Master Thesis

Uncertainty Calibration with Online Conformal Prediction in Neural Architecture Search: An Evaluation under the BANANAS Framework

Cheng Chen
(matriculation number 1662473)

July 31, 2025

Submitted to
Data and Web Science Group
Prof. Dr. Margret Keuper
University of Mannheim

Abstract

Some contents

Contents

List of Algorithms

List of Figures

List of Tables

1. Introduction

1.1. Motivation

1.2. Related Work

1.3. Contributions and Limitations

1.4. Outline

2. Literature Review

2.1. Neural Architecture Search (NAS)

2.1.1. Background

NAS is a subfield of AutoML.

Search Space Details will be introduced in 4

Search Strategy

Performance Evaluation

2.1.2. BANANAS

Bayesian Optimization with Neural Architectures for Neural Architecture Search (BANANAS)

BANANAS is an Bayesian Optimization based search strategy.

Bayesian optimization is a sequential decision-making process that seeks to find a global minimum x argminxX f(x) of an unknown black-box objective function f : X R over an input space X RD.

Give an introduction how Bayesian Optimization works. We list the five engineering decisions and review each field's related works. Maybe briefly cite Gaussian Process.

Architecture Encoding

Neural Predictor

Uncertainty Estimation

Acquisition Function

Acquisition Optimization

2.2. Uncertainty Quantification

Understanding uncertainty is important for real-world application of artificial intelligence, e.g., in autonomous driving, medical diagnosis.

2.2.1. Types of Uncertainty

- aleatory uncertainty (data uncertainty): uncertainty that arises due to inherent variations and randomness, and cannot be reduced by collecting more information
- epistemic uncertainty (model uncertainty): uncertainty that arises due to lack of knowledge, and can be reduced by collecting more information.

2.2.2. Alternative Uncertainty Estimation Methods

- Bayesian-based: e.g., Bayesian Neural Network
- Ensemble-based: e.g., Monte-Carlo dropout
- Bootstrapping

But these techniques are limited in several perspectives. First, quantifying uncertainty requires training models for several times, which means that the models cannot be applied for real-time prediction or in an online-learning setup. Second, some models are pre-trained and are only accessible via API. Besides, models (pre-)trained on certain datasets may struggle to generalize across different domains or contexts.

2.3. Conformal Prediction

2.3.1. Theoretical Background

Starting from i.i.d data, and provide an intuitive demonstration how the prediction interval is constructed (can add a figure illustrating why conformal prediction works, i.e., symmetry). From the most intuitive expression to the finite-sample adjusted expression.

Notation Then, relax the i.i.d assumption to exchangeability, and lay a formal definition of the conformal prediction. And list the most importance three ingredients of the conformal predictions.
- A trained predictor f
- A conformity score function s. The conformity score is an important engineering decision and has an impact on the size on the prediction set, i.e., the efficiency. The conformity score function can be either a negatively- or positively oriented, in which ... And it can be a random variable as well.
- A target coverage alpha

Marginal coverage is guaranteed regardless of the choices in dataset and black box model. Only the model predictions are required to apply the technique.

A Link to Statistical Testing (clarify the relationship between conformal prediction and hypothesis testing) In this video (22:21), it is explained the intuition why conformal prediction guarantees the coverage, which is quite similar to the spirit of hypothesis testing.

The coverage parameters which should be pre-set plays a similar role as the confidence interval in hypothesis testing. Conformal prediction is like hypothesis testing with hypotheses:

CP	Hypothesis Testing
(desired) Coverage level	Confidence level
Nominal error level (1 - Coverage level)	Significance level
The conformity score of the new instance	p-value (is an empirical term)

H0: test instance i conforms to the training instances.
H1: test instance i does not conform to the training instances.

2.3.2. Transductive Conformal Prediction

2.3.3. Extensions of Conformal Prediction

Since the transductive version of CP was first proposed in [2], several variants have been developed with different computational complexities, formal guarantees, and practical applications.

To address the aforementioned inefficient computation problem of TCP, Split Conformal Prediction (SCP), also known as Inductive Conformal Prediction (ICP), was first introduced in [4] by replacing the transductive inference with inductive inference. SCP aims to learn a general prediction rule about the data using the observed records. Then, this rule can be applied directly to obtain predictions when new data arrives in sequence, without re-using the training data and retraining the model repeatedly. The main concept involves splitting the data into two non-overlapping subsets, designated for training and calibration, respectively. A predictive model is fit exclusively on the training set, then non-conformity measures are computed on the calibration set to determine the prediction interval's width. Due to its simplicity and computational efficiency, SCP is one of the most commonly used technique in the CP family. We delve into methodological steps of SCP with pseudo-code in Section 3.2.1.

Limitations of split conformal predictions: - Distribution shift. The conformal prediction is built on the core assumption of exchangeability, which means the data points are identically distributed. However, this assumption is hard to meet in real-world application. For example, with time-series data this assumption is generally violated due to the temporal relationships. - Adaptivity. Once the conformity scores are computed on the calibration set, the decision threshold is settled and is applied to all test datapoints, regardless of the intrinsic complexity of the exact example. It is desirable that the threshold can adapts to the difficulty of the problem and produce a larger prediction interval/set on hard-to-solve example and smaller prediction interval/set on easy-to-solve example. This limitation echoes with the characteristic of Conformal Prediction that

the guaranteed coverage is only marginal over all datapoints but not conditional on a specific data points..

Variations of Conformal predictions have been proposed to overcome the limitations. There are three main streams: - find an empirical coverage rate which leads to the desired coverage level. For example, if the desired coverage rate is 90- find an efficient conformity score: Alternatively, [...] apply the conformal prediction in an online setting to dynamically incorporate the conformity score of new data points. - find suitable predictor: The trained predictor can be just a poor approximation of the real data generation process.

Besides, [...] proposes a CP algorithm that samples datapoints using Monte-Carlo sampling to approach the real distribution of labels in case the ground-truth is ambiguous and consequently cause a biased distribution in manually-annotated labels.

3. Methodology

To address the limitations of the Gaussian assumption in uncertainty estimation, this work introduces a new framework that integrates conformal prediction-based uncertainty calibration into the BANANAS framework in an online setting. An algorithm outlining the overall procedure is presented in Section 3.1, followed by detailed descriptions of each methodological step. Section 3.2 presents different conformal predictions algorithms to be explored. Next, methods for the estimation and evaluation of the conditional distribution of each candidate architecture are discussed in Section 3.3. Finally, in Section 3.4 we introduce how the calibrated distribution can be combined with different acquisition functions and acquisition search strategies.

3.1. The BANANAS–CP Framework

Refer to Section 2.1.2 for a detailed introduction of the original BANANAS algorithm. In this section, we emphasis the key ideas of the uncertainty calibration mechanism, as outlined in Step 1 to 6 of the inner iteration in Algorithm 1.

Bayesian optimization is a form of sequential decision-making task. In the applications of neural architecture search, the typical goal is to find the architecture that has the best evaluation performance on a fixed dataset under a given search budget. At each ietration t, a surrogate model is trained on all architectures evaluated at step $\{0, 1, 2..., t-1\}$, to predict the validation accuracy $f(a)$ of unseen architectures for the next search.

In the standard BANANAS setting, the surrogate model is an ensemble of m feed-forward neural networks (FNNs), typically $m = 5$. At iteration t, a set of candidate architectures is sampled, and a conditional Gaussian distribution is estimated for each candidate based on the ensemble predictions, as expressed below:

$$\hat{f}(a) \sim \mathcal{N}\left(\frac{1}{m}\sum_{i=1}^{m} f_i(a),\ \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(f_i(a) - \frac{1}{m}\sum_{j=1}^{m} f_j(a)\right)^2}\right) \tag{3.1}$$

where a denotes an architecture sampled from the search space, and $f_i(a)$ is the prediction of the i-th ensemble model for architecture a.

In the BANANAS–CP framework, a key distinction is that all previously evaluated architectures are split into a training set and a calibration set. Then, the surrogate model is trained exclusively using samples in the training set, while the calibration set is used to compute conformity scores for quantile calibration. In practice, at each iteration t, the surrogate model estimates a distribution $\hat{F}$ for an unseen architecture

Algorithm 1 The BANANAS–CP Framework

Input - NAS parameters: search space $\mathcal{A}$, evaluation dataset $\mathcal{D}$, exploration budget T, the number of initially sampled architectures t_0, acquisition function ϕ, surrogate model $\mathcal{M}$ that approximates the true objective function, function $f(\cdot)$ returning validation error of an architecture after training.

Input - Calibration parameters: a function $C(\cdot)$ to create calibration set, a non-conformity score function $s(\cdot)$, and an array of desired quantile levels q.

Draw t_0 architectures $\{a_1, a_2, ..., a_{t_0}\}$ uniformly at random from $\mathcal{A}$ and train each individual architecture on $\mathcal{D}$.

$\mathcal{A}_{t_0} \leftarrow \{a_1, a_2, ..., a_{t_0}\}$,

for t in $t_0 + 1, ..., T$ **do**

1. Apply $C(\cdot)$ and split all evaluated architectures into two disjoint datasets; use them as a training set $\mathcal{A}_{t,train}$, and a calibration set $\mathcal{A}_{t,cal}$.
2. Train the surrogate model $\mathcal{M}_t$ on $\{a, f(a)\}, a \in \mathcal{A}_{t,train}$ using the path encoding to represent each architecture.
3. Compute the conformity scores s on $\mathcal{A}_{t,cal}$.
4. Generate a set of candidate architectures from $\mathcal{A}$.
5. **for** each a_i in candidates **do**
   a) Estimate the quantile value for each level in q and calibrate with conformity scores computed in the previous step.
   b) Fit a distribution F_i based on the estimated quantile values.
   c) Compute the acquisition score $\phi(a_i)$.
6. **end for**
7. Denote a_t as the candidate architecture with maximum $\phi(a)$; evaluate $f(a_t)$.
8. $\mathcal{A}_t \leftarrow \mathcal{A}_{t-1} \cup \{a_t\}$

end for

Output: $a^* = \text{argmin}_{t=1,...,T} f(a_t)$

over its validation accuracy on the target dataset, typically either based on a specific distribution assumption or a probabilistic modeling approach, e.g., Bayesian Neural Network. Following the definition in [1], calibration means that for any quantile level $p \in [0, 1]$, the empirical fraction of data-points below the p-th percentile of the predicted distribution $\hat{F}$ should converge to p as the sample size goes to infinity. For example, if p = 80%, then the 80th percentile of $\hat{F}$ is set to the threshold value such that 80% of previously evaluated architectures fall below, thereby aligning with the empirical coverage. In an online setting, the objective of the calibration process can be defined as:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{I}\left\{y_t \leq Q_t(p)\right\} \to p \quad \text{for all } p \in [0,1] \tag{3.2}$$

as $t \to \infty$, where $\mathbb{I}$ is the indicator function and $Q_t(p)$ represents the distribution $\hat{F}$ in the format of quantile function.

Next, as in the standard Bayesian optimization process, the acquisition function picks the architecture for the next evaluation based on the conditional distribution of all sampled candidates.

3.2. Uncertainty Calibration Algorithms

As reviewed in Section 2.3, numerous conformal prediction algorithms have been proposed in recent research. This work identifies several approaches applicable in NAS for building a calibration set and computing non-conformity scores. This section provides an overview of these splitting strategies, as well as the conformity scoring functions that are commonly used for regression problems.

3.2.1. Split Conformal Prediction

To begin, a natural choice for a baseline calibration strategy is the Split Conformal Prediction (SCP). In this section, we start by introducing the standard SCP procedure, then proceed with the adaptions required to incorporate it into BANANAS-CP. Implementation steps of SCP can be summarized in Algorithm 2. For instance, imagine a regression task where the non-conformity level is measured by the absolute residual, i.e. $|y_i - \hat{y}(x_i)|$. In this case, the algorithm produces a prediction interval for the test point with a width of $[\hat{y}_{test} - \hat{q}\ ,\ \hat{y}_{test} + \hat{q}]$, where $\hat{q}$ is the conformity threshold.

In this work, we explore SCP in combination with different prediction algorithms. Firstly, we follow the settings in BANANAS and use an ensemble of five FNNs as the underlying surrogate model. In this case, note that the bounds of the prediction set as identified in Algorithm 2 should not be simply interpreted as the quantile values of a distribution, since the prediction algorithm does not directly model the τ-quantile of the variable Y, i.e., $Q_Y(\tau) = F_Y^{-1}(\tau) = \inf\{y\colon F_Y(y) \geq \tau\}$, with $\tau \in [0, 1]$ and F_Y is the CDF of the distribution. Thus, the ensemble predictor must be used in conjunction with a valid distribution assumption to obtain the quantile values. Motivated by the goal of achieving a completely distribution-agnostic solution, we next replace the ensemble

Algorithm 2 Split Conformal Prediction

Input: A set of observations $\{(x_i, y_i)\}_{i=1}^{n}$, a prediction algorithm $h(\cdot)$, a non-conformity measure $s(\cdot)$, nominal mis-coverage rate τ, fraction of data assigned to the training set p_{train}, test data x_{n+1}.

Output: a prediction set $\mathcal{C}_\tau(x_{n+1})$ that covers y_{n+1} with probability $1 - \tau$.

1: Allocate at random a proportion of p_{train} of the observations to the training set $\mathcal{D}_{train}$ and use the rest for calibration $\mathcal{D}_{cal}$.
2: Train the point predictor $h(\cdot)$ on $\mathcal{D}_{train}$.
3: Initialise a scoring set $S = \emptyset$
4: **for** (x_i, y_i) in $\mathcal{D}_{cal}$ **do**
$S \leftarrow S \cup \{s(h(x_i), y_i)\}$
5: **end for**
6: Return $\mathcal{C}_\tau(x_{n+1}) \leftarrow \{y \,|\, s((h(x_{n+1}), y) \leq q\}$, where q is the $\lceil (1-\tau)(n_s+1) \rceil$-th smallest value of S, with $n_s = |S|$.

model with a quantile regressor that directly models the quantiles of the distribution. In the remainder of this section, we discuss the configurations designated for each prediction algorithm.

Ensemble Predictor Following the settings in the original BANANAS, an ensemble by default consists of five neural networks. Each neural network is a fully-connected multi-layer perceptron with 20 layers of width 20, and is trained using the Adam optimizer with a learning rate of 0.01. The loss function for training is the mean absolute error (MAE). Similarly, we assume that the validation accuracy of each unseen candidate architecture a follows a Gaussian distribution, which is parameterized by the predictive mean ($\hat{\mu}$) and standard deviation ($\hat{\sigma}$) provided by the ensemble model, as demonstrated in equation 3.1. For a specific significance level α (suppose $\alpha < 0.5$), the central quantile interval can be written as:

$$\left[\hat{\mu} - \Phi^{-1}_{1-\alpha/2} \cdot \hat{\sigma} \ , \ \hat{\mu} + \Phi^{-1}_{1-\alpha/2} \cdot \hat{\sigma}\right] \tag{3.3}$$

where $\Phi^{-1}_{1-\alpha/2}$ denotes the $(1 - \frac{\alpha}{2})$-th quantile of the standard normal distribution.

Now, take a closer look at the formula 3.3 and recall the example based on the absolute residuals, which is presented earlier in this section. We observe that the confidence interval under the Gaussian assumption takes a close form to the prediction interval produced by CP when the conformity scoring function is chosen as:

$$s(\cdot) = \frac{|y_i - \hat{y}(x_i)|}{\hat{\sigma}(x_i)} \tag{3.4}$$

Hence, the bounds of the CP-derived prediction interval can be *approximately* interpreted as empirically calibrated quantile estimates under the Gaussian assumption, provided that τ is chosen appropriately. Note that the absolute residual can be seen as a special

case of equation 3.4, where the empirical standard deviation estimate is disregarded and set to 1. In fact, this scaled absolute residual (equation 3.4) is a widely adopted conformity score in practice. Ideally, we would like the CP-derived prediction interval also demonstrates local adaptivity, i.e., the prediction interval should have a larger width if the prediction task is difficult and vice versa. The scaled absolute residual accounts for heteroskedasticity and is able to adjust the width of the prediction band by multiplying the standard deviation estimate. In contrast, the band produced under a pure residual score has constant-width everywhere regardless of the input, which may limit its effectiveness in application. Therefore, in this work, we use the scaled absolute residual as the (non-)conformity scoring function for ensemble predictors, unless otherwise specified.

Quantile Regressor We now explain how a quantile regressor can be leveraged to build a probabilistic surrogate for Bayesian optimization, following previously established methods [5, 6].

We start by providing a brief introduction into quantile regression [3]. Let $(x, y) \sim F_{(X,Y)}$ denote data drawn from a joint distribution that is characterized by its cumulative distribution function F, the aim of the conditional quantile regression is to estimate a given quantile of the conditional distribution of Y given $X = x$. The conditional quantile function for α-quantile is:

$$Q(\alpha) = \inf \{y \in \mathbb{R} : \mathbb{P}(Y \leq y \mid X) \geq \alpha\} \tag{3.5}$$

and can be estimated by minimizing the Pinball loss [3]:

$$\ell_\alpha(y, \hat{y}) = \begin{cases} \alpha(y - \hat{y}), & \text{if } y \geq \hat{y} \\ (1 - \alpha)(\hat{y} - y), & \text{otherwise} \end{cases} \tag{3.6}$$

where $\hat{y}$ is the predicted quantile value. As illustrated in Figure 3.1, the Pinball loss is asymmetric and the intuition behind is that under-estimate and over-estimate receive different penalties across quantiles. For instance, if $\alpha = 0.9$, then we would expect that empirically 90% of observations should fall below the prediction. In this case, the loss function places a higher penalty for underestimate.

Quantile regression in the BANANAS-CP framework is implemented by training a dedicated neural network for each quantile level q_i in the array q as defined in Algorithm 1 using the corresponding Pinball loss $\ell_{q_i}(y, \hat{y})$.

While quantile regression can model the shape of any continuous distribution given enough data, the predictions are not guaranteed to be well calibrated in practice. In fact, it is not uncommon that quantile regression generates non-monotonic predictions, a phenomenon referred as quantile crossing. To address this issue, we apply a post-hoc calibration upon the predicted quantiles using the Conformal Quantile Regression (CQR) method from [5]. The authors develop a novel conformity score tailored for quantile estimation and the key idea of calibration is to offset the original quantile predictions by the conformity threshold that is computed on a calibration set.

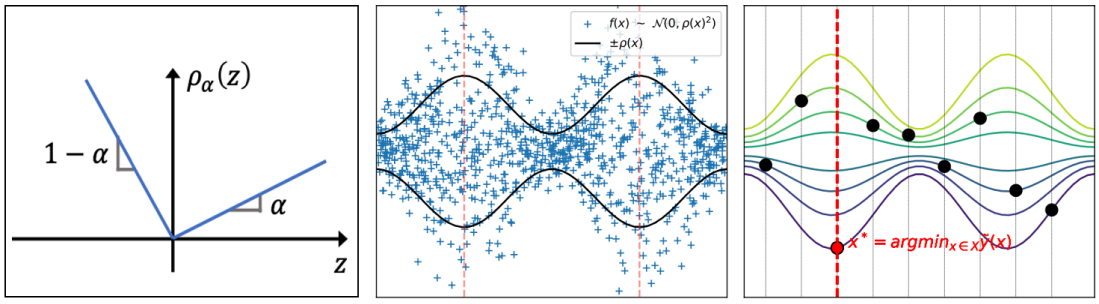

Figure 3.1: in the left is a visualization of the Pinball loss function, where $z = y - \hat{y}$ [5]; the middle displays samples from a synthetic heteroskedasticity function and the right illustrates the sampling procedure based on $|q| = 8$ predicted quantile [6].

In prior work, [6] employs CQR to obtain quantiles with robust coverage during hyperparameter tuning using Bayesian optimization. Specifically, to determine the next search, a set of candidates is first sampled uniformly at random, and then for each of those candidates a random quantile is simply picked and is treated as the acquisition scores (Figure 3.1). We follow their notation and interpretation in defining the conformity score:

$$E_i = \max\left\{\hat{q}_{\alpha_j}(x_i) - y_i,\ y_i - \hat{q}_{1-\alpha_j}(x_i)\right\} \tag{3.7}$$

Note that the sign of the score is positive when the target y_i is outside of the interval and negative when the target falls inside the predicted interval. This allows the conformity score to account for both overcoverage and undercoverage cases. In addition, the score amplitude always measures the distance to the closer quantile between $\hat{q}_{\alpha_j}(x_i)$ and $\hat{q}_{1-\alpha_j}(x_i)$ [5, 6].

3.2.2. Conformal Prediction with Cross-validation / Jackknife+

small datatset challenge

process for partitioning dataset and compute the non-conformity scores, which corresponds to Step 1 to 3 of the inner iteration in Algorithm 1.

combined with quantile regressor and ensemble

3.2.3. Conformal Prediction with Bootstrapping

3.3. Distribution Estimation

3.4. Acquisition Function and Search Strategy

.....

4. Dataset

To compare the performance of BANANAS–CP with the original BANANAS algorithms and assess the role of uncertainty calibration, we run experiments on the widely used benchmark dataset NAS-Bench-201.

5. Experiments and Results

5.1. Setup

5.2. Baseline

6. Conclusion

This chapter presents the central findings of this work as well as their critical discussion. Finally, it highlights limitations and corresponding opportunities for further research.

Bibliography

[1] Shachi Deshpande, Charles Marx, and Volodymyr Kuleshov. Online calibrated and conformal prediction improves bayesian optimization. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238, pages 7262–7273. PMLR, 2024.

[2] Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 148–155, Madison, WI, USA, 1998. Morgan Kaufmann.

[3] Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 33–50, 1978.

[4] Harris Papadopoulos, Kostas Proedrou, Vladimir Vovk, and Alexander Gammerman. Inductive confidence machines for regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Machine Learning: ECML 2002*, volume 2430 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2002.

[5] Yaniv Romano, Evan Patterson, and Emmanuel J Candès. Conformalized quantile regression. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[6] David Salinas, Jacek Gołebiowski, Aaron Klein, Matthias Seeger, and Cédric Archambeau. Optimizing hyperparameters with conformal quantile regression. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

A. Additional Experimental Results

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
ChatGPT	Debugging LaTeX syntax	Equation/Formula	+

Unterschrift
Mannheim, den 31.07.2025