Qualcomm AI research

Approximating Full Conformal Prediction for Neural Network Regression with Gauss-Newton Influence

Dharmesh Tailor[1], Alvaro H.C. Correia[2], Eric Nalisnick[3], Christos Louizos[2]

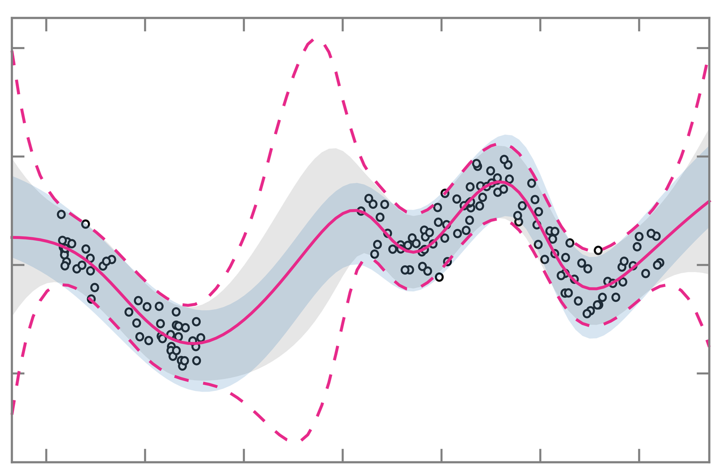
[1]University of Amsterdam, [2]Qualcomm AI Research, [3]Johns Hopkins University

[1]d.v.tailor@uva.nl
[2]{acorreia,clouizos}@qti.qualcomm.com
[3]nalisnick@jhu.edu

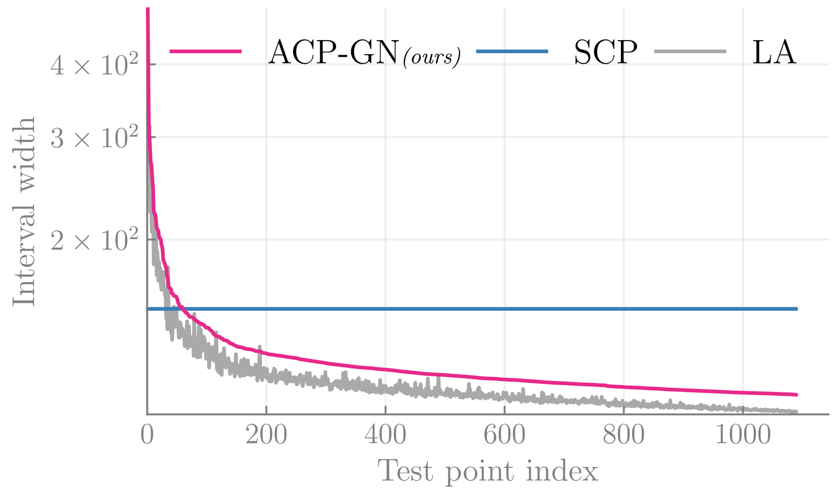
Conformal Prediction (CP) is a popular uncertainty quantification framework that comes in two flavours:

- **Split-CP**: splits the data into training and calibration sets; compute-efficient but data-inefficient.
- **Full-CP**: retrains the model for each new test point; data-efficient but compute-inefficient.

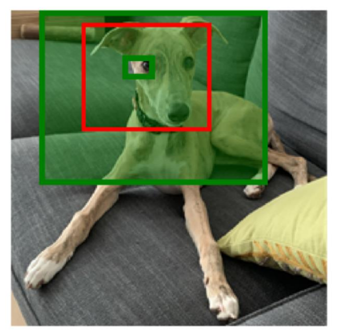
We propose a new conformal regression method that is both compute- and data-efficient. Ours is an **approximate full CP** method, where the model is trained only once. We use **Gauss-Newton influence** to perturb the model parameters locally, simulating the effect of retraining. By linearizing the neural network, we exploit a computational shortcut from **conformal least-squares** to avoid the infinite search space over labels. We demonstrate our method on benchmark **regression** problems and **bounding box localization**, with promising results in limited-data regimes.



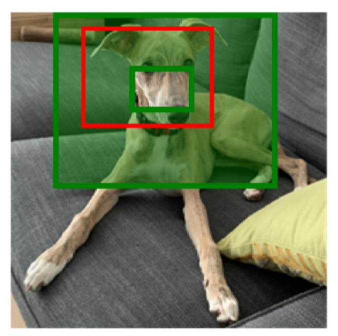
Our approximate full-CP via Gauss-Newton influence (ACP-GN) produces adaptive intervals (bottom)—like Bayes via Laplace approximation (LA)—while satisfying coverage as seen in the high-overlap with split-CP (SCP) close to the data (top).



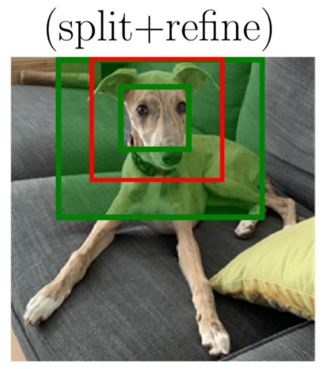
SCP



ACP-GN



ACP-GN (split+refine)



Object localization with bounding box prediction (multi-output regression). Two-sided prediction regions shown in green and NN prediction in red.

Conformal Prediction

Distribution-free UQ in the form of prediction intervals $C(X_{N+1})$ that, for a user-defined miscoverage rate α, satisfies marginal coverage

$$\mathbb{P}(y_{N+1} \in C_\alpha(\mathbf{x}_{N+1})) \geq 1 - \alpha$$

where $\mathbf{x}_{N+1}$ is the test point, y_{N+1} its (unknown) label, and $\alpha \in [0,1]$ the user-defined error rate.

How to Construct Prediction Intervals

Given model f_{θ^*}, and training data $\{(x_i, y_i)\}_{i=1}^N$

1. Define a nonconformity score, e.g., residual
$$R_i = R(x_i, y_i) = |y_i - f_{\theta^*}(x_i)|$$

2. Construct prediction intervals as
$$C_\alpha(\mathbf{x}_{N+1}) = \{y \in \mathcal{Y} : \pi(y) \leq \lceil (1-\alpha)(N+1) \rceil\}$$

where $\pi(y) = \sum_{i=1}^{N+1} \mathbb{1}\{R_i < R(\mathbf{x}_{N+1}, y)\}$ is the rank of $R(\mathbf{x}_{N+1}, y)$ among the other N residuals. If the set of $\{R_i\}_{i=1}^{N+1}$ is exchangeable, the CP guarantee holds.

Algorithm 1: Standard Full-CP.

for each test point $\mathbf{x}_{N+1}$ **do**
  for each y in a given grid **do**
    optimize $\boldsymbol{\theta}_*^+(y)$ as in Eq. (4)
    $R_{N+1}(y) = |y - f_{N+1}(\boldsymbol{\theta}_*^+(y))|$
    for $i \in \{1, \dots N\}$ **do**
      $R_i(y) = |y_i - f_i(\boldsymbol{\theta}_*^+(y))|$
    $\pi(y) = \sum_{i=1}^{N+1} \mathbb{1}\{R_i(y) \leq R_{N+1}(y)\}$
    if $\pi(y) \leq \lceil (1-\alpha)(N+1) \rceil$ **then**
      include y in $C_\alpha(\mathbf{x}_{N+1})$

Algorithm 2: ACP-GN (ours).

optimize $\boldsymbol{\theta}_*$ as in Eq. (1)
for each test point $\mathbf{x}_{N+1}$ **do**
  compute a_{N+1}, b_{N+1} as in Eq. (14)
  for $i \in \{1, \dots N\}$ **do**
    compute a_i, b_i as in Eq. (13)
    if $b_{N+1} - b_i > 0$ **then**
      $l_i = u_i = (a_i - a_{N+1})/(b_{N+1} - b_i)$
    else
      $l_i = -\infty$ and $u_i = \infty$
  sort $\{l_i\}_{i=1}^N$ and $\{u_i\}_{i=1}^N$ in ascending order
  $C_\alpha(\mathbf{x}_{N+1}) = [l_{(\lfloor (N+1)(\alpha/2) \rfloor)}, u_{(\lceil (N+1)(1-\alpha/2) \rceil)}]$

Computational Challenges

1. Search space: One must consider all $y \in \mathcal{Y}$ labels for the test point $\mathbf{x}_{N+1}$ in the construction of $C_\alpha(\mathbf{x}_{N+1})$.

2. Ensuring exchangeability: The computation of residuals must be symmetric on all $\{(\mathbf{x}_i, y_i)\}_{i=1}^N \cup (\mathbf{x}_{N+1}, y)$ points.

- Split-CP: $f_{\theta^*}(\cdot)$ trained in another dataset and kept fixed, hence data-inefficient.
- Full-CP: $f_{\theta^*}(\cdot)$ is retrained on augmented dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N \cup (\mathbf{x}_{N+1}, y)$ where y is a postulated label.

Whilst [Martinez et al., 2023] addressed (2) using influence function, they did not consider regression settings so (1) is left unaddressed.

We address both challenges in our method.

Validity of ACP-GN

- CP guarantee not assured since retraining step is locally approximated (train + test points not treated symmetrically)
- Bounds on approximation error exist for Newton-step influence
- Proposed variant with linear model "refinement" and train-calibration splits

Baseline Methods:

- Linearized Laplace (LA)
- Split-CP (SCP)
- Conformalized residual fitting (CRF)
- Conformalized quantile regression (CQR)

Our Methods:

- ACP-GN
- ACP-GN (split+refine)*
  see validity section
- SCP-GN: approximate in-sample score using influence

Our Method (ACP-GN)

Approximate FCP via Gauss-Newton Influence

1. Solution to infinite search space

Linearization of the network about pretrained model parameters $\boldsymbol{\theta}^*$

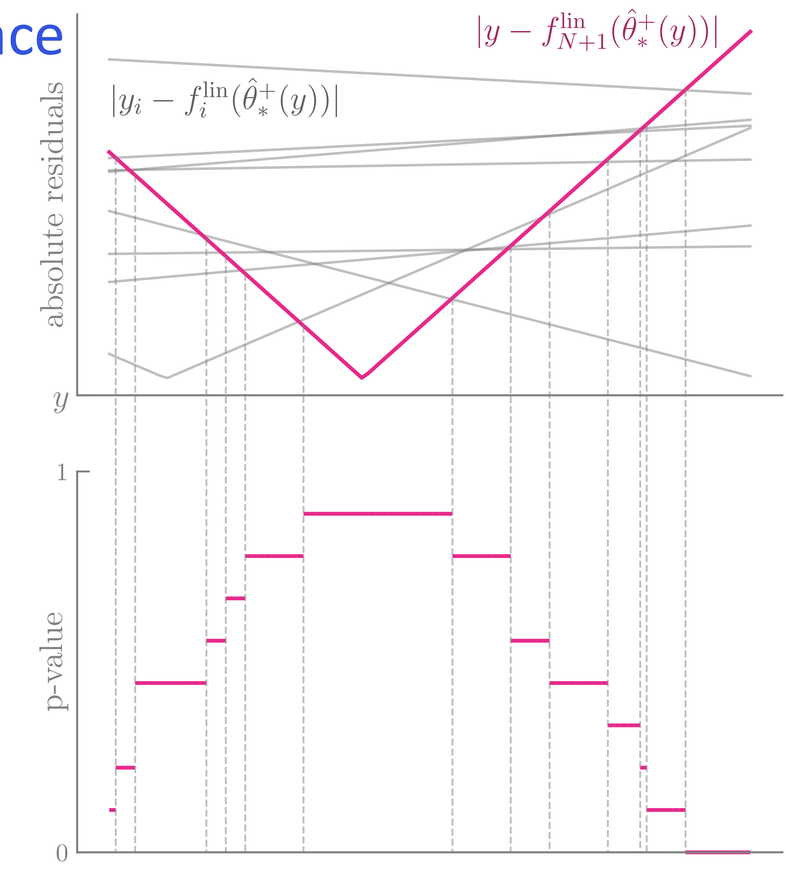
$$f_i(\boldsymbol{\theta}) \approx f_i^{lin}(\boldsymbol{\theta})$$
$$= f_i(\boldsymbol{\theta}_*) + \nabla_{\boldsymbol{\theta}} f_i(\boldsymbol{\theta}_*)^T (\boldsymbol{\theta} - \boldsymbol{\theta}_*)$$

In a linear model, we only need to consider a few values for postulated y.

2. Solution to retraining

Newton-step influence [Pregibon, 1981] with Gauss-Newton approximation to get an efficient estimate

$$\boldsymbol{\theta}_*^+(y) \approx \boldsymbol{\theta}_* + \frac{e_{N+1}(y)}{1 + h_{N+1}} \mathbf{H}_{\mathrm{GN}}^{-1} \nabla_{\boldsymbol{\theta}} f_{N+1}(\boldsymbol{\theta}_*)$$



In a linear model, the ranks (p-values) only change when the residual lines cross (test point residual in pink).

Approximate scores by piecewise linear function of postulated label

$$R_i(y) \approx |a_i + b_i y| \qquad R_{N+1}(y) \approx |a_{N+1} + b_{N+1} y|$$

Obtain exact form of prediction set by applying ridge regression confidence machine procedure on $\{(a_i, b_i)\}_{i=1}^{N+1}$

$$\pi(y) = \sum_{i=1}^{N+1} \mathbb{1}\{y \in S_i\} \qquad S_i = \{y : |a_i + b_i y| \leq |a_{N+1} + b_{N+1} y|\}$$

Table: Our ACP-GN almost always gives the tightest intervals in limited-data regimes whilst satisfying the target coverage. On larger datasets (not shown), ACP-GN remains competitive on efficiency compared with other conformal methods but can sometimes miscover. As a remedy, we propose two variants inspired by ACP-GN.

		Avg. Width			Avg. Coverage		
		90%	95%	99%	90%	95%	99%
yacht N=308 I=6	**LA**	1.690±0.017	2.014±0.020	2.647±0.027	88.73±0.61 (✓)	90.78±0.59 (✗)	93.89±0.60 (✗)
	SCP	2.553±0.093	4.001±0.115	10.018±0.361	89.56±0.66 (✓)	94.07±0.39 (✓)	99.32±0.08 (✓)
	CRF	2.526±0.092	3.947±0.115	9.674±0.294	89.53±0.64 (✓)	94.10±0.38 (✓)	99.29±0.10 (✓)
	CQR	4.090±0.105	5.845±0.187	18.650±0.484	89.94±0.42 (✓)	94.42±0.32 (✓)	99.02±0.17 (✓)
	ACP-GN	**1.594**±0.016	**2.385**±0.029	**6.915**±0.067	87.36±0.58 (✓)	92.56±0.68 (✓)	99.03±0.11 (✓)
	SCP-GN	2.270±0.086	3.349±0.098	**7.216**±0.254	89.85±0.51 (✓)	94.91±0.32 (✓)	99.19±0.15 (✓)
	ACP-GN (split + refine)	1.993±0.020	2.954±0.037	**7.307**±0.178	89.35±0.62 (✓)	94.90±0.51 (✓)	99.45±0.08 (✓)
boston N=506 I=13	**LA**	9.398±0.046	**11.199**±0.055	14.718±0.072	91.24±0.31 (✓)	94.34±0.22 (✓)	97.53±0.11 (✗)
	SCP	10.635±0.123	14.509±0.171	36.272±1.847	89.56±0.42 (✓)	94.64±0.32 (✓)	99.11±0.13 (✓)
	CRF	11.932±0.605	16.073±0.862	40.690±3.333	90.01±0.33 (✓)	94.77±0.22 (✓)	99.30±0.08 (✓)
	CQR	11.692±0.129	15.115±0.213	31.628±1.822	90.10±0.33 (✓)	95.12±0.24 (✓)	99.07±0.14 (✓)
	ACP-GN	**9.182**±0.046	12.111±0.038	**20.512**±0.057	90.64±0.26 (✓)	95.49±0.16 (✓)	99.11±0.08 (✓)
	SCP-GN	10.301±0.089	13.418±0.151	24.714±0.865	89.52±0.50 (✓)	94.82±0.32 (✓)	99.05±0.12 (✓)
	ACP-GN (split + refine)	13.103±0.072	16.729±0.134	27.561±0.445	90.12±0.26 (✓)	95.41±0.20 (✓)	99.27±0.10 (✓)
energy N=768 I=8	**LA**	1.502±0.006	1.790±0.007	2.353±0.009	88.96±0.35 (✓)	92.92±0.33 (✗)	96.95±0.23 (✗)
	SCP	1.942±0.032	2.486±0.046	3.772±0.093	89.44±0.28 (✓)	94.80±0.20 (✓)	99.18±0.08 (✓)
	CRF	1.923±0.031	2.454±0.046	3.728±0.092	89.39±0.28 (✓)	94.78±0.22 (✓)	99.14±0.08 (✓)
	CQR	4.670±0.030	5.139±0.029	6.438±0.120	90.08±0.26 (✓)	95.24±0.21 (✓)	98.96±0.09 (✓)
	ACP-GN	**1.462**±0.006	**1.884**±0.008	**3.076**±0.015	88.28±0.33 (✓)	93.69±0.33 (✓)	98.88±0.11 (✓)
	SCP-GN	1.911±0.029	2.449±0.044	3.609±0.071	89.69±0.29 (✓)	94.79±0.18 (✓)	99.21±0.09 (✓)
	ACP-GN (split + refine)	1.745±0.016	2.174±0.021	3.300±0.045	90.54±0.25 (✓)	94.96±0.22 (✓)	99.18±0.10 (✓)