Uncertainty Quantification for Deep Regression using Contextualized Normalizing Flows

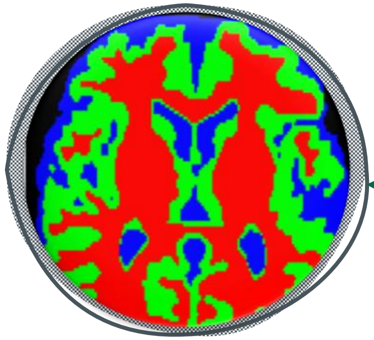
Adriel Sosa Marco‡, John Daniel Kirwan‡, Alexia Toumpa§, Simos Gerasimou§*

‡Arquimea Research Center, Spain
§,Department of Computer Science, University of York, UK
*Department of Elect. Eng. And Computer Science and Eng., Cyprus University of Technology, Cyprus

Problem Statement

Drug Discovery

Medical Image Segmentation



DL-model decision-making

Autonomous systems

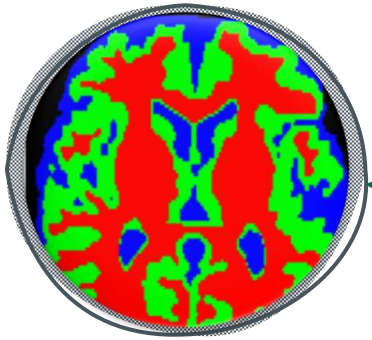
Credit Scoring

Energy forecasting

Problem Statement

Drug Discovery

Medical Image Segmentation



Autonomous systems

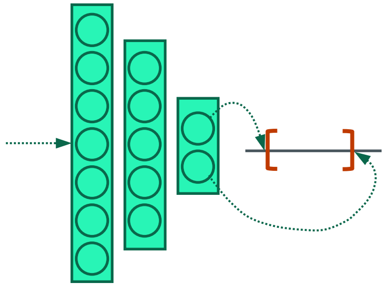
How certain are these models?

Credit Scoring

Energy forecasting

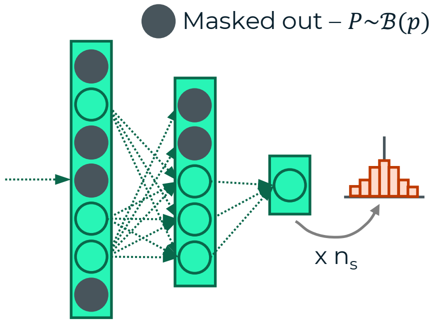
Antecedents

Quantile Regression



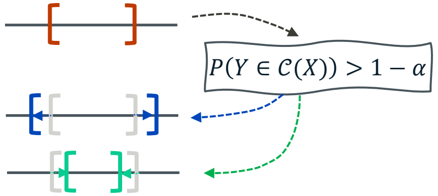
✔ Low network instrumentalization

Monte Carlo Dropout



✔ Principled propagation of epistemic uncertainty

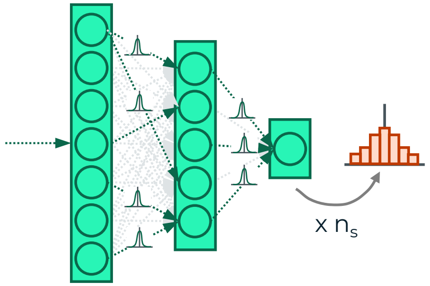
Conformal Pred.



✔ Distribution-free

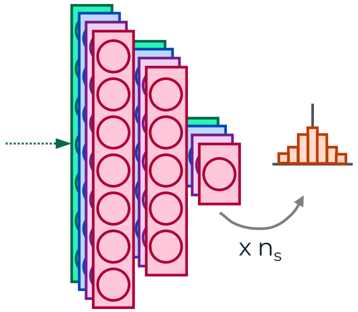
✔ Well-calibrated Post hoc estimate

Full Bayesian Inference



✔ Native accounting for aleatoric and epistemic uncertainties

Deep Ensembles



✔ Uncertainty as a predictive distribution

MCNF: Desiderata

1 Full predictive pdf

Characterize uncertainty using a probabilistic approach

2 Epistemic + Aleatoric

Uses MCD to propagate epistemic uncertainty and further refine it in a second step

3 Post hoc UQ

Leverage internal status of predictive model

4 Distribution-free

Leverage Normalizing Flows to avoid strong distributional assumptions

MCNF: Implementation

1 **Full predictive pdf**

$$p(y|\mathbf{x},\mathcal{D}) = \mathbb{E}_{p(y_{MCD}|\boldsymbol{x},\mathcal{D})}[p(y|y_{MCD},\boldsymbol{x},\mathcal{D})]$$

MCNF: Implementation

1 **Full predictive pdf**

$$p(y|\mathbf{x},\mathcal{D}) = \mathbb{E}_{p(y_{MCD}|\boldsymbol{x},\mathcal{D})}[p(\delta|y_{MCD},\boldsymbol{x},\mathcal{D})], \delta = y - y_{MCD}$$

MCNF: Implementation

1. **Full predictive pdf**

$$p(y|\mathbf{x},\mathcal{D}) = \mathbb{E}_{p(y_{MCD}|\boldsymbol{x},\mathcal{D})}[p(\delta|y_{MCD},\boldsymbol{x},\mathcal{D})], \delta = y - y_{MCD}$$

2. **Epistemic + Aleatoric**

Monte Carlo Dropout: $p(y_{MCD}|\boldsymbol{x},\mathcal{D})$

MCNF: Implementation

1 **Full predictive pdf**

$p(y|\mathbf{x}, \mathcal{D}) = \mathbb{E}_{p(y_{MCD}|\boldsymbol{x}, \mathcal{D})}[p(\delta|y_{MCD}, \boldsymbol{x}, \mathcal{D})], \delta = y - y_{MCD}$

2 **Epistemic + Aleatoric**

Monte Carlo Dropout: $p(y_{MCD}|\boldsymbol{x}, \mathcal{D})$

3 **Post hoc UQ**

$\boldsymbol{c} = \{\bar{y}_{MCD}, \log s^2, \boldsymbol{h}(\boldsymbol{x})\}$

$p(y|\mathbf{x}, \mathcal{D}) = \mathbb{E}_{p(y_{MCD}|\boldsymbol{x}, \mathcal{D})}[p(\delta|y_{MCD}, \boldsymbol{c}, \mathcal{D})]$

MCNF: Implementation

ARQUIMEA

UNIVERSITY of York

1 **Full predictive pdf**

$$p(y|\mathbf{x},\mathcal{D}) = \mathbb{E}_{p(y_{MCD}|\boldsymbol{x},\mathcal{D})}[p(\delta|y_{MCD},\boldsymbol{x},\mathcal{D})], \delta = y - y_{MCD}$$

2 **Epistemic + Aleatoric**

Monte Carlo Dropout: $p(y_{MCD}|\boldsymbol{x},\mathcal{D})$

3 **Post hoc UQ**

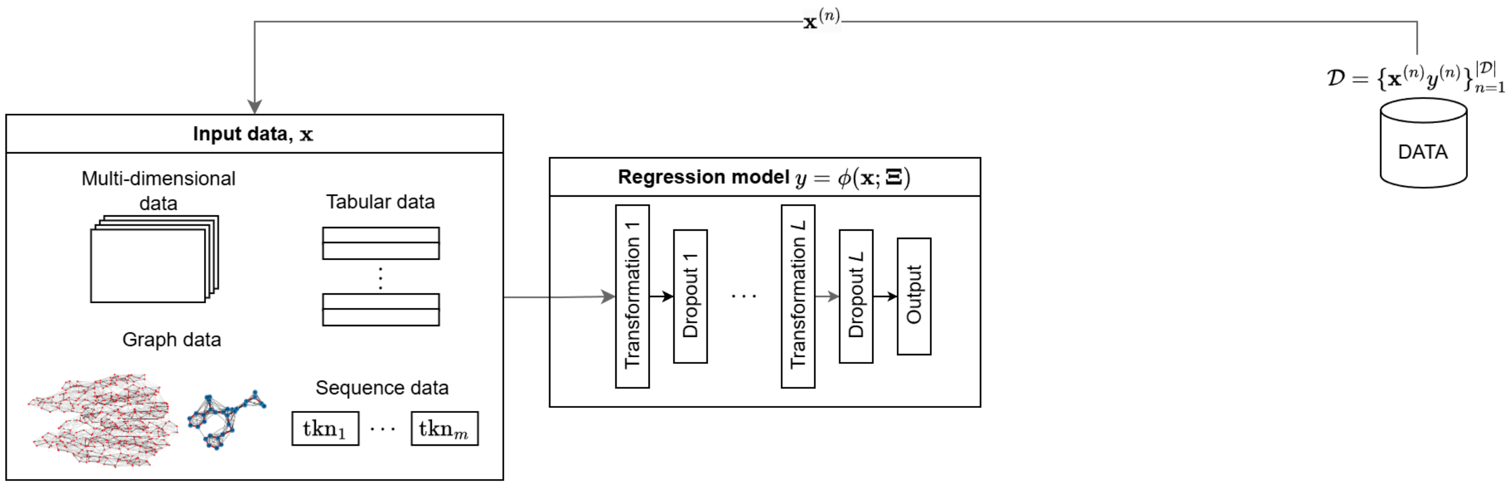
$$\boldsymbol{c} = \{\bar{y}_{MCD}, \log s^2, \boldsymbol{h}(\boldsymbol{x})\}$$

$$p(y|\mathbf{x},\mathcal{D}) = \mathbb{E}_{p(y_{MCD}|\boldsymbol{x},\mathcal{D})}[p(\delta|y_{MCD},\boldsymbol{c},\mathcal{D})]$$

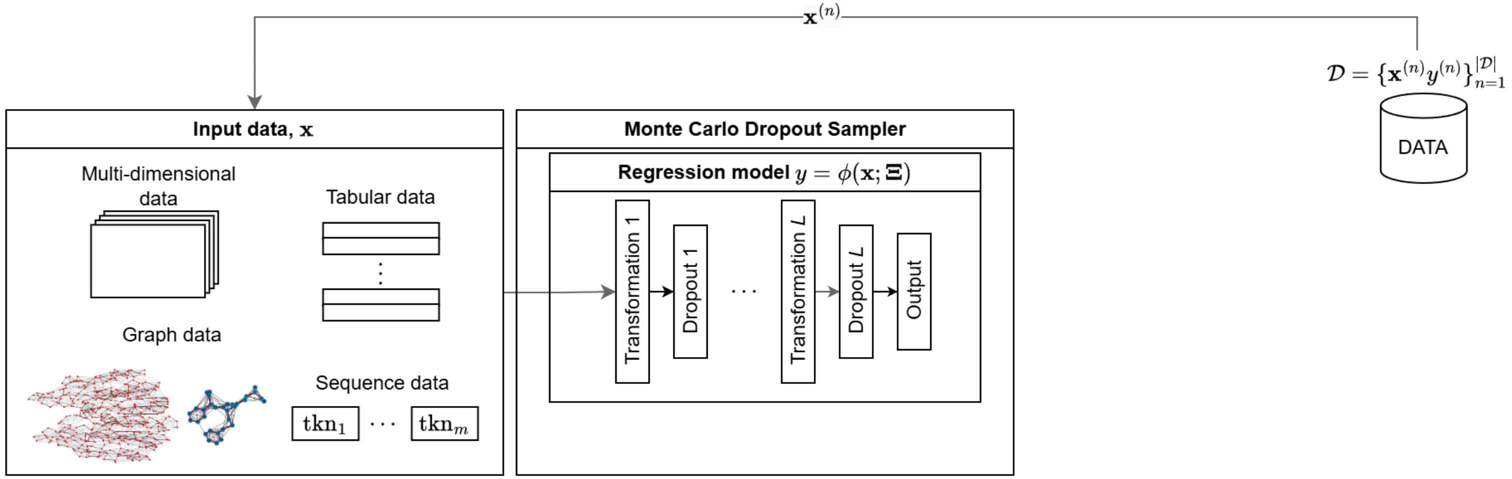
4 **Distribution-free**

$$p(\delta|y_{MCD},\boldsymbol{c},\mathcal{D}) \sim p_{\boldsymbol{\psi}}\left(g^{-1}(\delta,\boldsymbol{c},\boldsymbol{\theta})\right)\prod_{l=1}^{L}\left|\det\left(J_{g_l}\left(g^{-1}(\delta,\boldsymbol{c},\boldsymbol{\theta})\right)\right)\right|$$

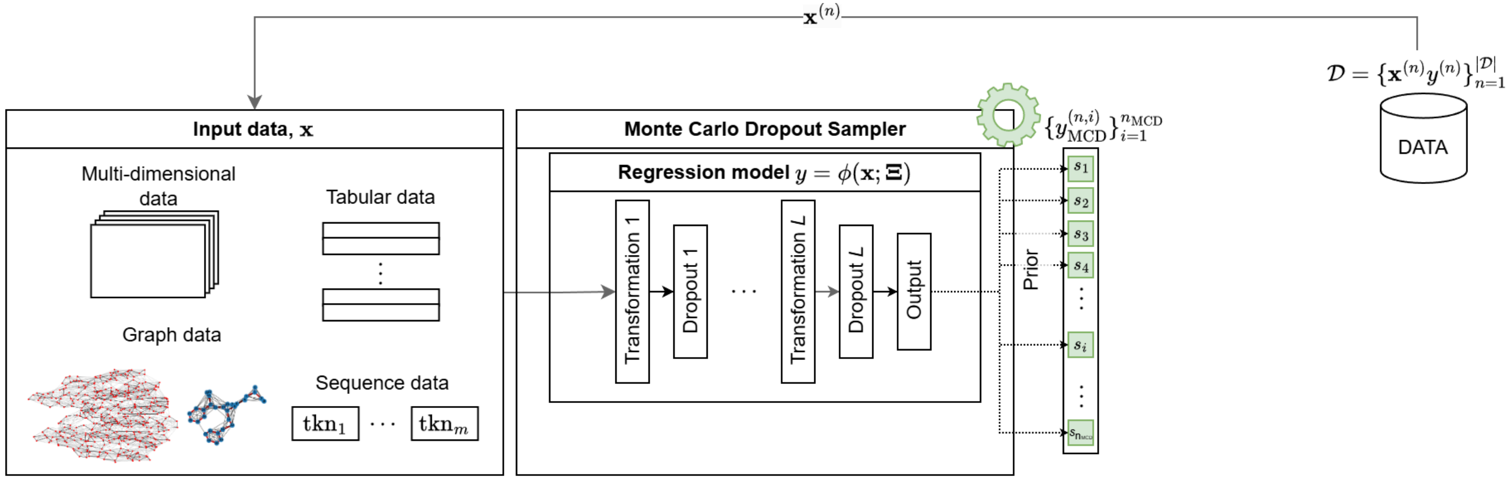
MCNF: How does it operate?



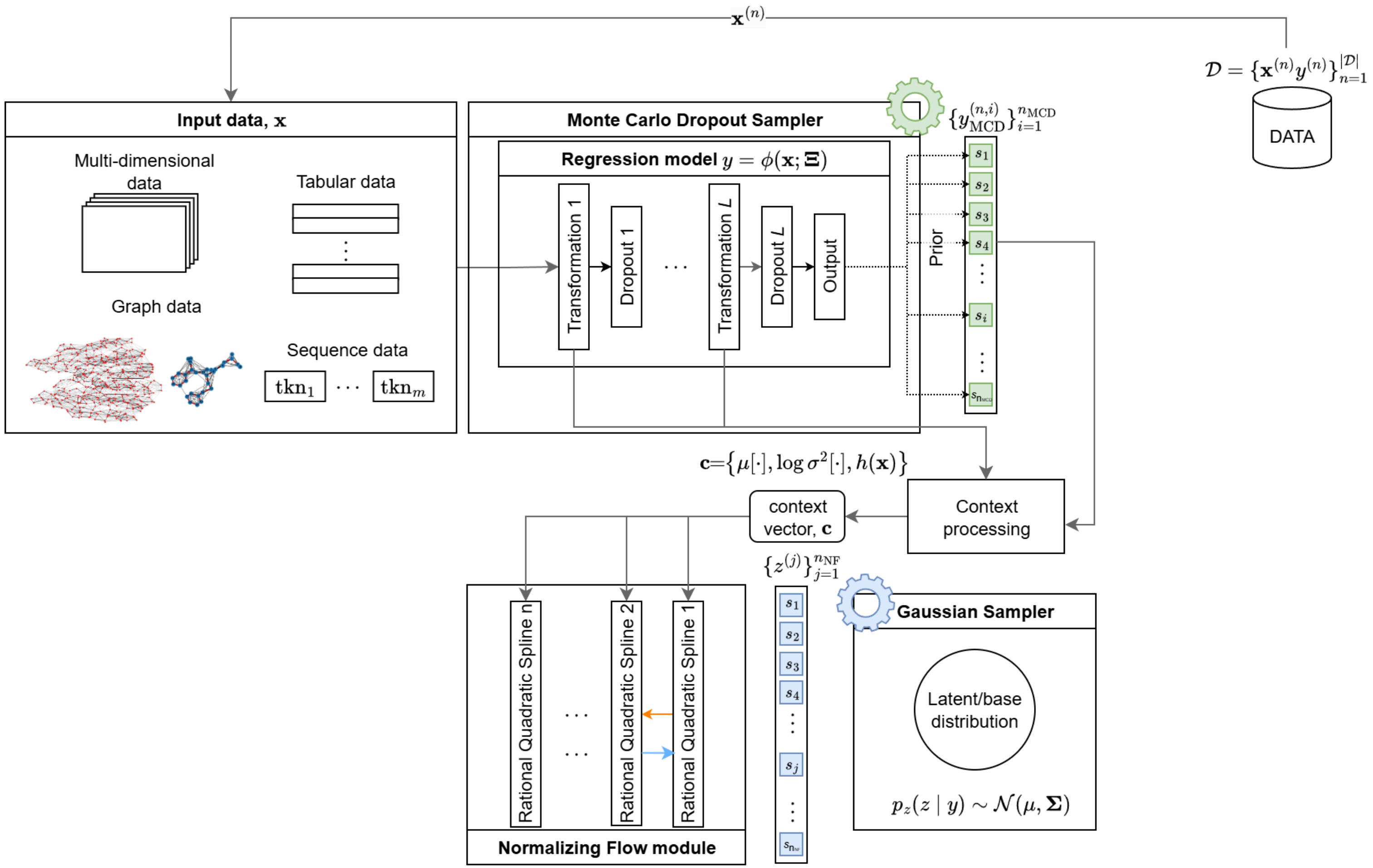
MCNF: How does it operate?



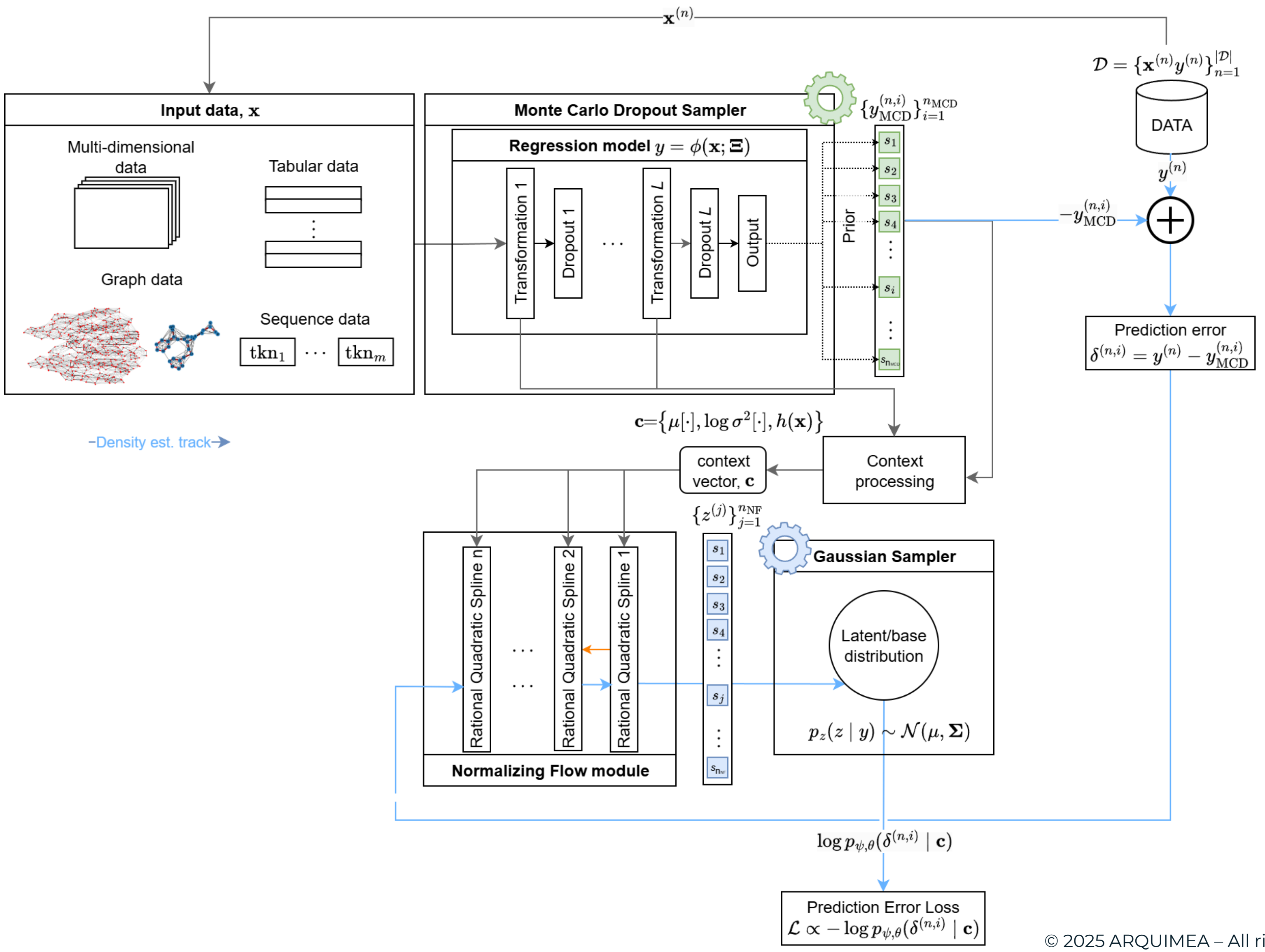
MCNF: How does it operate?



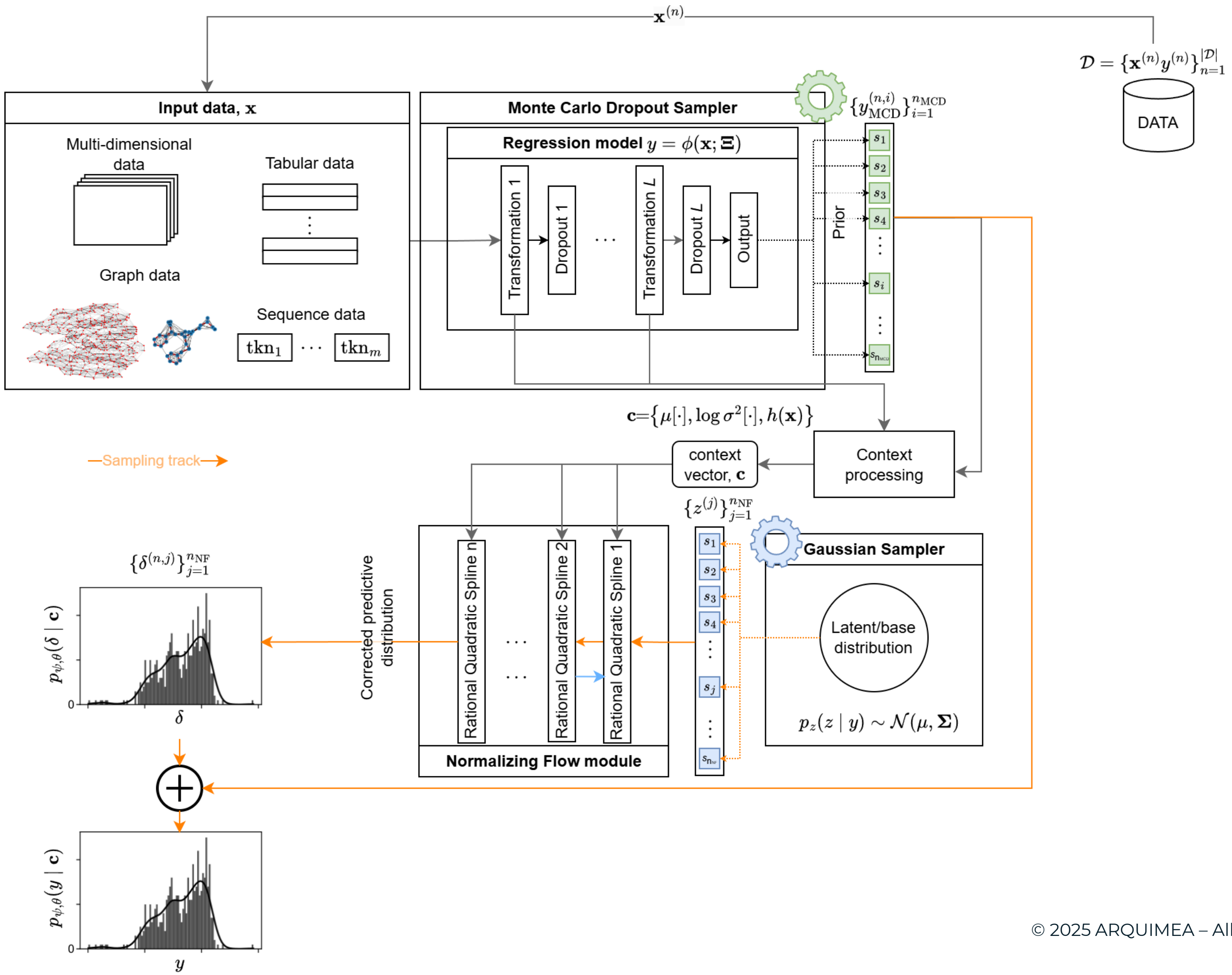
MCNF: How does it operate?



MCNF: How does it operate?



MCNF: How does it operate?



MCNF: Training Loss

Negative log-likelihood

$$\mathcal{L}_{NL}(\boldsymbol{\theta}, \psi) = D_{KL}\left[p_{(y|\boldsymbol{x})}(y|\boldsymbol{x}) \,||\, p_{\boldsymbol{\theta},\psi}(y|y_{MCD}, \boldsymbol{c}, \mathcal{D})\right]$$

$$\approx -\frac{1}{N}\sum_{n=1}^{N} \log p_{\boldsymbol{\psi}}\left(g^{-1}(y_n, \boldsymbol{c}, \boldsymbol{\theta})\right) + \log\left|\det\left(J_{g_l}\left(g^{-1}(y_n, \boldsymbol{c}, \boldsymbol{\theta})\right)\right)\right| + \text{const.}$$

Weighted negative log-likelihood

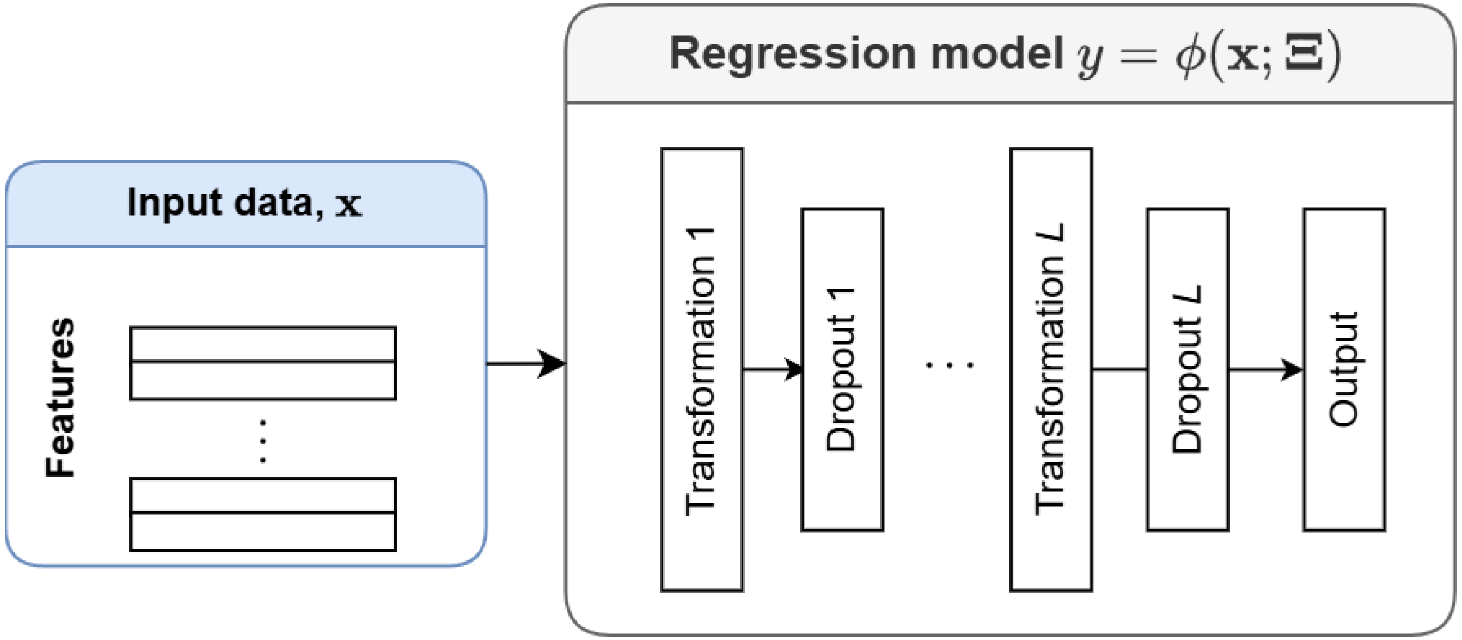
$$\mathcal{L}_{NL}(\boldsymbol{\theta}, \psi) = -\sum_{n=1}^{N} \boldsymbol{w}_n \left(\log p_{\boldsymbol{\psi}}\left(g^{-1}(y_n, \boldsymbol{c}, \boldsymbol{\theta})\right) + \log\left|\det\left(J_{g_l}\left(g^{-1}(y_n, \boldsymbol{c}, \boldsymbol{\theta})\right)\right)\right|\right)$$

$$\boldsymbol{w}_n = \sigma\left(-\frac{\log p_{MCD}(y_n|\boldsymbol{x}_n)}{\tau}; \tau\right) \in (0,1)$$

Experimental setup

UC Irvine
Machine Learning
Repository

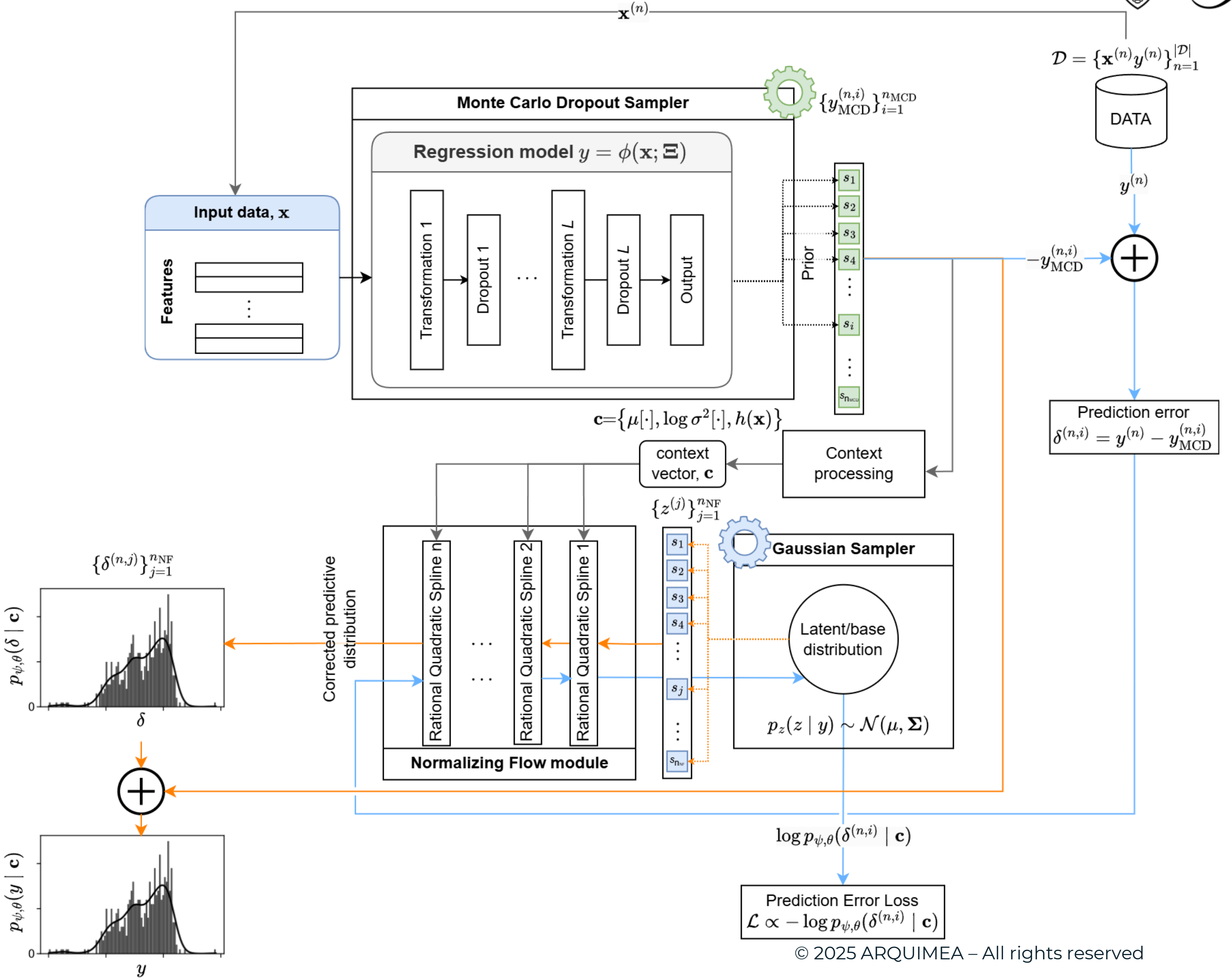
- Boston Housing
- Concrete
- Abalone
- Protein
- Wave Energy
- Superconductor
- Romano-OG
- Romano-Mod



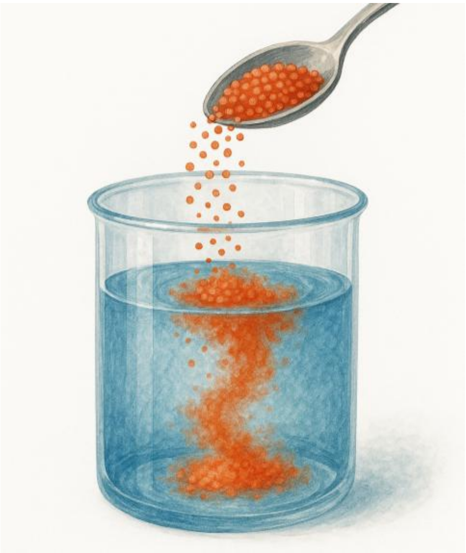
Experimental setup

UC Irvine Machine Learning Repository

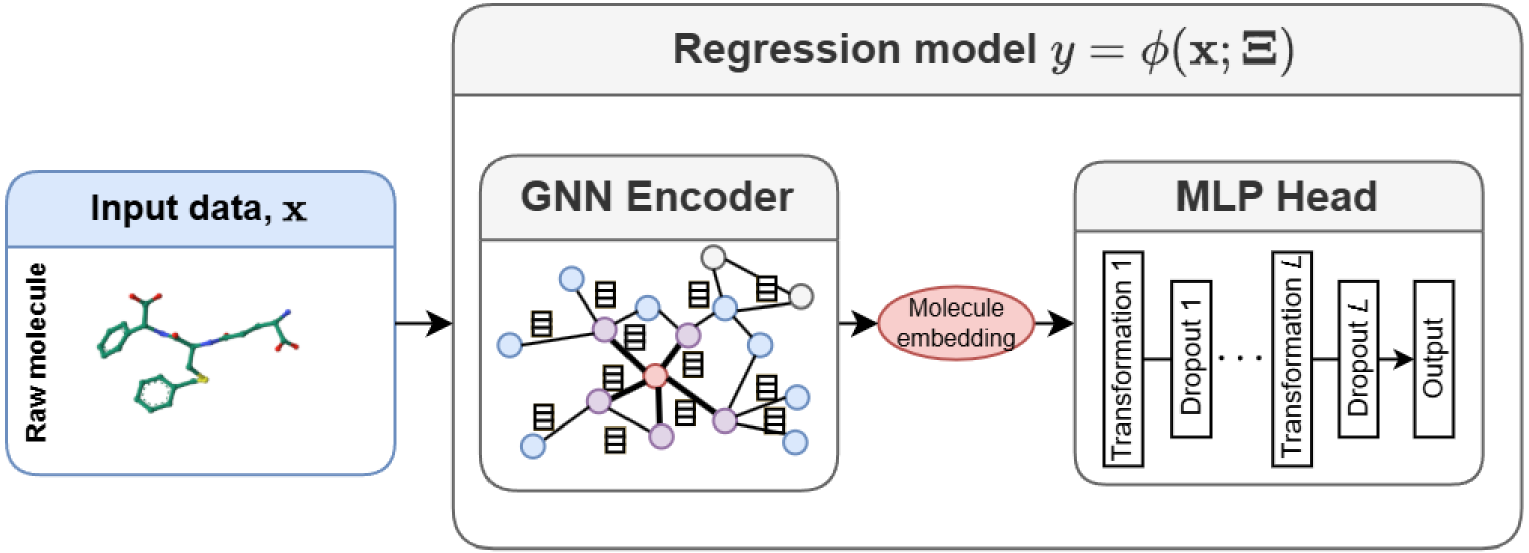
- Boston Housing
- Concrete
- Abalone
- Protein
- Wave Energy
- Superconductor
- Romano-OG
- Romano-Mod



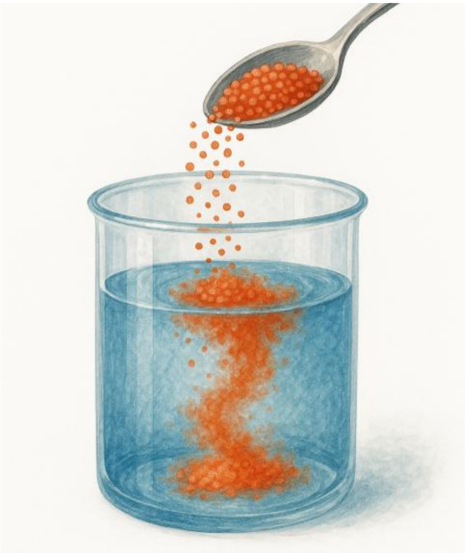
Experimental setup



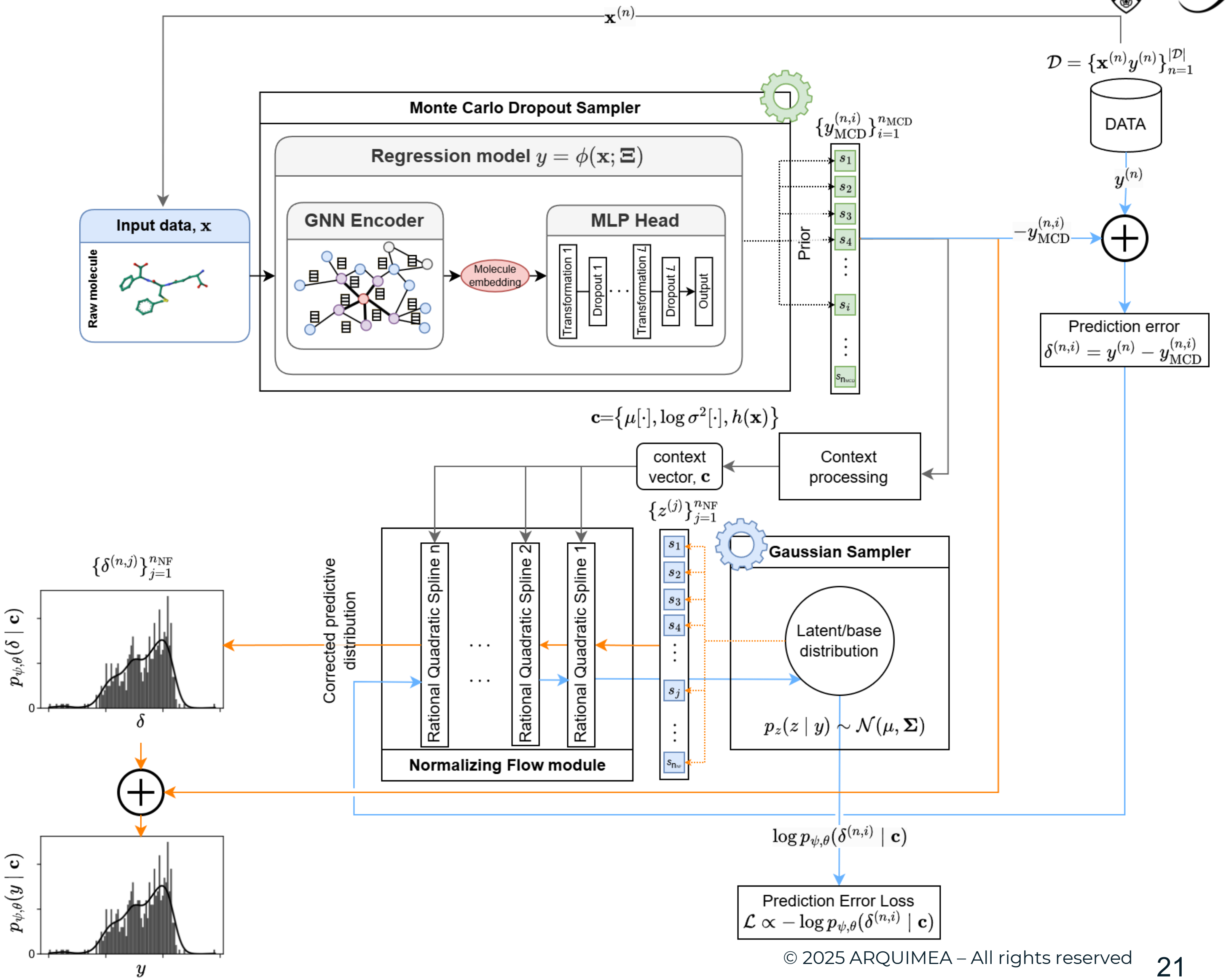
Solubility



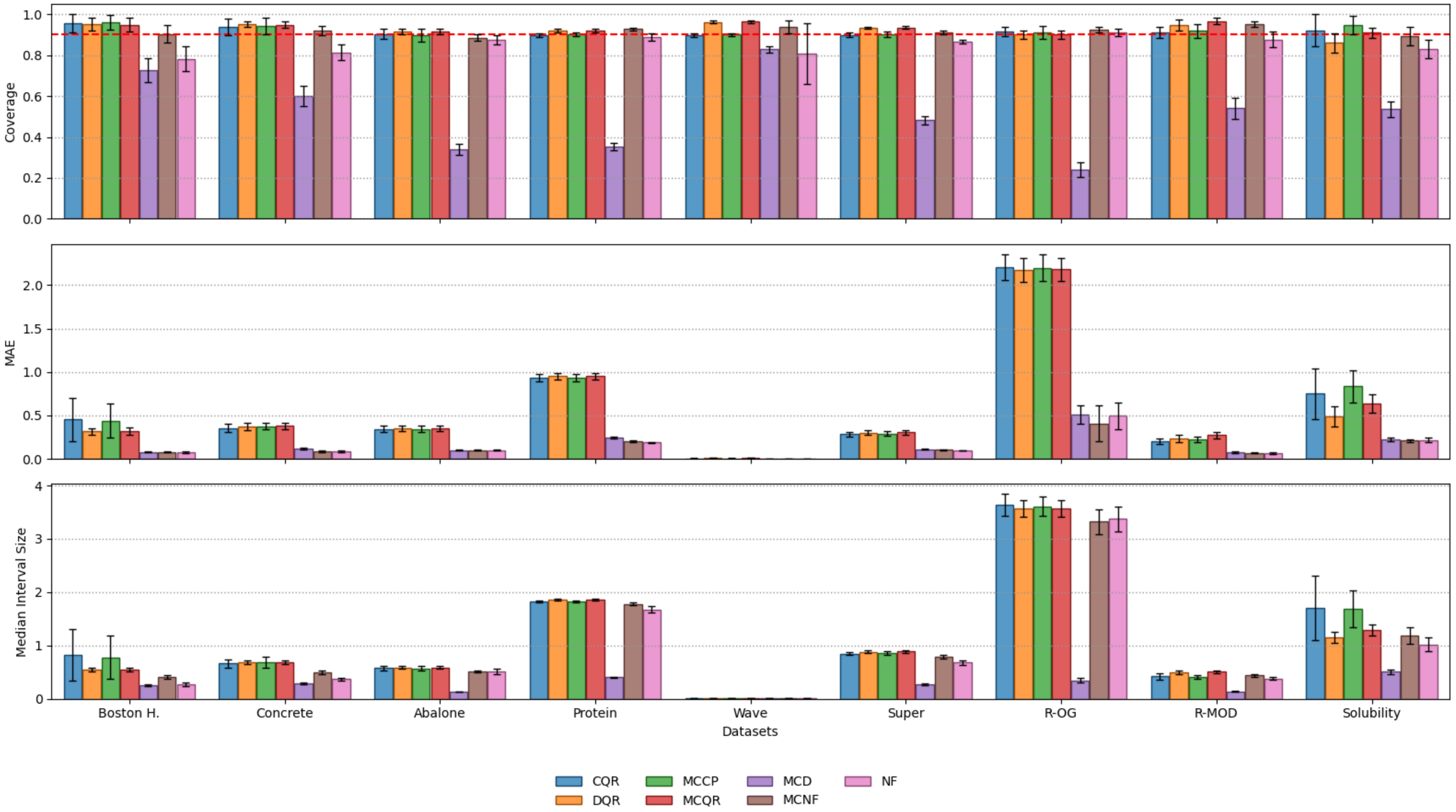
Experimental setup



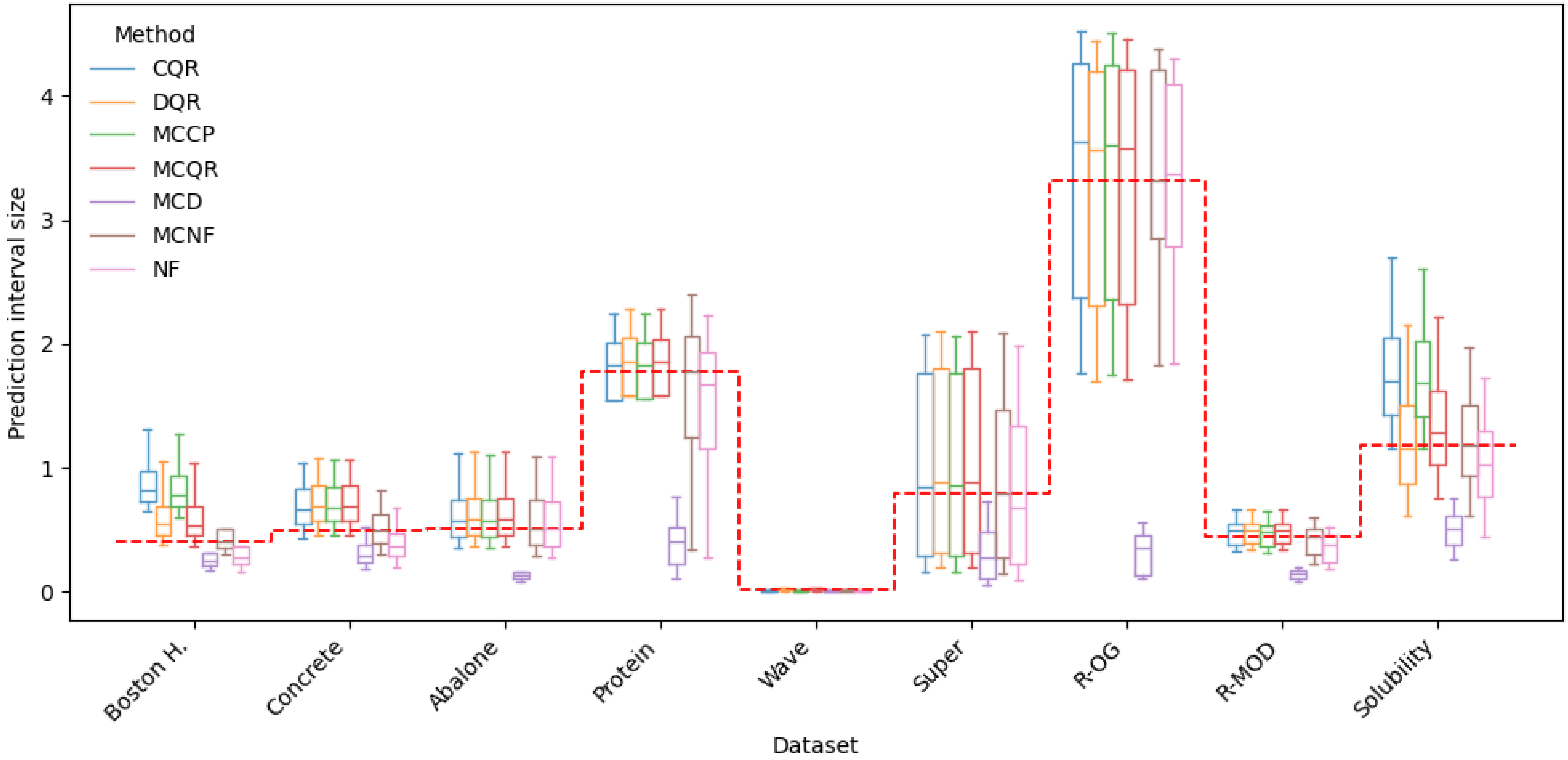
Solubility



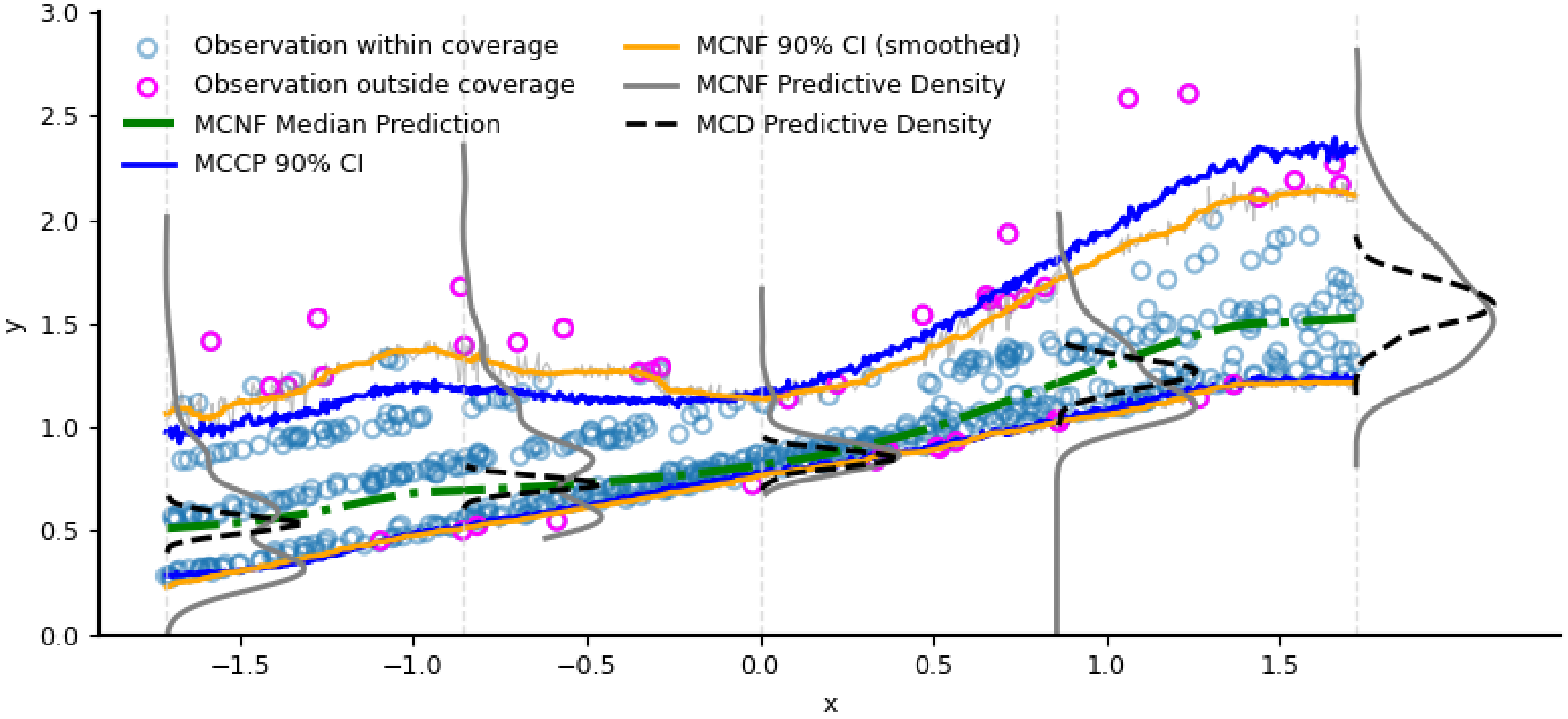
Results: Coverage, MAE, Size



Results: Adaptivity



Results: Multimodality



Conclusions and future work

In summary

- MCNF is a UQ post hoc method for deep regression models
- MCNF produces well-calibrated predictive intervals (coverage, size) while providing richer information than baselines
- We show that the method generalizes well to other DL architectures, such as pre-trained GNNs

Future work

- Extend the MCNF formalism to classification problems
- Improve computational efficiency by replacing sampling-based elements of the method

Thank you!