for the bounding box localization problem include the use of the Probably Approximatly Correct (PAC) framework to produce a guarantee by composition of PAC prediction sets at multiple model pipeline stages [20]; providing guarantees on multiple risks such as recall and coverage using a sequential testing procedure based on p-values obtained from concentration inequalities [3]; and [1, 11], who leverage conformal prediction and can be considered the closest prior works.

Our approaches differ from the former methods in that we base our obtained guarantee purely on conformal prediction, as opposed to other frameworks. Our approaches differ from [1, 11] in that we consider multiple classes at once and establish *class-conditional* guarantees, investigate the use of novel adaptive conformity scores, propose a less conservative multiple testing correction procedure, and validate our results across several datasets. Furthermore, our methods are designed for the construction of two-sided prediction intervals, as opposed to an outer prediction box only, providing more granular bounding box information.

B. Mathematical details

B.1. Split conformal prediction

The general conformal procedure for split conformal prediction is provided in Algorithm 1. We point to [4] for a user-friendly introduction to conformal prediction.

In classification tasks, prediction sets $\hat{C}(X_{n+1}) \subseteq \{1, \dots, K\}$ may be a finite subset of the K class labels. In regression tasks, $\hat{C}(X_{n+1}) \subseteq \mathcal{Y}$ may be a prediction interval on the domain of $\mathcal{Y}$, e.g., $\hat{C}(X_{n+1}) = [\hat{f}(X_{n+1}) - \hat{q}, \hat{f}(X_{n+1}) + \hat{q}]$.

Algorithm 1 Split conformal prediction

```
**Input:** data $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$, prediction algorithm $\mathcal{A}$, miscoverage level $\alpha \in (0, 1)$.
**Output:** Prediction set $\hat{C}(X_{n+1})$ for test sample $(X_{n+1}, Y_{n+1})$.
**Procedure:**
Split data $\mathcal{D}$ into two disjoint subsets: a proper training set $\mathcal{D}_{train}$ and calibration set $\mathcal{D}_{cal} = \{(X_i, Y_i)\}_{i=1}^{n}$.
Fit a prediction model on the proper training set:
    $\hat{f}(\cdot) \leftarrow \mathcal{A}(\mathcal{D}_{train})$.
Define a scoring function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ applied to $\mathcal{D}_{cal}$, resulting in (non)conformity scores
    $S = \{s(\hat{f}(X_i), Y_i)\}_{i=1}^{n}$.
  $s(\cdot)$ encodes a notion of agreement (conformity) between prediction $\hat{f}(X_i)$ and ground truth $Y_i$.
Compute a conformal quantile $\hat{q}$, defined as the
    $\lceil (n+1)(1-\alpha)/n \rceil$-th empirical quantile of $S$.
  $\hat{q}$ is a sample-corrected quantile choice that guarantees target coverage $(1-\alpha)$ by construction.
For a new test sample $(X_{n+1}, Y_{n+1})$, a valid conformal prediction set for $X_{n+1}$ is given by
    $\hat{C}(X_{n+1}) = \{y \in \mathcal{Y} : s(\hat{f}(X_{n+1}), y) \leq \hat{q}\}$.
  Validity refers to satisfying the guarantee in Equation 1.
**End procedure**
```

B.2. Multiple testing problem

We first need to establish the equivalence of several events in the context of obtained prediction sets (here in the form of intervals). Let us extend the notation for the nonconformity score of a single sample (X_i, Y_i) from $\mathcal{D}_{cal}$ to s_i, and for all samples to $s_{1:n}$. The conformal quantile $\hat{q}$ at coverage level $(1-\alpha)$ computed from the scores $s_{1:n}$ can also be denoted as $\hat{q}(1-\alpha; s_{1:n})$. Furthermore, let $p(s_{n+1}, s_{1:n})$ be a valid p-value of the hypothesis test on the nonconformity of s_{n+1}, i.e. the null hypothesis supports that s_{n+1} conforms to $s_{1:n}$. We note that this is an alternative interpretation of the conformal prediction set construction from the hypothesis testing perspective [32].[6]

Then, by construction of $\hat{C}(X_{n+1})$ and duality of hypothesis testing and confidence intervals we have the equivalence of following events:

$$\begin{aligned} Y_{n+1} \notin \hat{C}(X_{n+1}) &\Leftrightarrow s_{n+1} > \hat{q}(1-\alpha; s_{1:n}) \\ &\Leftrightarrow p(s_{n+1}, s_{1:n}) \leq \alpha. \end{aligned} \tag{6}$$

[6] A valid p-value P is defined as $\mathbb{P}(P \leq \alpha) \leq \alpha \quad \forall \alpha \in [0, 1]$.

This can be used to showcase the arising multiple testing problem for conformalizing on coordinate-level as follows:

$$\begin{aligned}\mathbb{P}(\bigcap_{k=1}^{m}(c_{n+1}^{k} \in \hat{C}^{k}(X_{n+1}))) \\ &= 1-\mathbb{P}(\bigcup_{k=1}^{m}(c_{n+1}^{k} \notin \hat{C}^{k}(X_{n+1}))) \\ &\geq 1-\sum_{k=1}^{m}\mathbb{P}(c_{n+1}^{k} \notin \hat{C}^{k}(X_{n+1})) \\ &= 1-\sum_{k=1}^{m}\mathbb{P}(p(s_{n+1}^{k}, s_{1:n}^{k}) \leq \alpha_k) \\ &= 1-\sum_{k=1}^{m}\alpha_k \\ &= 1-m\alpha \quad \text{since } \alpha_k=\alpha\,\forall k \text{ is fixed.}\end{aligned} \tag{7}$$

We observe that we cannot guarantee target coverage since $1-m\alpha \leq 1-\alpha$ for any $m \in \mathbb{N}^{+}$.

B.3. Multiple testing correction (max-rank)

We show that our approach satisfies exchangeability and validity by demonstrating that these properties hold for the two key components of it: 1) operating in the domain of ranks of scores, as opposed to working with scores directly; and 2) applying a $\max(\cdot)$ operation on these ranks (or any set of exchangeable random variables (RVs)).

We begin by showing the exchangeability property, followed by validity. We then formulate our procedure algorithmically in Algorithm 2.

Exchangeability preservation. Following the notation in [16], let us abbreviate the set of indices $\{1,\ldots,n\}$ by $[n]$ for any $n \geq 1$. We also define the rank of element x_i in a set $\{x_1,\ldots,x_n\}$ of n distinct elements as

$$\text{rank}(x_i;\{x_{1:n}\}) := |\{j \in [n] : x_j \leq x_i\}|, \tag{8}$$

i.e., the size of the set of elements smaller or equal x_i. Note that elements and therefore ranks need to be distinct (no ties), potentially by introducing jitter noise. We can then invoke the following theorem:

Theorem 1 (Distribution of ranks [16], Thm. 2)**.** *For exchangeable RVs $X_1,\ldots,X_n$ we have that*

$$(rank(X_i;\{X_{1:n}\}) : i \in [n]) \sim Unif(\{\pi : [n] \to [n]\}),$$

where Unif$(\cdot)$ is the uniform distribution over all permutations of [n], i.e., each permutation occurs with equal probability $1/n!$.

If we consider the ranks over conformal scores, which can be shown to be exchangeability-preserving [13, 30], by Theorem 1 the ranks of $X_i, i \in [n]$ are therefore exchangeable and their distribution does not depend on the distribution of X_i.

For exchangeability of the $\max(\cdot)$ operator, we make use of the following theorem:

Theorem 2 (Exchangeability under transformations [9, 16])**.** *Given a vector of exchangeable RVs $X = (X_1,\ldots,X_n) \in \mathcal{X}^n$ and a fixed transformation G, we consider $G(\cdot)$ exchangeability-preserving if for each permutation $\pi_1 : [m] \to [m]$ there exists a permutation $\pi_2 : [n] \to [n]$ s.t.*

$$\forall x \in \mathcal{X}^n : \pi_1 G(x) = G(\pi_2 x).$$

Given the definition of exchangeable RVs in our setting as $Y_i = (c_i^1,\ldots,c_i^m)$, $i=1,\ldots,n$, a vector of RVs is given as $Y=(Y_1,\ldots,Y_n) \in \mathcal{Y}^n$ and we fix the transformation

$$G : Y \mapsto (\max_{1\leq k\leq m} c_1^k,\ldots,\max_{1\leq k\leq m} c_n^k). \tag{9}$$

Then for any $y \in \mathcal{Y}^n$ we have that

$$\pi_1 G(y) = \pi_1 \max_{1\leq k\leq m} y = \max_{1\leq k\leq m} \pi_2 y = G(\pi_2 y) \tag{10}$$

since $G(\cdot)$ is symmetric and indifferent to permutation. Thus $G(Y)$ is also exchangeable, i.e., the $\max(\cdot)$ operator is exchangeability-preserving following Theorem 2. We can replace RVs Y_i, $i=1,\ldots,n$ with any other set of exchangeable RVs of the same shape, such as the ranks of scores.

Validity. We show validity of operating in the domain of ranks by use of the following corollary:

Corollary 1 ([16], Corr. 1)**.** *Under assumptions of Theorem 1, we have that*

$$\mathbb{P}(rank(X_n;\{X_{1:n}\}) \leq t) = \frac{\lfloor t \rfloor}{n},$$

for $t \in \mathbb{R}$. In addition, the RV $rank(X_n;\{X_{1:n}\})/n$ is a valid p-value.

Note that for the ranks of scores we have $\text{rank}(s_i;\{s_{1:n}\}) = i$, i.e., the rank of the score at position i is the position index itself (assuming unique scores). We set $t = \text{rank}(s_{\lceil(n+1)(1-\alpha)\rceil};\{s_{1:(n+1)}\}) = \lceil(n+1)(1-\alpha)\rceil$ and using the exchangeability of ranks and Corollary 1 we have

$$\begin{aligned}\mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1})) &= \mathbb{P}(\text{rank}(s_{n+1}; \{s_{1:(n+1)}\}) \leq t) \\ &= \sum_{i=1}^{t} \mathbb{P}(\text{rank}(s_{n+1}; \{s_{1:(n+1)}\}) = i) \\ &= \frac{t}{n+1} \\ &= \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1} \\ &\geq \frac{(n+1)(1-\alpha)}{n+1} = 1 - \alpha. \end{aligned} \tag{11}$$

For validity of the $\max(\cdot)$ operator, consider working with scores directly. The set of scores $\{s_i^{\max}\}_{i=1}^{n+1}$ where $s_i^{\max} = \max_{1 \leq k \leq m} s_i^k$ is exchangeable since the $\max(\cdot)$ operator is exchangeability-preserving. Then we have for the conformal quantiles that $\hat{q}(1-\alpha; s_{1:n}^{\max}) \geq \hat{q}(1-\alpha; s_{1:n}^k)$ and thus

$$\begin{aligned}\mathbb{P}(s_{n+1}^k \leq \hat{q}(1-\alpha; s_{1:n}^{\max})) &\geq \mathbb{P}(s_{n+1}^k \leq \hat{q}(1-\alpha; s_{1:n}^k)) \\ &\geq 1 - \alpha \end{aligned} \tag{12}$$

for any $k \in 1, \ldots, m$, where the last inequality follows from the construction of $\hat{q}(1-\alpha; s_{1:n}^k)$ to be valid if we conformalize in each dimension k. In other words, since $\hat{C}^{\max} \supseteq \hat{C}^k$ and $\hat{C}^k$ is valid, then $\hat{C}^{\max}$ also has to be. The argumentation can be translated to working with ranks of scores, given that ranks are also exchangeable and valid.

Procedure. We now formalize and describe our multiple testing correction approach `max-rank`. For notation purposes, we denote matrices upper-case and bold ($\mathbf{X}$), vectors lower-case and bold ($\mathbf{x}$), scalars without bolding (x or X). Define the l^∞-norm as $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$ for a discrete vector $\mathbf{x}$ of size n. We further abbreviate the rank of a score $\text{rank}(s_i; \{s_{1:n}\})$ as r_i.

Then for $\mathcal{D}_{cal} = \{(X_i, Y_i)\}_{i=1}^n$ we have the following relevant matrices:

$$\begin{aligned} &\text{Score matrix } \mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_n) = [s_{ik}] \in \mathbb{R}^{n \times m} \\ &\text{Rank matrix } \mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_n) = [r_{ik}] \in \mathbb{N}^{n \times m}, \end{aligned} \tag{13}$$

where $1 \leq i \leq n$, $1 \leq k \leq m$. The algorithmic procedure is described in Algorithm 2.

Note that the given algorithm optimizes for coverage across all coordinate dimensions simultaneously by use of the l^∞-norm, however it is also possible to select a *coverage optimization criterion* and optimize for coverage in a specific subset of coordinate dimensions limited by indices $i, j \in [m] : i < j$. The algorithm then undergoes some modifications which we omit here.

Algorithm 2 Multiple testing correction via `max-rank`

```
Input: Score matrix $\mathbf{S}$, rank matrix $\mathbf{R}$
Output: Coordinate-wise quantiles $\hat{q}^k \; \forall k \in [m]$
Procedure:
    Apply $\|\cdot\|_\infty$ row-wise to $\mathbf{R}$:
        $\mathbf{r}^{\max} = (\|\mathbf{r}_1\|_\infty, \ldots, \|\mathbf{r}_n\|_\infty) = (\max_{1 \leq k \leq m} |r_{1k}|, \cdots, \max_{1 \leq k \leq m} |r_{nk}|) \in \mathbb{N}^n$
    Sort $\mathbf{r}^{\max}$ in ascending order (sort) and select $r_t^{\max}$ as the rank at index $t = \hat{q}(1-\alpha; \mathbf{r}^{\max})$ to ensure minimum desired coverage.
    Sort $\mathbf{S}$ column-wise such that
        $\forall k \in [m] : s_1^k < \cdots < s_n^k$.
    Select for each column in $\mathbf{S}$ the score at rank $r_t^{\max}$ as conformal quantile, i.e.,
        $\forall k \in [m] : \hat{q}^k = s_{r_t^{\max}}^k$.
End procedure
```

C. Implementation details

C.1. Conformal quantile regression (`CQRConf`)

We modify an object detection model to regress to estimated conditional quantiles of the bounding box coordinates alongside a standard mean prediction. This is achieved by supplementing the model's final regression output layer with additional box prediction heads, freezing all pre-trained weights, and training the additional heads with a *quantile loss* function, also called *pinball loss* [15, 29].

The loss for some quantile estimator $\hat{q}_\tau$ of the τ-th quantile is given by

$$\mathcal{L}(y, \hat{q}_\tau) = \begin{cases} \tau \left(y - \hat{q}_\tau(x)\right) & \text{if } y - \hat{q}_\tau(x) > 0 \\ (1-\tau)\left(\hat{q}_\tau(x) - y\right) & \text{else.} \end{cases} \tag{14}$$

It intuitively penalizes both under- and overcoverage weighted by the target quantile τ, and recovers the $L1$-loss for $\tau = 0.5$. Since the box heads are architecturally independent, we can train arbitrary many quantile estimators in parallel, where we obtain an individual loss $\mathcal{L}(y, \hat{q}_\tau)$ for each τ. The final loss for model updating is the sum of all individual quantile losses.

For CQR, we require only lower and upper quantiles τ^{low} and τ^{high}. If we aim for target coverage $(1-\alpha)$, a reasonable choice is $\tau^{low} = \alpha/2$ and $\tau^{high} = 1 - \alpha/2$, since the obtained interval $[\hat{q}_{\alpha/2}^{low}, \hat{q}_{1-\alpha/2}^{high}]$ will asymptotically achieve target coverage. However, in practice we require further interval scaling via CQR to obtain valid coverage in finite samples. Note that the choices for τ^{low} and τ^{high} are a modelling decision, and can in fact be tuned to produce more efficient PIs without invalidating the conformal coverage guarantee [25]. However, we only consider the single setting with $\tau^{low} = \alpha/2$ and $\tau^{high} = 1 - \alpha/2$.

Dataset	# images	Object class					
		person	bicycle	car	motorcycle	bus	truck
COCO-val	5000	10777	314	1918	367	283	414
cityscapes	5000	24713	5871	33658	895	477	577
BDD100k-train	70000	96929	7124	701507	3023	11977	27963

Table 2. Image counts and object distributions for the selected set of classes.

For a given target coverage of 90%, these correspond to $\tau^{low} = 0.05$ and $\tau^{high} = 0.95$.

C.2. Dataset splits and class mappings

We display the distribution of objects per class for our selected set of classes in Table 2. Objects are assigned to either calibration or test data based on the assignment to either split for the respective image they belong to. We randomly split the images according to the following calibration set sizes as a fraction of total available data for each dataset: 50% for COCO-val, 50% for cityscapes and 70% for BDD100k-train.

Class mappings. Our pre-trained object detection models are trained on COCO-train and recognize all 80 COCO object instance classes. In order to permit the use of pre-trained models without further finetuning as well as find a common intersection of classes across all three datasets, we map relevant classes with available object instance annotations from cityscapes and BDD100k-train to equivalent COCO classes. For the considered set of classes {person, bicycle, car, motorcycle, bus, truck}, we find 1:1 correspondences for most classes. We additionally do the following mappings:

- for cityscapes, we map classes 'pedestrian' and 'rider' to class 'person';
- for BDD100k-train, we map classes 'person' and 'rider' to class 'person'.

C.3. Model details and parameter settings

The primarily used pre-trained model from detectron2 is a Faster R-CNN backbone model with feature pyramid network and a fully connected bounding box predition head, trained for $\sim$ 37 epochs on COCO-train[7].

Inference parameters. We identify two key parameters that filter the proposal boxes to produce the final bounding box predictions, which we fix as follows:

- The score parameter removes any box proposals that receive a model confidence score below a specified threshold, which we fix at 0.5.
- The non-maximum surpression parameter removes any superfluous box proposals that record an IoU overlap above the specified threshold except for the box with the highest confidence score. We set this value to an IoU of 0.6.

Quantile head training. We freeze all pre-trained model weights and only train the new box prediction heads with a compounded *quantile loss*. We set the learning rate to 0.02 and train for $\sim$ 3000 iterations on COCO-train with a batch size of 16.

C.4. Baseline comparison

We compare our approaches to the conformal scoring methods presented in [1], which have been designed for a one-sided, outer PI construction. Let us once again consider the 2D bounding box setting with coordinate tuples $Y = (x_0, y_0, x_1, y_1)$. Specifically, we compare to the following baselines:

(add + Bonf) We use the scores

$$s(\hat{f}(x), y) = (\hat{x}_0 - x_0, \hat{y}_0 - y_0, x_1 - \hat{x}_1, y_1 - \hat{y}_1) \quad (15)$$

and obtain conformalized outer PI coordinates as

$$\begin{aligned}\hat{C}(X_{n+1}) = (&\hat{x}_0 - \hat{q}(1-\alpha_B; s_{1:n}), \hat{y}_0 - \hat{q}(1-\alpha_B; s_{1:n}), \\ &\hat{x}_1 + \hat{q}(1-\alpha_B; s_{1:n}), \hat{y}_1 + \hat{q}(1-\alpha_B; s_{1:n})),\end{aligned} \quad (16)$$

where $\hat{q}(1-\alpha_B; s_{1:n})$ is the coordinate-level conformal quantile at coverage level $(1-\alpha_B)$ based on scores $s_{1:n}$, and $\alpha_B = \alpha/4$ is the Bonferroni correction (Bonf) for target coverage $(1-\alpha)$.

(mult + Bonf) We use the scores

$$s(\hat{f}(x), y) = (\frac{\hat{x}_0 - x_0}{\hat{w}}, \frac{\hat{y}_0 - y_0}{\hat{h}}, \frac{x_1 - \hat{x}_1}{\hat{w}}, \frac{y_1 - \hat{y}_1}{\hat{h}}), \quad (17)$$

where $\hat{w} = \hat{x}_1 - \hat{x}_0$ and $\hat{h} = \hat{y}_1 - \hat{y}_0$ are the predicted box width and height respectively. We obtain conformalized

[7] model name X101-FPN, see https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

outer PI coordinates as

$$\hat{C}(X_{n+1}) = (\hat{x}_0 - \hat{w}\,\hat{q}(1-\alpha_B; s_{1:n}), \hat{y}_0 - \hat{h}\,\hat{q}(1-\alpha_B; s_{1:n}), \hat{x}_1 + \hat{w}\,\hat{q}(1-\alpha_B; s_{1:n}), \hat{y}_1 + \hat{h}\,\hat{q}(1-\alpha_B; s_{1:n})), \tag{18}$$

where once again $\alpha_B = \alpha/4$ is the Bonferroni correction.

(add + max, mult + max) Instead of computing coordinate-level quantiles corrected via Bonferroni, the authors in [1] suggest taking a $\max(\cdot)$ operation over coordinate scores in Equation 15 and Equation 17 respectively, resulting in a set of scores $s^{\max}_{1:n}$. A conformal quantile $\hat{q}(1-\alpha; s^{\max}_{1:n})$ is then computed directly at target coverage over these scores, alleviating the need for further correction. The idea is close to our multiple testing correction approach max-rank, but operates directly in the domain of scores instead. Resulting PI coordinates are constructed as in Equation 16 and Equation 18 simply by replacing the quantile.

We compare the above baselines to the following methods of ours:

(add + max-rank, mult + max-rank) We use the proposed conformal scores from Equation 15 and Equation 17 in conjunction with our own multiple testing correction max-rank. Note that Equation 15 is a one-sided (signed) version of the scores employed for StdConf, and Equation 17 can be considered related to the normalized scores as used for EnsConf.

(EnsConf + max-rank) We also compare to an adapted version of EnsConf for one-sided PIs, where we use Equation 17 but normalize by the obtained uncertainties $\hat{\sigma}(\cdot)$. We do not consider a modification to CQRConf because it is not straightforward how a one-sided version of its conformal scores should be constructed.

Box stretch metric. We implement the proposed 'box stretch' evaluation metric from [1] that assesses the additional box surface area incurred by conformalization. Formally, we denote the metric as

$$\texttt{stretch} = \frac{1}{n_t} \sum_{j=n+1}^{n+n_t} \sqrt{\frac{\mathcal{A}(\hat{C}(X_j))}{\mathcal{A}(\hat{f}(X_j))}}, \tag{19}$$

where $\mathcal{A}(\cdot)$ is the computed surface area of the bounding box formed by the respective input, i.e., the predicted bounding box coordinates, and the outer conformal PI bounds. Ideally, we desire stretch to be close to 1.0.

Mean prediction interval width. In order to further allow comparison using the MPIW metric, which is formally defined for two-sided PIs only, we additionally construct a two-sided version of each of the above methods by considering the distances to the predicted box center, i.e., we place a lower PI bound at the bounding box center coordinates. Note that we also do the same for our own (initially two-sided) methods to allow for a fair comparison.

D. Additional results

Predictive performance. We validate the predictive performance as measured via average precision (AP) metrics for our primary pre-trained object detection model across datasets in Table 3. Obtained scores are in line with expectations, confirming that the underlying predictive model performs adequately.

Main results. Table 4 displays our results for conformalization using a Bonferroni correction (rather than max-rank) to account for multiple testing. We observe overall inferior performance, as measured by generally larger MPIW caused by overcoverage tendencies. Table 6 displays our main results from Table 1, but instead of averaging across all classes we report results separately for each individual class. We also display additional exemplary conformalized bounding boxes for different classes on COCO-val, cityscapes and BDD100k-train in Figure 2, Figure 3 and Figure 4 respectively.

Baseline comparison. Table 5 displays the results for the comparison of our CP approaches to the baselines taken from [1] (see subsection C.4). We observe a superior performance for our methods, as measured by a lower stretch and MPIW.

Dataset	AP@IoU=.50:.05:.95	AP@IoU=.75	AP@IoU=.50	AP-small	AP-med	AP-large
COCO-val	0.4521	0.4937	0.6655	0.2184	0.2781	0.4281
cityscapes	0.432	0.4641	0.6637	0.027	0.0459	0.2782
BDD100k-train	0.3098	0.3141	0.5256	0.0745	0.14	0.3055

Table 3. Average precision (AP) scores following the COCO detection challenge metrics for our primarily employed pre-trained object detection model X101-FPN (see subsection C.3). Results are the mean over our selected set of COCO classes. The primary metric AP@IoU=.50:.05:.95 averages AP scores for 10 different IoU thresholds in $[0.5, 0.95]$ with step size 0.05. AP-small, AP-med and AP-large compute scores across object sizes (see similarly for cov in subsection 4.1).

Dataset	Method	calib. size n	cov	cov-small	cov-mid	cov-large	MPIW
COCO-val	StdConf	905	0.9445	0.9990	0.9781	0.8722	109.3209
	EnsConf	914	0.9406	0.9327	0.9329	0.9450	139.7342
	CQRConf	905	0.9417	0.9917	0.9629	0.8859	93.3717
cityscapes	StdConf	3010	0.9263	0.9983	0.9773	0.8802	86.2404
	EnsConf	2947	0.9275	0.8844	0.9241	0.9286	131.6933
	CQRConf	3010	0.9228	0.9436	0.9504	0.8929	82.7069
BDD100k-train	StdConf	53133	0.9138	0.9986	0.9698	0.7770	52.3321
	EnsConf	52278	0.9092	0.8880	0.8959	0.9270	64.2526
	CQRConf	53133	0.9091	0.9757	0.9449	0.8096	52.9734

Table 4. Metrics comparison of tested bounding box conformalization procedures across three datasets using the Bonferroni correction (Bonf). Values are means over trials and selected set of classes. cov is expressed as a fraction and should be close to 0.9 (i.e. 90%), while MPIW is expressed in pixels.

Dataset	Method	stretch (↓)	cov	cov-small	cov-mid	cov-large	MPIW (↓)
COCO-val	add + Bonf*	2.203	0.9452	0.9982	0.9767	0.8753	108.1652
	mult + Bonf*	1.5784	0.9401	0.9503	0.9274	0.9492	109.7565
	add + max*	1.6896	0.9081	0.9949	0.9610	0.8103	89.3792
	mult + max*	1.3877	0.9079	0.9211	0.8847	0.9221	94.5110
	add + max-rank	1.5966	0.8819	0.9954	0.9481	0.7688	**86.4771**
	mult + max-rank	**1.3529**	0.8821	0.8948	0.8626	0.8954	91.9430
	EnsConf + max-rank	1.3744	0.8830	0.8946	0.8608	0.8977	91.5822
cityscapes	add + Bonf*	1.7386	0.9265	0.9998	0.9822	0.8779	116.9056
	mult + Bonf*	1.5601	0.9220	0.8579	0.9210	0.9325	123.3368
	add + max*	1.6383	0.9053	0.9999	0.9746	0.8498	109.8954
	mult + max*	1.5229	0.9054	0.8966	0.8884	0.9178	117.7613
	add + max-rank	1.5592	0.8906	0.9999	0.9735	0.8308	**105.7331**
	mult + max-rank	**1.4580**	0.8907	0.8303	0.8833	0.9040	112.8632
	EnsConf + max-rank	1.5195	0.8903	0.9180	0.9059	0.8852	117.7881
BDD100k-train	add + Bonf*	1.7213	0.9128	0.9993	0.9662	0.7796	76.6727
	mult + Bonf*	1.5095	0.9069	0.8822	0.9000	0.9183	79.9462
	add + max*	1.6958	0.9007	0.9985	0.9576	0.7608	75.8324
	mult + max*	1.5361	0.9008	0.8903	0.8904	0.9145	80.9566
	add + max-rank	1.6633	0.8991	0.9989	0.9575	0.7536	**74.8984**
	mult + max-rank	**1.4943**	0.8991	0.8727	0.8906	0.9126	79.0366
	EnsConf + max-rank	1.6102	0.8992	0.8898	0.8858	0.9160	84.6317

Table 5. Metrics comparison of CP methods proposed in [1] (denoted with *) against one-sided versions of our bounding box conformalization procedures. Values are means over trials and selected set of classes. stretch is a fraction desired to be close to 1.0, cov is expressed as a fraction and should be close to 0.9 (i.e. 90%), while MPIW is expressed in pixels and desired to be low.

Dataset	Class	Method	calib. size n	cov	cov-small	cov-mid	cov-large	MPIW
COCO-val	person	StdConf	4326	0.8995	0.9993	0.9607	0.795	43.6731
		EnsConf	4374	0.899	0.8887	0.885	0.9165	43.5564
		CQRConf	4326	0.8995	0.995	0.9418	0.8137	38.2321
	bicycle	StdConf	93	0.8646	1.0	0.9298	0.7447	61.2584
		EnsConf	93	0.8737	0.8738	0.8371	0.909	60.9342
		CQRConf	93	0.8756	0.9869	0.9018	0.8057	63.5989
	car	StdConf	665	0.8963	0.9865	0.8844	0.6576	22.2448
		EnsConf	669	0.8964	0.8957	0.8985	0.8927	26.1741
		CQRConf	665	0.896	0.9539	0.895	0.7239	21.4515
	motorcycle	StdConf	131	0.8779	1.0	0.9849	0.8021	75.9179
		EnsConf	134	0.8795	0.7914	0.8765	0.8894	91.1611
		CQRConf	131	0.894	0.9939	0.9596	0.8443	78.4748
	bus	StdConf	109	0.8773	0.9964	0.9623	0.838	44.216
		EnsConf	107	0.8742	–	0.8243	0.896	43.772
		CQRConf	109	0.8778	0.8889	0.8902	0.8728	62.0776
	truck	StdConf	107	0.8741	1.0	0.9808	0.7519	87.6273
		EnsConf	106	0.8744	0.9147	0.852	0.8817	83.4402
		CQRConf	107	0.8739	0.9982	0.9321	0.7941	77.0087
cityscapes	person	StdConf	6017	0.8998	0.9949	0.9453	0.8013	37.7866
		EnsConf	5863	0.8994	0.8838	0.8925	0.9137	46.802
		CQRConf	6017	0.8999	0.9863	0.9381	0.8159	33.9054
	bicycle	StdConf	1132	0.8981	–	0.9779	0.8144	65.1831
		EnsConf	1142	0.8974	–	0.8972	0.8992	92.0829
		CQRConf	1132	0.8978	–	0.9331	0.861	66.0216
	car	StdConf	10472	0.8998	0.999	0.9498	0.8329	38.6513
		EnsConf	10263	0.8996	0.8725	0.9004	0.9022	50.1122
		CQRConf	10472	0.8996	0.9765	0.9258	0.8606	36.5904
	motorcycle	StdConf	184	0.8849	1.0	0.9806	0.8258	75.2226
		EnsConf	176	0.8858	–	0.9003	0.8748	107.983
		CQRConf	184	0.8849	0.652	0.9537	0.858	80.1514
	bus	StdConf	129	0.8836	–	0.9291	0.8773	72.9359
		EnsConf	124	0.8796	–	0.7926	0.8907	97.0354
		CQRConf	129	0.8827	–	0.8583	0.8865	90.725
	truck	StdConf	123	0.8794	–	0.9602	0.8631	85.6053
		EnsConf	115	0.8789	–	0.9143	0.8736	98.5978
		CQRConf	123	0.8796	–	0.9272	0.87	84.3987
BDD100k-train	person	StdConf	35417	0.8999	0.9895	0.9085	0.5928	25.0930
		EnsConf	35634	0.8997	0.8885	0.8993	0.9321	37.9262
		CQRConf	35417	0.9000	0.9775	0.9045	0.6520	23.3128
	bicycle	StdConf	1836	0.8986	1.0	0.9701	0.7043	50.8057
		EnsConf	1801	0.8984	0.9044	0.8897	0.9178	72.6298
		CQRConf	1836	0.8983	0.9773	0.9446	0.7697	51.4883
	car	StdConf	269917	0.8999	0.9979	0.9413	0.7599	34.0415
		EnsConf	264998	0.9001	0.8815	0.8953	0.9209	42.5502
		CQRConf	269917	0.9000	0.9867	0.9205	0.8011	30.6899
	motorcycle	StdConf	710	0.8961	1.0	0.9803	0.7253	57.4015
		EnsConf	660	0.8976	0.8923	0.8897	0.9101	75.3002
		CQRConf	710	0.8967	0.9215	0.9470	0.8087	54.7263
	bus	StdConf	3284	0.8991	1.0	0.9800	0.8653	59.8519
		EnsConf	3030	0.8997	–	0.8590	0.9158	67.8621
		CQRConf	3284	0.8981	0.9451	0.9379	0.8815	80.0147
	truck	StdConf	7632	0.8994	1.0	0.9854	0.8360	60.1658
		EnsConf	7543	0.8996	0.8017	0.8764	0.9165	66.5856
		CQRConf	7632	0.8997	0.9841	0.9633	0.8524	63.6431

Table 6. Metrics comparison of tested bounding box conformalization procedures across three datasets using the max-rank correction. Values are means over trials and per selected class. cov is expressed as a fraction and should be close to 0.9 (i.e. 90%), while MPIW is expressed in pixels.

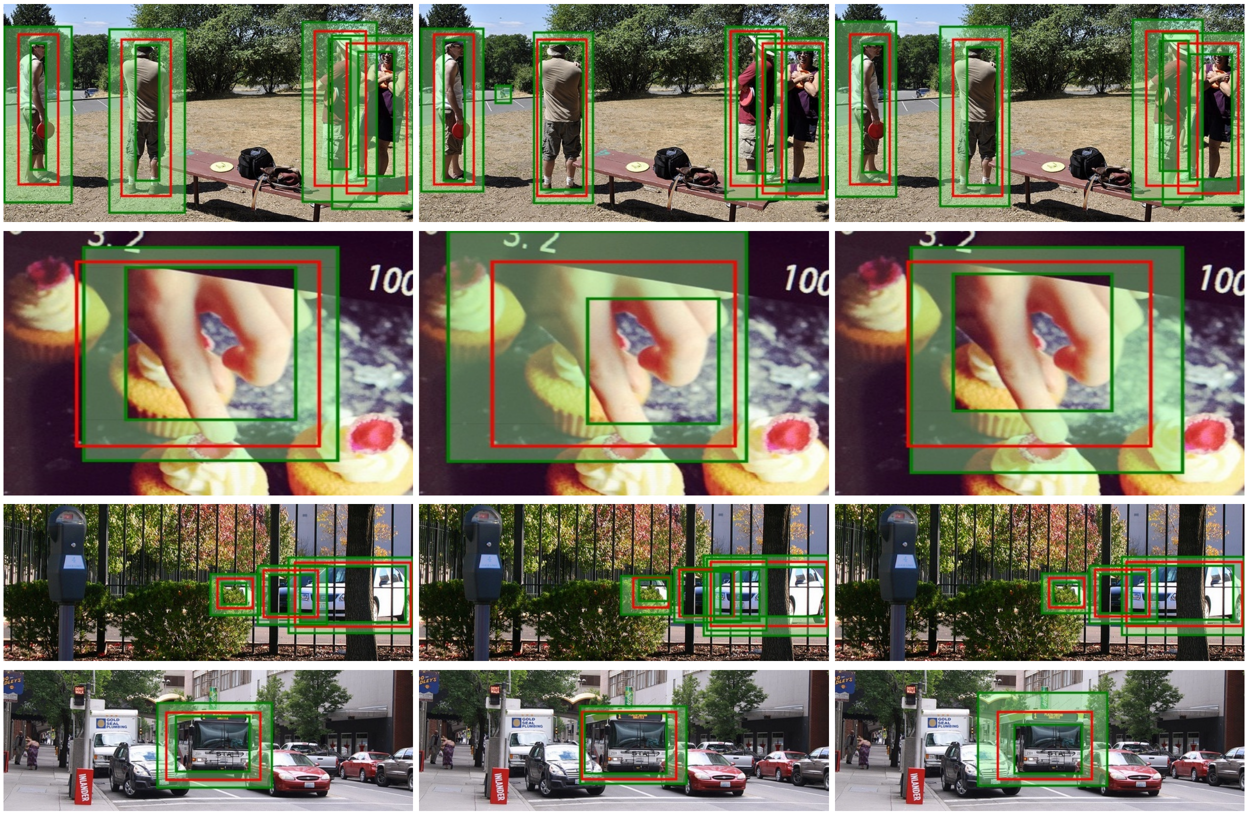

Figure 2. Examples for conformalized bounding boxes on `COCO-val` for classes {`person`, `car`, `bus`}. Left to right by column: `StdConf`, `EnsConf`, `CQRConf`. Ground truth boxes are in red, two-sided conformal prediction interval regions are shaded in green.

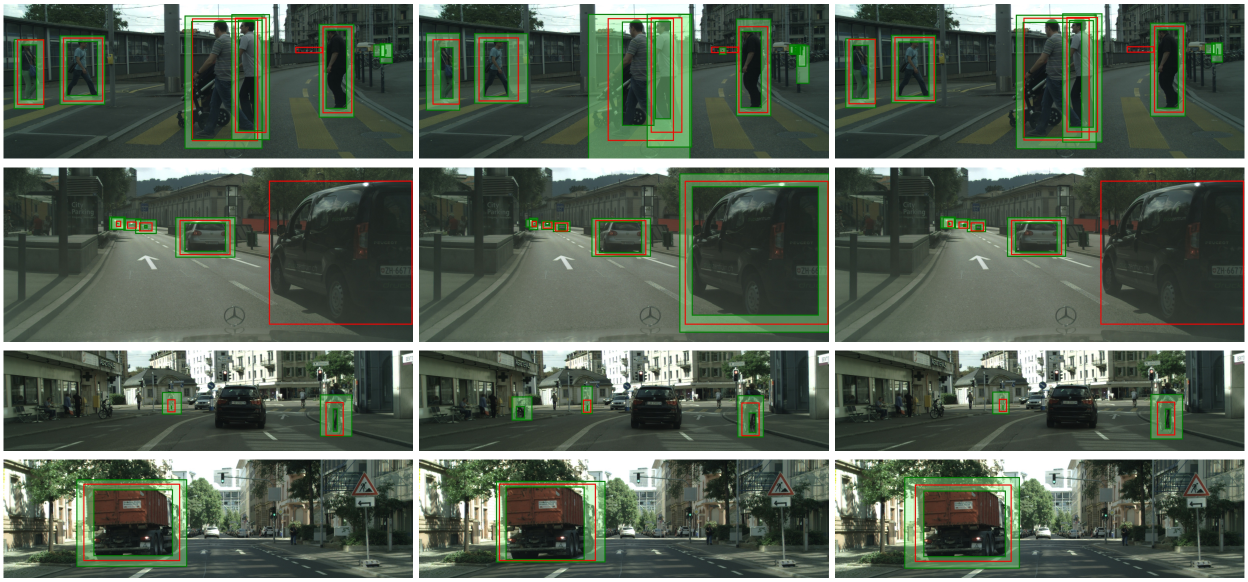

Figure 3. Examples for conformalized bounding boxes on `cityscapes` for classes {`person`, `car`, `bicycle`, `truck`}. Left to right by column: `StdConf`, `EnsConf`, `CQRConf`. Ground truth boxes are in red, two-sided conformal prediction interval regions are shaded in green.

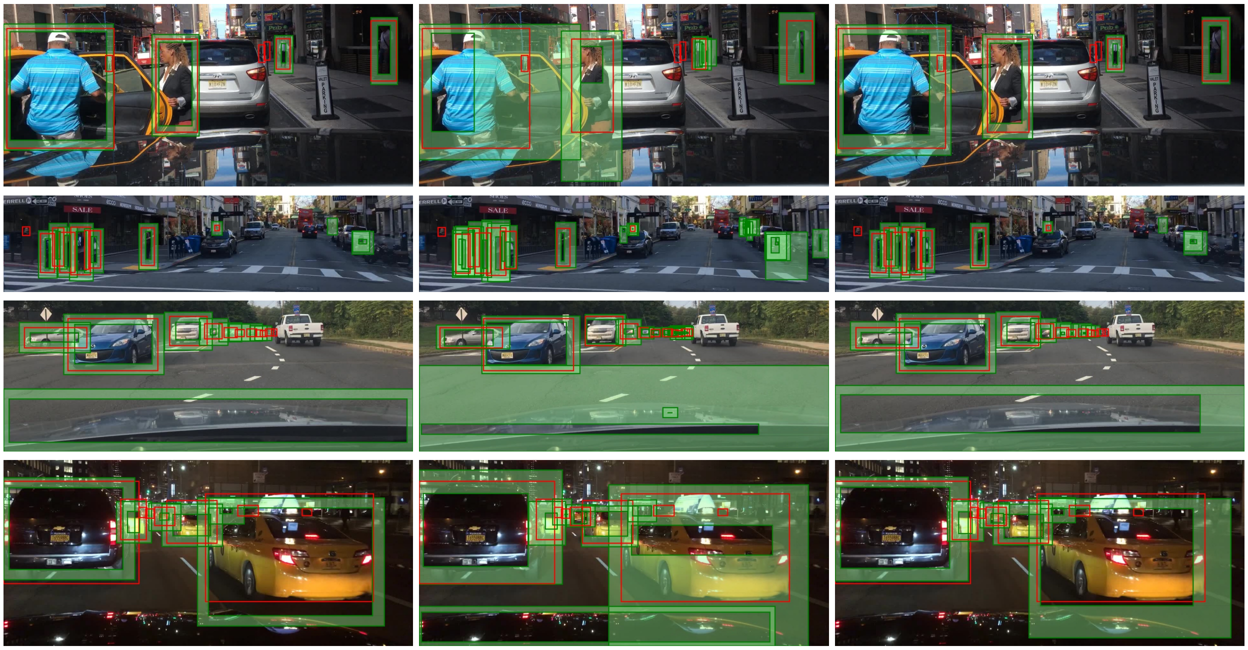

Figure 4. Examples for conformalized bounding boxes on `BDD100k-train` for classes {`person`, `car`}. Left to right by column: `StdConf`, `EnsConf`, `CQRConf`. Ground truth boxes are in red, two-sided conformal prediction interval regions are shaded in green.