Learning-based Estimation and Uncertainty Quantification of Nonlinear (Inverse) Operators

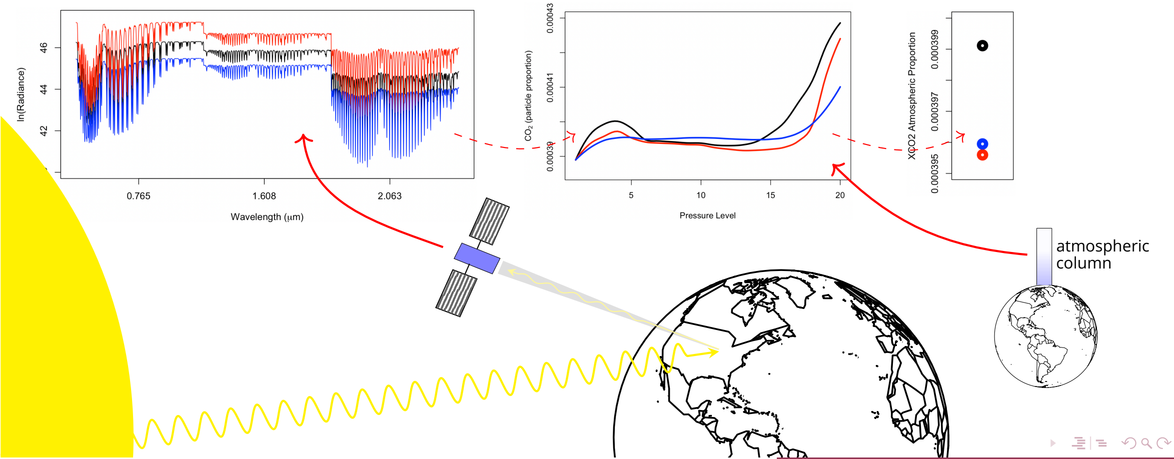
Rick Lucas

Emily L. Kang, University of Cincinnati, U.S.
Jonathan Hobbs, NASA, Jet Propulsion Laboratory, U.S.

Table of Contents

1. Problem & Notation
2. Statistical Operator Learning
3. UQ via Conformalized Quantile Regression
4. Numerical Results
5. Conclusion

The Challenge



OCO-2 Data & Notation

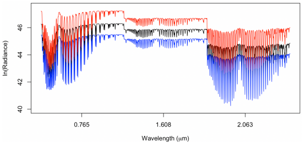
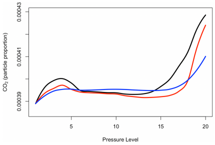
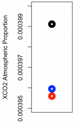
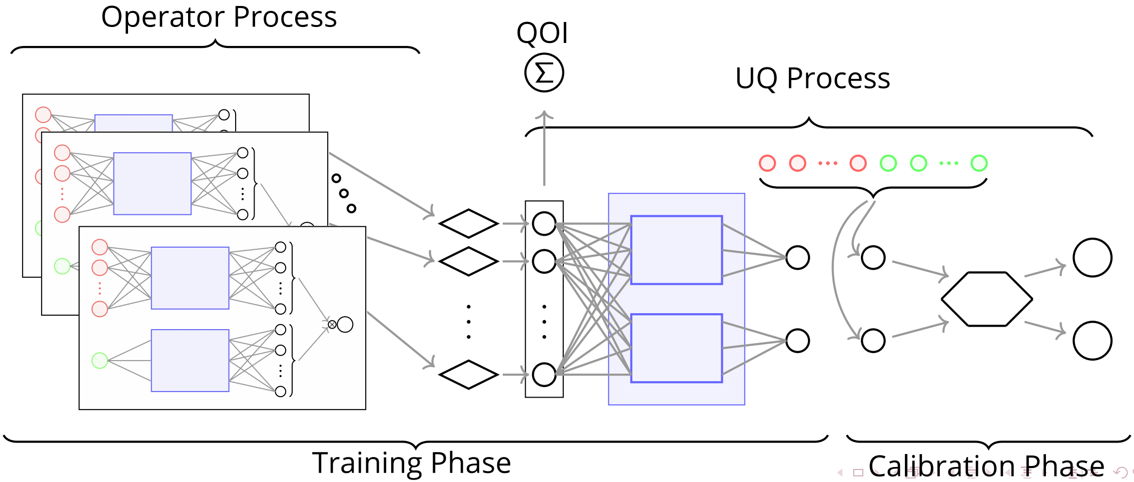
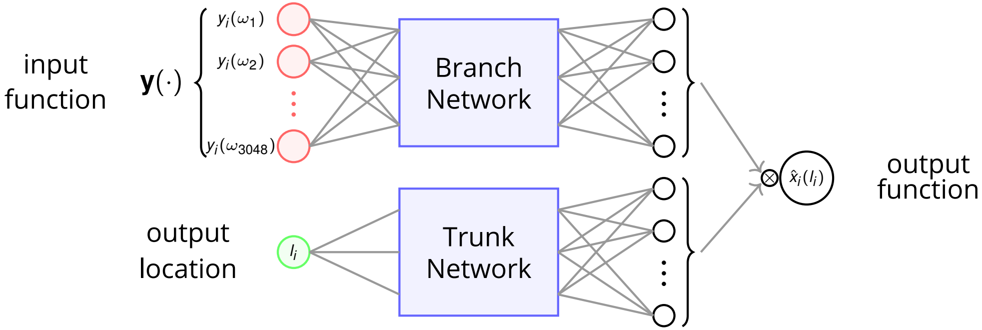
	Radiance function		CO_2 profile function		X_{CO_2} value
					
		$\mathcal{G}(y(\cdot))$		$\mathcal{H}(x(\cdot))$	
System	$y(\omega)$		$x(l)$		$\overset{*}{x}_{X_{CO_2}}$
Data	$\big(y_i(\omega_1), \ldots, y_i(\omega_{3048})\big)$	$\mathcal{G}(y_i)(l_j)$	$\big(x_i(l_1), \ldots, x_i(l_{20})\big)$	$\big(h_i(l_1), \ldots, h_i(l_{20})\big)^\top$	$\overset{*}{x}_{i,X_{CO_2}}$

Figure: OCO-2 Variable Notation

Schematic


Operator Process
QOI
Σ
UQ Process
Training Phase
Calibration Phase

Operator Learning



Operator Learning

- (Chen and Chen 1995)
  - Investigates the question: can "a neural network model...be used to approximate the output of some dynamic system as a whole"?
  - Provides two universal approximation theorems; one for continuous functionals and one for continuous operators.
  - The output function representation, $\mathbf{x}(l)$, is modelled as:

$$\sum_{k=1}^{q} \left\{ \left[\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k y(\omega_j) + \theta_i^k \right) \right] \cdot \left[\sigma \left(\varphi_k \cdot l + \zeta_k \right) \right] \right\} \approx \mathcal{G}(\mathbf{y})(l).$$

- (Lu et al. 2021) – DeepONet
  - Highlights potential in Chen and Chen and improves upon success by:
    1. proposing specific architecture to achieve smaller total error,
    2. extending the shallow NN structure to a deep NN structure for greater flexibility,
    3. and develops computation more efficient alternative single branch network

Operator Learning – continued

- (Lu et al. 2021) – DeepONet
  - The output function, $\mathbf{x}(l)$, representation:

$$\sum_{k=1}^{q}\left\{\underbrace{\left[\sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{ij}^k y(\omega_j) + \theta_i^k\right) + \theta_k\right]}_{=: b_k \ (k^{\text{th}} \text{ branch})} \cdot \underbrace{\left[\sigma\left(\varphi_k \cdot l + \zeta_k\right)\right]}_{=: t_k \ (k^{\text{th}} \text{ trunk})}\right\} + b_0 = \sum_{k=1}^{q} b_k t_k + b_0 \approx \mathcal{G}(\mathbf{y})(l)$$

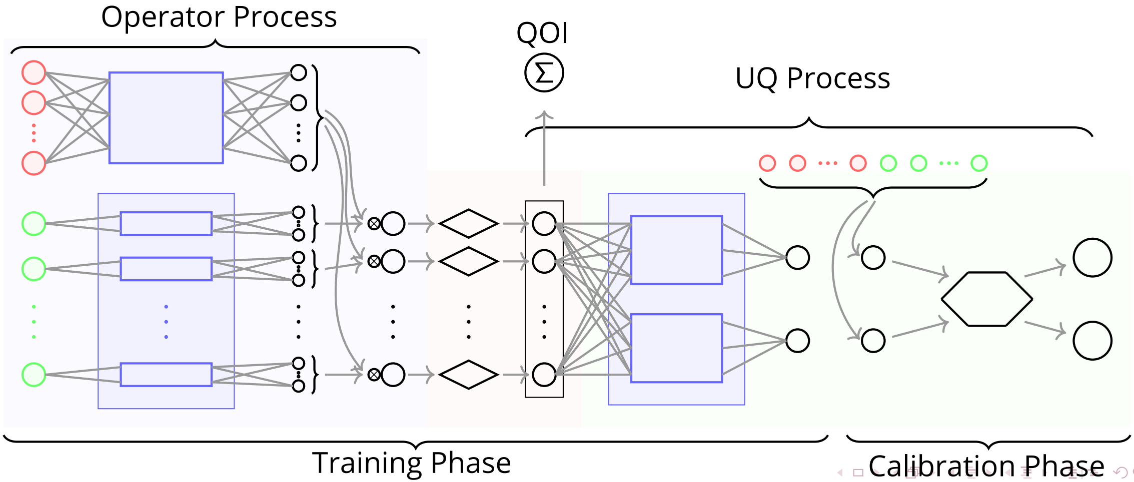
  - Given training data

$$\begin{aligned}\text{Given training data } &((y_1(\omega_1), \ldots, y_1(\omega_{3048}), x_1(l_1), \ldots, x_1(l_{20})),\\ &(y_2(\omega_1), \ldots, y_2(\omega_{3048}), x_2(l_1), \ldots, x_2(l_{20})), \ldots,\\ &(y_N(\omega_1), \ldots, y_N(\omega_{3048}), x_N(l_1), \ldots, x_N(l_{20}))),\end{aligned}$$

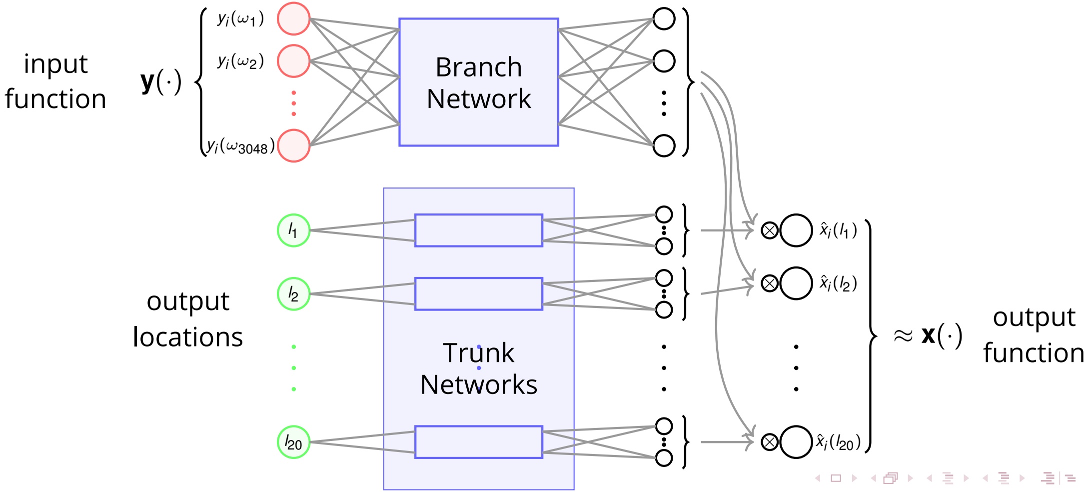
we utilize the standard MSE loss function to estimate network parameters:

$$\min_{c_i^k, \xi_{ij}^k, \theta_i^k, \theta_k, \varphi_k, \zeta_k, b_0} \frac{1}{20N} \sum_{s=1}^{N} \sum_{r=1}^{20} \left(x_s(l_r) - \hat{\mathcal{G}}(y_s)(l_r)\right)^2.$$

Schematic


Operator Process
QOI
Σ
UQ Process
Training Phase
Calibration Phase

Operator Learning



Operator Learning – continued

- Deep Operator Network Attempt
  - Given training data:

$$\begin{aligned}&((y_1(\omega_1),\dots,y_1(\omega_{3048}),x_1(l_1),\dots,x_1(l_{20})),\\&(y_2(\omega_1),\dots,y_2(\omega_{3048}),x_2(l_1),\dots,x_2(l_{20})),\dots,\\&(y_N(\omega_1),\dots,y_N(\omega_{3048}),x_N(l_1),\dots,x_N(l_{20}))),\end{aligned}$$

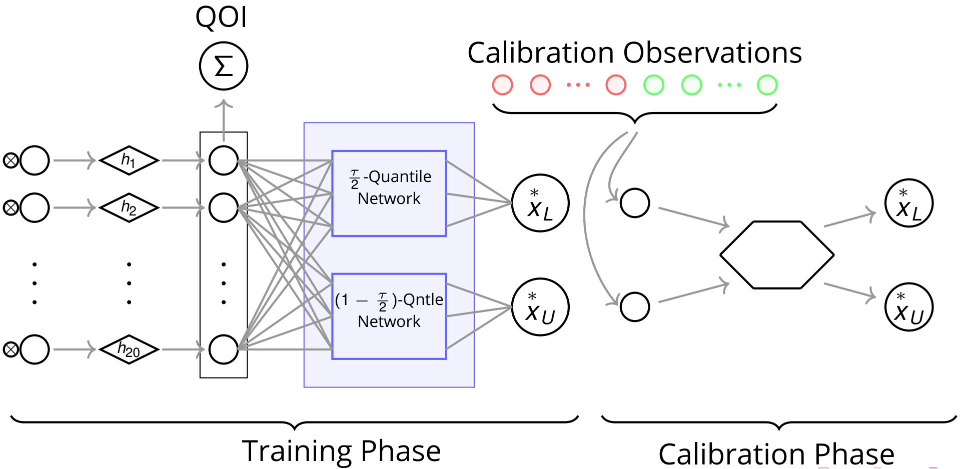
  - Output function **x** representation is a 1×20 vector with the following r^{th} entry:

$$\sum_{k=1}^{q}\left\{\left[\sum_{i=1}^{n}c_i^k\sigma\left(\sum_{j=1}^{m}\xi_{ij}^k y(\omega_j)+\theta_i^k\right)+\theta_k\right]\cdot\left[\sigma\left(\varphi_k^r\cdot l_r+\zeta_k^r\right)\right]\right\}+b_0\approx\mathcal{G}(\mathbf{y})(l_r).$$

  - With loss function:

$$\min_{c_i^k,\xi_{ij}^k,\theta_i^k,\theta_k,\varphi_k^r,\zeta_k^r,b_0}\frac{1}{N}\sum_{s=1}^{N}\sum_{r=1}^{20}\left(x_s(l_r)-\hat{\mathcal{G}}(y_s)(l_r)\right)^2.$$

UQ Schematic



Conformal Prediction

Geared toward machine learning models, conformal prediction provides a framework to supplying confidence intervals (or sets) to individual predictions.

The key behind conformal prediction is in establishing a measure of conformity for new data points that can be used to provide intervals that contain true outcomes with a specified probability.

Goals

1. Provide prediction intervals containing the true outcome with a specified probability.
2. Provide prediction intervals that guarantee valid coverage probabilities, regardless of the data distribution.
3. Provide prediction intervals for any pre-trained model.

(Split or Inductive) Conformal Prediction

Split conformal prediction uses a portion of the training dataset to calibrate intervals of uncertainty (i.e., splitting training data into calibration and proper training data). This is done by comparing the new data points to the training data and determining how similar they are in terms of their predicted outcomes.

Procedure

1. Split training dataset into proper training and calibration datasets.
2. Identify a heuristic notion of uncertainty (using proper training set).
3. Define the score function $s(\hat{\mathcal{G}}(y), \overset{*}{x}) \in \mathbb{R}$.
4. Compute $\hat{q}_c$, the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$-quantile of the calibration data scores $s(\hat{\mathcal{G}}(y)_1, \overset{*}{x}_n), \ldots, s(\hat{\mathcal{G}}(y)_n, \overset{*}{x}_n)$.
5. Form the prediction intervals $\mathcal{C}(\hat{\mathcal{G}}(y)_{\text{test}}) = \{\hat{\overset{*}{x}}_{\text{test}} \mid s(\hat{\mathcal{G}}(y)_{\text{test}}, \hat{\overset{*}{x}}_{\text{test}}) \geq \hat{q}_c\}$.

Conformal Prediction

Uncertainty Concept & Method

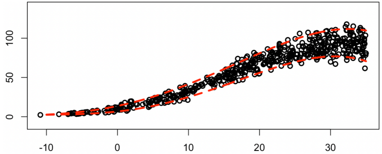
- Intent – to represent a system's uncertainty.
- Pitfall – failure to adapt to a dataset's variability in uncertainty. That is, we want a procedure that applies the desired coverage, conditionally.

Score Function

- Intent – to functionalize some concept of uncertainty that induces larger intervals when the model is uncertain.

Quantile Regression via ANN

To motivate a variable-width conformal prediction interval, we use quantile regression as our uncertainty concept.



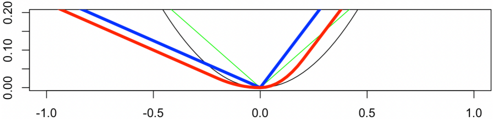
Since conformal prediction can utilize any form of quantile regression, we extend the use of NN models used in the operator learning phase through the initial prediction interval learning phase.

The standard τ^{th}-quantile loss function is not differentiable at 0:

$$\rho_\tau(\varepsilon) = \begin{cases} \tau|\varepsilon| & \text{if } \varepsilon \geq 0 \\ (1-\tau)|\varepsilon| & \text{if } \varepsilon < 0. \end{cases}$$

To remedy this, we estimate the quantile loss by replacing ε with its Huber function (Chen 2007) (i.e., $\rho_\tau(h_\delta(\varepsilon)) \approx \rho_\tau(\varepsilon)$):

$$h(\varepsilon) = \begin{cases} \varepsilon^2/(2\delta) & \text{if } 0 \leq |\varepsilon| \leq \delta \\ |\varepsilon| - \delta/2 & \text{if } |\varepsilon| > \delta. \end{cases}$$

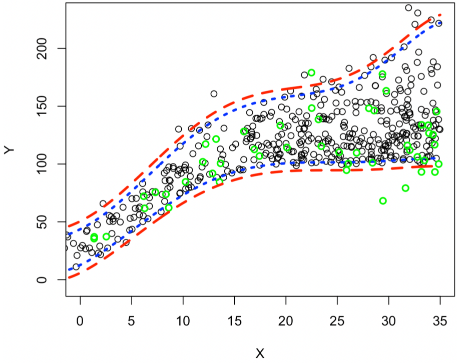


Conformalized Quantile Regression

Procedure

1. Split training data into proper training and calibration data.
2. Model uncertainty in the form of Quantile Regression bands $[\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}]$ (training data).
3. Define the score function $s_i = \max\{\hat{q}_{\alpha/2}(\hat{\mathcal{G}}(y)_i) - \overset{*}{x}_i, \overset{*}{x}_i - \hat{q}_{1-\alpha/2}(\hat{\mathcal{G}}(y)_i)\}$.
4. Compute $\hat{q}_c$, the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$-quantile of the calibration data scores $s(\hat{\mathcal{G}}(y)_1, \overset{*}{x}_n), \ldots, s(\hat{\mathcal{G}}(y)_n, \overset{*}{x}_n)$.
5. Form the prediction intervals

$$\mathcal{C}(\hat{\mathcal{G}}(y)_{\text{test}}) = \left[\hat{\overset{*}{x}}_{\text{test}} - \hat{q}_c \,,\ \hat{\overset{*}{x}}_{\text{test}} - \hat{q}_c\right].$$

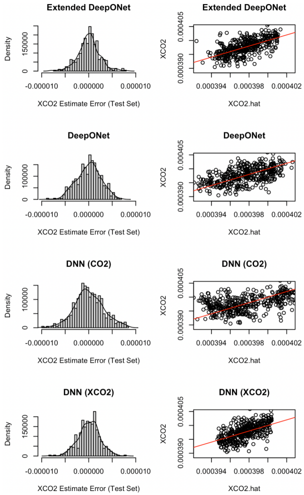


Numerical Results – Operator Learning

Accuracy measures.

X_{CO_2}	rMSE	MAPE	MAE	Time
ExDeepONet	2.282e-6	4.372e-3	1.738e-6	24.5
DeepONet	2.558e-6	5.146e-3	2.046e-6	14.7
DeepONet Rv	2.268e-6	4.276e-3	1.700e-6	
DNN (CO_2)	3.084e-6	6.110e-3	2.428e-6	8.6
DNN (X_{CO_2})	2.302e-6	4.552e-3	1.810e-6	1.0

CO_2	rMSE	MAPE	MAE
ExDeepONet	7.667e-6	1.037e-2	4.118e-6
DeepONet	7.808e-6	1.194e-2	4.737e-6
DeepONet Rv	7.424e-6	1.048e-2	4.159e-6
DNN (CO_2)	9.495e-6	1.455e-2	5.777e-6



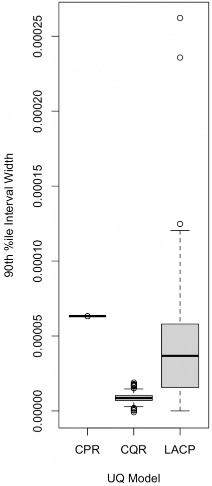
Numerical Results – Uncertainty Quantification

All models were picked up on the same operator learned model.

Method	UQ	Score Fcn
Conformalized Quantile Regression	$[\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}]$	$\max\{\hat{q}_{\alpha/2}(\hat{\mathcal{G}}(y)_i) - \overset{*}{x}_i, \overset{*}{x}_i - \hat{q}_{1-\alpha/}$
Conformal Prediction (residuals)	$\lvert\overset{*}{x} - \hat{f}(\hat{\mathcal{G}}(y_i))\rvert$	$\lvert\overset{*}{x} - \hat{f}(\hat{\mathcal{G}}(y_i))\rvert$
Locally-Adaptive Conformal Prediction	$\lvert\overset{*}{x} - \hat{f}(\hat{\mathcal{G}}(y_i))\rvert$	$\frac{\lvert\overset{*}{x} - \hat{f}(\hat{\mathcal{G}}(y_i))\rvert}{\hat{R}(\hat{\mathcal{G}}(y_i))}$

Numerical Results – Uncertainty Quantification

Method	Empirical Coverage Probability
Conformalized Quantile Regression	0.911
Conformal Prediction (residuals)	0.926
Locally-Adaptive Conformal Prediction	0.872



Discussion

Conclusions

- Utilizing an operator network structure exhibits reduced error, even amidst transformation to a QOI compared to direct function-based estimation.
- Utilizing the natural variable-width modelling of quantile regression as the uncertainty notion for conformal prediction shows major improvements over standard conformal prediction methods.

Discussion

- The potential use of this kind of method as initial model estimates (e.g., for surrogates models or bayesian model priors).
- Is there an ability to incorporate known physical models into the learning mechanics (e.g., physics-informed or physics-induced NN)?

References I

- Chen, Colin (2007). "A finite smoothing algorithm for quantile regression". In: *Journal of Computational and Graphical Statistics* 16.1, pp. 136–164.
- Chen, Tianping and Hong Chen (1995). "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems". In: *IEEE transactions on neural networks* 6.4, pp. 911–917.
- Lu, Lu et al. (2021). "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature machine intelligence* 3.3, pp. 218–229.