Probabilistic Pretraining for Improved Neural Regression

Anonymous authors
Paper under double-blind review

Abstract

While transfer learning has revolutionized computer vision and natural language processing, its application to probabilistic regression remains underexplored, particularly for tabular data. We introduce NIAQUE (Neural Interpretable Any-Quantile Estimation), a novel permutation-invariant architecture that enables effective transfer learning across diverse regression tasks. Through extensive experiments on 101 datasets, we demonstrate that pre-training NIAQUE on multiple datasets and fine-tuning on target datasets consistently outperforms both traditional tree-based models and transformer-based neural baseline. On real-world Kaggle competitions, NIAQUE achieves competitive performance against heavily hand-crafted and feature-engineered solutions and outperforms strong baselines such as TabPFN and TabDPT, while maintaining interpretability through its probabilistic framework. Our results establish NIAQUE as a robust and scalable approach for tabular regression, effectively bridging the gap between traditional methods and modern transfer learning.

1 Introduction

Tabular data underpins high-stakes decision-making across healthcare (Rajkomar et al., 2018), real estate valuation De Cock (2011), energy systems Olson et al. (2017a), and e-commerce McAuley et al. (2015). In these settings, models must be accurate, data-efficient and, crucially for operational use, able to quantify predictive uncertainty. Historically, tree-based ensembles such as Random Forests Breiman (2001), XGBoost Chen & Guestrin (2016), LightGBM Ke et al. (2017), and CatBoost Prokhorenkova et al. (2019) have been the de facto standard due to their robustness, ease of use, and strong performance with heterogeneous features and modest sample sizes. Recent work has revisited deep learning for tabular data, motivated by the promise of end-to-end representation learning, joint optimization with downstream objectives, and straightforward multimodal fusion. Architectures such as TabNet Arik & Pfister (2021) and TabTransformer Huang et al. (2021) demonstrate that appropriately constrained inductive biases (e.g., learned feature selection, attention over categorical embeddings) can close the gap to boosted trees on many benchmarks while enabling capabilities such as differentiable feature learning and integration with text or images. Yet, unlike computer vision and natural language processing, where transfer learning via large-scale pretraining across heterogenous real datasets is now foundational, the corresponding paradigm for tabular data remains comparatively under-explored, particularly for *probabilistic regression* tasks that require calibrated predictive distributions rather than point estimates (Hollmann et al., 2023; Levin et al., 2023).

Challanges. This work addresses several fundamental challenges in tabular regression. *Transfer Learning Gap:* While recent work has explored transfer learning for tabular classification, probabilistic regression remains underexplored, lacking frameworks that effectively transfer knowledge across diverse regression tasks. *Scalability–Interpretability Trade-off:* Existing approaches often sacrifice either scalability or interpretability, limiting their practical utility in real-world applications requiring both robust performance and explainable predictions. *Benchmark Limitations:* Current multi-dataset tabular benchmarks predominantly focus on classification tasks, hindering systematic evaluation of regression models.

Our approach: NIAQUE. We introduce *NIAQUE* (Neural Interpretable Any-Quantile Estimation), a probabilistic regression model designed to resolve these challenges. First, NIAQUE enables effective *co-training across multiple disjoint datasets*, exhibiting positive transfer and strong scalability when fine-tuned

on new regression tasks. Second, its probabilistic framework supports *interpretable* analysis via marginal posterior distributions that yield feature-level importance summaries, enhancing transparency and reliability in decision-critical settings. Third, we provide *theoretical guarantees* showing that, under appropriate regularity conditions, NIAQUE approximates the inverse of the posterior distribution, thereby formalizing its ability to recover task-conditional quantiles. To enable comprehensive assessment, we introduce a new multi-dataset regression benchmark comprising *101* diverse datasets spanning multiple domains. We evaluate with proper scoring rules (e.g., CRPS), calibration diagnostics, and standard accuracy metrics. Across this suite, NIAQUE outperforms strong tree-based baselines and neural approaches while maintaining interpretability. Furthermore, on real-world Kaggle regression challenges, NIAQUE achieves competitive performance against highly engineered solutions.

Contributions. Our primary contributions are:

1. A novel deep probabilistic regression model (**NIAQUE**) that enables effective transfer learning across diverse tabular datasets via co-training and task-adaptation.
2. Theoretical analysis establishing NIAQUE's convergence to inverse posterior distribution estimation.
3. Comprehensive empirical validation demonstrating superior performance over strong baselines—both boosted trees and modern neural methods—in transfer learning settings, alongside calibration and robustness analyses.
4. A new benchmark suite for tabular regression (101 datasets), facilitating systematic evaluation of transfer-learning approaches.

Paper outline. Section 2 reviews relevant literature and Section 3 provides background and presents problem formulation. Section 4 describes NIAQUE. Section 5 reports our empirical methodology and results, including dataset and baseline descriptions, cross-training and transfer learning results, interpretability analysis and ablations. We discuss limitations and implications in Section 6 and conclude in Section 7.

2 Related Work

Probabilistic Regression. This work builds on probabilistic time-series modeling approach Smyl et al. (2024), refining its theoretical underpinnings and extending architectural design for tabular transfer learning applications. Alternative methods, such as Neural Processes Garnelo et al. (2018b) and Conditional Neural Processes Garnelo et al. (2018a), offer conditional probabilistic solutions to regression but are constrained to fixed-dimensional input spaces, limiting their applicability to cross-dataset, multi-task regression. Our approach effectively transfers knowledge across datasets with varying feature spaces and target domains, establishing a flexible and scalable framework for conditional probabilistic regression.

Transfer Learning in Tabular Data. Transfer learning has driven major advances in computer vision Sun et al. (2021); Radford et al. (2021) and language modeling Devlin et al. (2019) by leveraging shared representations across tasks. Recent successes in time-series forecasting Garza & Mergenthaler-Canseco (2023); Ansari et al. (2024) demonstrate transfer learning's potential for numerical prediction tasks. However, probabilistic transfer learning for tabular data remains largely unexplored, with existing work primarily focusing on classification tasks or point estimates. Our work bridges this gap by introducing a framework specifically designed for probabilistic transfer across diverse tabular datasets.

Deep Learning vs. Tree-based Models. The comparison between deep learning and tree-based approaches for tabular data has been extensively studied, particularly for classification tasks. Previous evaluations include: transformer architectures across 20 classification datasets, MLPs versus TabNet and tree-based models on 40 classification datasets, Comprehensive comparison of various architectures across 45 datasets Grinsztajn et al. (2022). Most relevant to our work are TabPFN Hollmann et al. (2023) and TabDPT Ma et al. (2024), which introduce Transformer-based approaches for synthetic pre-training. While these methods share similar goals, our work differs in three key aspects. First, we propose a novel architectural approach based on deep prototype aggregation, which scales linearly in the number of input features. Second, we focus

specifically on probabilistic pretraining on real public datasets and transfer learning. Finally, we demonstrate superior empirical accuracy on real-world Kaggle competitions through direct probabilistic pretraining and fine-tuning on downstream regression tasks.

Permutation-invariant Architectures. Our architectural design builds upon advances in permutation-invariant representations, crucial for handling variable feature spaces in multi-task learning. We extend the prototype-based architecture proposed by Oreshkin et al. (2022) for pose completion to support any-quantile modeling in tabular regression. This approach relates to several key developments in the field. PointNet Qi et al. (2017) and DeepSets Zaheer et al. (2017) pioneered pooling techniques for variable-dimensional inputs. ResPointNet Niemeyer et al. (2019) generalized these approaches with residual connections. Prototypical Networks Snell et al. (2017) demonstrated effectiveness of prototype-based learning by leveraging average-pooled embeddings for few-shot classification. Transformer architecture Vaswani et al. (2017) established flexible processing of variable-length sequences.

3 Background and Problem Formulation

Let $\mathbb{R}$ denote the set of real numbers and $\mathcal{U}(0,1)$ the uniform distribution over the interval $(0,1)$. For a vector $\mathbf{x}$, we denote its dimensionality as $|\mathbf{x}|$. For a random variable Y with cumulative distribution function (CDF) $F(y) = P(Y \leq y)$, the q-th quantile $q \in (0,1)$ is defined as:

$$F^{-1}(q) = \inf\{y \in \mathbb{R} : F(y) \geq q\}.$$

3.1 Problem Formulation

Let $\mathcal{X}$ be the input feature space and $\mathcal{Y} \subseteq \mathbb{R}$ be the space of the target variable. We consider a probability distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. For any instance $\mathbf{x} \in \mathcal{X}$, the relationship between features and target variable is given by:

$$y = \Psi(\mathbf{x}, \varepsilon) \tag{1}$$

where $\Psi : \mathcal{X} \times \mathcal{E} \to \mathcal{Y}$ is an unknown non-linear function and $\varepsilon \in \mathcal{E}$ represents stochastic noise with unknown distribution.

Given a finite training sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ drawn i.i.d. from $\mathcal{D}$, we aim to learn a probabilistic regression function $f_\theta : \mathbb{R}^{|\mathbf{x}| \times Q} \to \mathbb{R}^Q$, parameterized by $\theta \in \Theta$, which maps an input $\mathbf{x}$ to a Q-tuple of quantiles $(q_1, ..., q_Q)$, where $q_i \in (0,1)$ for $i \in [Q]$, thereby capturing the conditional distribution of $y|\mathbf{x}$.

3.2 Performance Metrics

Let y_i denote the ground truth sample and $\hat{y}_{i,q}$ its q-th quantile prediction for a dataset with S samples. To evaluate the quality of distributional predictions, we use Continuous Ranked Probability Score (CRPS). The theoretical definition of CRPS for a predicted cumulative distribution function F and observation y is:

$$\mathrm{CRPS}(F, y) = \int_{\mathbb{R}} \left(F(z) - \mathbb{1}_{\{z \geq y\}}\right)^2 \mathrm{d}z, \tag{2}$$

where $F : \mathbb{R} \to [0,1]$ is the predicted CDF derived from the quantile predictions, and $\mathbb{1}_{\{z \geq y\}}$ is the indicator function. For practical computation with finite samples S and a discrete set of Q quantiles, we approximate this using:

$$\mathrm{CRPS} = \frac{2}{SQ} \sum_{i=1}^{S} \sum_{j=1}^{Q} \rho(y_i, \hat{y}_{i,q_j}) \tag{3}$$

where $\rho(y, \hat{y}_q)$ is the quantile loss function defined as:

$$\rho(y, \hat{y}_q) = (y - \hat{y}_q)(q - \mathbb{1}_{\{y \leq \hat{y}_q\}}) \tag{4}$$

Coverage measures empirical calibration of predictive confidence intervals. For confidence level $\alpha \in (0, 1)$:

$$\text{COVERAGE}(\alpha) = \frac{1}{S}\sum_{i=1}^{S} \mathbb{1}[y_i > \hat{y}_{i,0.5-\alpha/2}]\mathbb{1}[y_i < \hat{y}_{i,0.5+\alpha/2}] \cdot 100\%$$

We employ the following metrics to evaluate the accuracy of point predictions: Symmetric Mean Absolute Percentage Error (sMAPE), Average Absolute Deviation (AAD), Root Mean Square Error (RMSE), Root Mean Square Logarithmic Error (RMSLE). Detailed mathematical definitions of the metrics are provided in Appendix A.

4 NIAQUE

In this section, we present NIAQUE (Neural Interpretable Any-Quantile Estimation), a probabilistic regression model. We first introduce the any-quantile learning approach as a general solution to the probabilistic regression problem defined in Section 3. We prove that this approach converges to the inverse cumulative distribution function of the conditional distribution, providing a theoretical foundation for our method. We then detail NIAQUE's neural architecture, demonstrate how it enables transfer learning across diverse tabular datasets, and present an approach to model interpretability based on probabilistic considerations.

4.1 Any-Quantile Learning

We formulate the any-quantile learning approach by augmenting the input space to include a quantile level $q \in (0, 1)$, allowing the neural network f_θ to learn mappings from $(\mathbf{x}, q)$ to the corresponding q-th conditional quantile of the target variable $y|\mathbf{x}$. Let $\widehat{y}_q = f_\theta(\mathbf{x}, q)$ represent the predicted q-th quantile of the conditional distribution of $y|\mathbf{x}$. The objective is to learn parameters θ that minimize the expected quantile loss:

$$\min_\theta \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},q\sim\mathcal{U}(0,1)}[\rho(y, f_\theta(\mathbf{x}, q))]\,, \tag{5}$$

where $\rho(\cdot, \cdot)$ is the quantile loss function defined in Equation equation 4.

We use gradient descent and mini-batch to learn the parameters. Precisely, the neural network is trained on dataset of S samples, $(\mathbf{x}_i, y_i)$ drawn from the joint probability distribution $\mathcal{D}$. During training the quantile value q is sampled from $\mathcal{U}(0, 1)$ and the loss is minimized using stochastic gradient descent (SGD). For a mini-batch of size B, the parameter update at iteration k is:

$$\theta_{k+1} = \theta_k - \eta_k \nabla_\theta \frac{1}{B}\sum_{i=1}^{B} \rho(y_i, f_\theta(\mathbf{x}_i, q_i))\,. \tag{6}$$

As $k \to \infty$, the parameters converge to the solution of the following empirical risk minimization problem (Karimi et al., 2016):

$$\theta^* = \arg\min_{\theta\in\Theta} \frac{1}{S}\sum_{i=1}^{S} \rho(y_i, f_\theta(\mathbf{x}_i, q_i))\,. \tag{7}$$

By the strong law of large numbers, as S grows, the empirical risk converges to the expected quantile loss:

$$\mathbb{E}_{\mathbf{x},y}\mathbb{E}_q \rho(y, f_\theta(\mathbf{x}, q)) = \mathbb{E}_{\mathbf{x},y}\int_0^1 \rho(y, f_\theta(\mathbf{x}, q))dq\,. \tag{8}$$

This expected loss has a direct connection to the Continuous Ranked Probability Score (CRPS), which can be expressed as an integral over quantile loss (Gneiting & Ranjan, 2011):

$$\text{CRPS}(F, y) = 2\int_0^1 \rho(y, F^{-1}(q))dq\,. \tag{9}$$

Based on this fact, the following theorem proves that the expected pinball loss equation 8 is minimized when $f_\theta(\mathbf{x}, q)$ corresponds to the inverse of the posterior CDF $P_{y|\mathbf{x}}$.

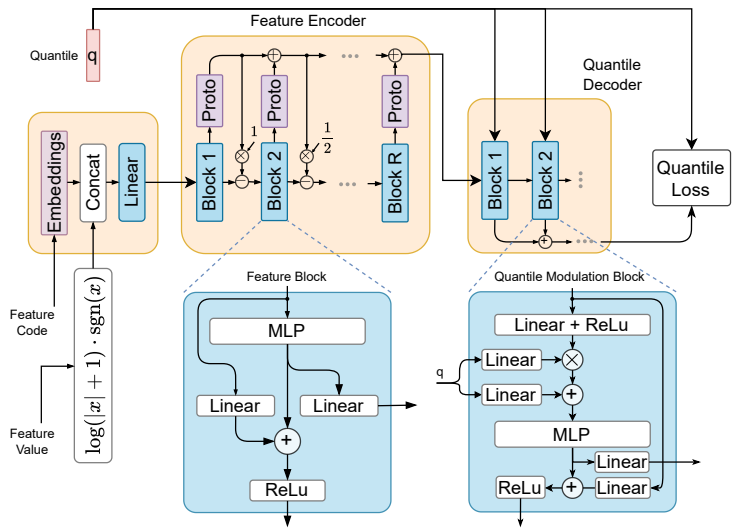


Figure 1: NIAQUE's encoder-decoder architecture transforms variable-dimensional inputs into fixed-size representations, enabling transfer learning and multi-task knowledge sharing across datasets.

Theorem 1. *Let F be a probability measure over variable y such that inverse F^{-1} exists and let $P_{y,\mathbf{x}}$ be the joint probability measure of variables $\mathbf{x}, y$. Then the expected loss, $\mathbb{E}_{\mathbf{x},y,q}\, \rho(y, F^{-1}(q))$, is minimized if and only if $F = P_{y|\mathbf{x}}$.*

The following conclusions emerge. First, the quantile loss SGD update equation 6 optimizes the empirical risk equation 7 corresponding to the expected loss equation 8. Based on (8,9) and Theorem 1, $f_{\theta^\star} = \arg\min_{f_\theta} \mathbb{E}_{\mathbf{x},y,q}\, \rho(y, f_\theta(\mathbf{x}, q))$ has a clear interpretation as the inverse CDF corresponding to $P_{y|\mathbf{x}}$. Second, as both the SGD iteration index k and training sample size S increase, and if f_θ is implemented as an MLP whose width and depth scale appropriately with sample size S, then (Farrell et al., 2021, Theorem 1) implies that the SGD solution converges to $f_{\theta^\star}(\mathbf{x}, q) \equiv P^{-1}_{y|\mathbf{x}}(q)$. Therefore, given uniform $q \sim \mathcal{U}(0, 1)$, $\widehat{y}_q = f_{\theta^\star}(\mathbf{x}, q)$ has the interpretation of a sample from the posterior distribution $p(y|\mathbf{x})$, which follows from the proof of the inversion method (Devroye, 1986, Theorem 2.1).

4.2 Neural Encoder-Decoder Architecture

NIAQUE adopts a modular encoder-decoder design (Fig. 1) to process observation samples $\mathbf{x}_i$ with variable dimensionality d_i. The encoder maps each observation into a fixed-size latent embedding of dimension E, enabling downstream processing independent of input dimensionality. Feature values and associated codes (IDs) (dimension $1 \times d_i$) are embedded into a tensor of size $1 \times d_i \times E_{in}$, where E_{in} is the embedding size per feature. These embeddings are aggregated using a prototype-based method to generate the latent observation representation. The decoder conditions this representation on arbitrary-length quantile vectors $\mathbf{q} \in \mathbb{R}^Q$, modulating the output using FiLM-based transformations Perez et al. (2018). This separates input processing and quantile conditioning, achieving computational efficiency of $O(d_i + Q)$ per sample $\mathbf{x}_i$, compared to $O(d_i Q)$ complexity required to process quantiles and observations jointly. Note that our processing is linear if feature dimensionality, as opposed to quadratic scaling of attention-based approaches.

Inputs: NIAQUE incorporates both raw value and learnable embedding of its ID (encoded as an integer) for each feature in the observation vector $\mathbf{x}$. The feature ID embedding learns feature-specific statistical properties, inter-feature dependencies, and their relationship with the target. The embedded feature ID is concatenated with its value after the log-transform:

$$z = \log(|x| + 1) \cdot \mathrm{sgn}(x)\,, \tag{10}$$

which normalizes the features' dynamic range, aligning it with that of ID embeddings while preserving sign information, facilitating stable training, as validated by the ablation study in Appendix J.

Feature Encoder: The encoder employs a two-loop residual network architecture to efficiently handle variable-dimensional inputs. The following equations define the encoder's transformations, with the sample index i omitted for brevity. Let the encoder input be $\mathbf{x}_{in} \in \mathbb{R}^{d \times E_{in}}$, where d is the number of features and E_{in} is the embedding size per feature. A fully-connected layer $\mathrm{FC}_{r,\ell}$ in residual block $r \in \{1, \ldots, R\}$, layer $\ell \in \{1, \ldots, L\}$, with weights $\mathbf{W}_{r,\ell}$ and biases $\mathbf{a}_{r,\ell}$, is defined as:

$$\mathrm{FC}_{r,\ell}(\mathbf{h}_{r,\ell-1}) \equiv \textsc{ReLU}(\mathbf{W}_{r,\ell}\mathbf{h}_{r,\ell-1} + \mathbf{a}_{r,\ell})\,.$$

We also define a prototype layer as: $\textsc{Prototype}(\mathbf{x}) \equiv \frac{1}{d}\sum_{i=1}^{d} \mathbf{x}[i,:]$. The observation encoder is then described by the following equations:

$$\mathbf{x}_r = \textsc{ReLU}(\mathbf{b}_{r-1} - 1/(r-1) \cdot \mathbf{p}_{r-1}), \tag{11}$$

$$\mathbf{h}_{r,1} = \mathrm{FC}_{r,1}(\mathbf{x}_r),\ \ldots,\ \mathbf{h}_{r,L} = \mathrm{FC}_{r,L}(\mathbf{h}_{r,L-1}), \tag{12}$$

$$\mathbf{b}_r = \textsc{ReLU}(\mathbf{L}_r\mathbf{x}_r + \mathbf{h}_{r,L}),\ \mathbf{f}_r = \mathbf{F}_r\mathbf{h}_{r,L}, \tag{13}$$

$$\mathbf{p}_r = \mathbf{p}_{r-1} + \textsc{Prototype}(\mathbf{f}_r). \tag{14}$$

These equations implement a dual-residual mechanism:

1. Equations equation 12 and equation 13 form an MLP with a residual connection (see Feature Block in Fig. 1 bottom left).

2. Equations equation 11 and equation 14 form a second residual loop with the following key properties: a) Eq. equation 14 consolidates individual feature encodings into a prototype-based representation of the observation. b) Eq. equation 11 introduces an inductive bias by enforcing a delta-mode constraint, ensuring that feature contributions are only relevant when they deviate from the existing observation embedding, $\mathbf{p}_{r-1}$. c) The observation representation accumulates across residual blocks Eq. equation 14, effectively implementing skip connections.

Quantile Decoder: The decoder implements a fully-connected conditioned residual architecture (Fig. 1, top-right) with FiLM-modulated MLP blocks (Fig. 1, bottom-right). Taking the observation embedding $\widetilde{\mathbf{b}}_0 = \mathbf{p}_R \in \mathbb{R}^E$ as input, it generates quantile-modulated representations $\widetilde{\mathbf{f}}_R \in \mathbb{R}^{Q \times E}$ for quantiles $\mathbf{q} \in \mathbb{R}^Q$ through:

$$\begin{aligned}
\mathbf{h}_{r,1} &= \mathrm{FC}^{\mathrm{QD}}_{r,1}(\widetilde{\mathbf{b}}_{r-1})\,, \quad \gamma_r, \beta_r = \textsc{Linear}_r(\mathbf{q})\,,\\
\mathbf{h}_{r,2} &= \mathrm{FC}^{\mathrm{QD}}_{r,1}((1+\gamma_r)\cdot \mathbf{h}_{r,1} + \beta_r)\,,\\
&\ldots\\
\mathbf{h}_{r,L} &= \mathrm{FC}^{\mathrm{QD}}_{r,L}(\mathbf{h}_{r,L-1})\,,\\
\widetilde{\mathbf{b}}_r &= \textsc{ReLU}(\mathbf{L}^{\mathrm{QD}}_r\widetilde{\mathbf{b}}_{r-1} + \mathbf{h}_{r,L}),\ \ \widetilde{\mathbf{f}}_r = \widetilde{\mathbf{f}}_{r-1} + \mathbf{F}^{\mathrm{QD}}_r\mathbf{h}_{r,L}\,.
\end{aligned} \tag{15}$$

The final prediction $\widehat{\mathbf{y}}_q \in \mathbb{R}^Q$ is obtained via linear projection: $\widehat{\mathbf{y}}_q = \textsc{Linear}[\widetilde{\mathbf{f}}_r]$.

4.3 Interpretability

NIAQUE's probabilistic framework facilitates interpretability via quantile predictions conditioned on individual features. Given $f_\theta(\mathbf{x}_s, q)$ as NIAQUE's estimate of quantile q using only feature $\mathbf{x}_s$, the posterior confidence interval for this feature is defined as:

$$\mathrm{CI}_{\alpha,s} = f_\theta(\mathbf{x}_s, 1-\alpha/2) - f_\theta(\mathbf{x}_s, \alpha/2)\,, \tag{16}$$

where $1-\alpha$ represents the probability that the target true value lies within the interval. Intuitively, more informative features produce narrower confidence intervals. We leverage this to quantify feature importance

through normalized weights:

$$W_s = \frac{\overline{W}_s}{\sum_s \overline{W}_s}\,, \quad \overline{W}_s = \frac{1}{\overline{\mathrm{CI}}_{0.95,s}}\,, \quad \overline{\mathrm{CI}}_{\alpha,s} = \frac{1}{S}\sum_i f_\theta(\mathbf{x}_{s,i}, 1-\alpha/2) - f_\theta(\mathbf{x}_{s,i}, \alpha/2)\,. \tag{17}$$

where $\overline{\mathrm{CI}}_{\alpha,s}$ is the average confidence interval width over validation samples $\{\mathbf{x}_i : \mathbf{x}_i \in \mathcal{D}_{\text{val}}\}$. To enhance marginal distribution modeling and support interpretability, we introduce single-feature samples during training, comprising approximately 5% of the dataset. An ablation study in Appendix J confirms the necessity of this augmentation for robust feature importance estimation.

4.4 Transfer Learning

NIAQUE facilitates effective transfer learning through two core mechanisms ensuring effective knowledge transfer across diverse tabular regression tasks with varying feature spaces. First, its feature ID embeddings learn both dataset-specific and cross-dataset relationships, as evidenced by the structured representation space observed in Fig. 3. Second, its prototype-based aggregation enables flexible processing of arbitrary feature combinations, inherently supporting cross-dataset learning.

The transfer learning process consists of two phases. During pretraining, NIAQUE learns from multiple heterogeneous datasets simultaneously, sampling rows uniformly at random from all datasets, with each dataset contributing only its relevant features. The model processes these diverse inputs through shared parameters, learning both task-specific characteristics and generalizable patterns. The learned embeddings capture statistical properties at multiple levels: individual feature distributions, feature interactions within datasets, and common patterns across different regression problems. This pretrained knowledge can then be leveraged in two ways. First, transfer across seen datasets, in which co-training on multiple datasets enables the learning of larger-capacity networks using data diversity as a regularizer, while also leveraging commonalities across classes of problems—implicitly extracted during training—to strengthen performance on related seen tasks. Second, few-shot transfer to unseen datasets through fine-tuning, where the pretrained feature representations provide a strong foundation for learning new tasks with minimal data via strong regression prior stored in pretrained model weights. In both cases, the model learns to distinguish and process dataset-specific features through their semantic embeddings, whereas shared model parameters enable knowledge transfer across related datasets via multi-task training.

The effectiveness of this approach stems from NIAQUE's ability to maintain dataset-specific information while learning transferable representations. Learnable feature ID embeddings act as task identifiers, allowing the model to adapt its processing based on the combination of input features, effectively serving as an implicit task ID. This capability is particularly valuable in tabular domains, where feature relationships and their predictive power can vary significantly across tasks. Our experimental results validate both the efficacy of cross-dataset pretraining and the model's adaptability to novel regression tasks via fine-tuning, establishing NIAQUE as a robust framework for transfer learning in tabular domains.

5 Empirical Results

We conduct extensive experiments to evaluate NIAQUE's effectiveness for transfer learning in tabular regression. Our evaluation addresses three key aspects: 1) the model's transfer learning capabilities across both seen and unseen tasks, 2) the quality of learned representations and interpretability, and 3) its practical effectiveness in a real-world competition setting.

5.1 Datasets and Experimental Setup

Dataset and Evaluation: We introduce TabRegSet-101 (Tabular Regression Set 101), a curated collection of 101 publicly available regression datasets gathered from UCI Kelly et al. (2017), Kaggle Kaggle (2024), PMLB Romano et al. (2021); Olson et al. (2017b), OpenML Vanschoren et al. (2013), and KEEL Alcalá-Fdez et al. (2011). These datasets span diverse domains, including *Housing and Real Estate*, *Energy and Efficiency*, *Retail and Sales*, *Computer Systems*, *Physics Models* and *Medicine* and exhibit different characteristics in

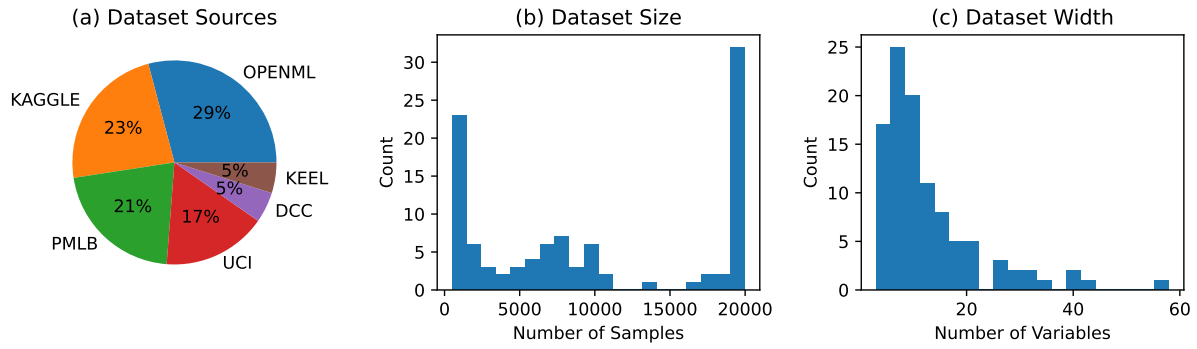


Figure 2: Statistics of the evaluation dataset: (a) distribution by source, (b) dataset sizes, and (c) feature counts.

terms of sample size and feature dimensionality. We focus specifically on the regression task in which the target variable is continuous or, if it has limited number of levels, these are ordered such as student exam scores or wine quality. The target variable in each dataset is normalized to the [0, 10] range and the independent variables are used as is, raw. The target variable scaling is applied to equalize the contributions of the evaluation metrics from each dataset. Datasets have variable number of samples, the lowest being just below 1000. For very large datasets we limit the number of samples used in our benchmark to be 20,000 by subsampling uniformly at random. This allows us (i) to model imbalance, and at the same time (ii) avoid the situation in which a few large datasets could completely dominate the training and evaluation of the model. The distribution of datasets by source, number of samples and number of variables is shown in Figure 2. The datasets, along with their sample count, number of variables and source information are listed in Table 4 of Appendix C.

Baselines: We compare NIAQUE against traditional tree-based models and deep learning approaches: a) Tree-based models: XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2019) b) Deep learning: Transformer encoder with NIAQUE quantile decoder (details in Appendix H). We evaluate three training scenarios: a) Global models (denoted by the -Global suffix): trained jointly on all datasets. b) Domain-specific models (suffix -Domain): trained on datasets from the same domain (e.g., housing, medical). c) Local models (suffix -Local): trained individually per dataset. XGBoost and CatBoost are trained using multi-quantile loss with fixed quantiles, with additional quantiles obtained through linear interpolation. LightGBM, is trained with separate models per quantile. For training global tree-based models, we construct a unified table containing samples from all datasets, filling missing features with NULL values.

Implementation Details: NIAQUE uses encoder and decoder containing 4 residual blocks, 2 layers each with latent dimension $E = 1024$ and input embedding size $E_{in} = 64$. Training uses Adam optimizer with initial learning rate 10^{-4} and batch size 512. The learning rate is reduced by $10\times$ at 500k, 600k, and 700k batches. We apply feature dropout with rate 0.2. Hyperparameters are selected using validation split and metrics are computed on the test split. Training requires approximately 24 hours for NIAQUE and 48 hours for Transformer on 4×V100 GPUs. In comparison, XGBoost training takes about 30 minutes on a one V100 for 3 quantiles, scaling linearly with the number of quantiles. All models are trained on the same train splits, with samples drawn uniformly at random across datasets. During training, quantile values are randomly generated for each instance in a batch.

5.2 Results

We evaluate NIAQUE's cross-dataset learning capabilities through following complementary analyses: (i) large-scale multi-dataset co-training, (ii) domain-specific co-training and (iii) transfer of pretrained model to unseen tasks via fine-tuning.

Table 1: Performance comparison, co-training across all 101 datasets. Lower values are better for all metrics except COVERAGE @ 95 (target: 95). For BIAS, lower absolute values are better.

Model	SMAPE	AAD	RMSE	BIAS	COV@95	CRPS
XGBoost-Global	31.4	0.574	1.056	-0.15	94.6	0.636
XGBoost-Local	25.6	0.433	0.883	-0.03	90.8	0.334
LightGBM-Global	27.5	0.475	0.930	-0.06	94.8	0.426
LightGBM-Local	25.7	0.427	0.865	-0.03	91.5	0.327
CatBoost-Global	31.3	0.561	1.030	-0.12	94.9	0.443
CatBoost-Local	24.3	0.408	0.840	-0.03	92.7	0.315
Transformer-Local	26.9	0.462	0.904	-0.05	93.6	0.329
Transformer-Global	23.1	0.383	0.806	-0.01	94.6	0.272
NIAQUE-Local	22.8	0.377	0.797	-0.03	94.9	0.267
NIAQUE-Global	22.1	0.367	0.787	-0.02	94.6	0.261

Cross-Dataset Learning. To evaluate NIAQUE's ability to handle large-scale multi-dataset learning, we conduct experiments across all 101 datasets simultaneously. Table 1 presents results, aggregated across datasets at sample level, comparing global and local training scenarios for various models. NIAQUE-Global achieves the best performance, outperforming both traditional tree-based methods and the Transformer baseline. Notably, while tree-based methods show better performance in local training compared to their global variants (e.g., CatBoost-Local SMAPE: 24.3 vs CatBoost-Global: 31.3), NIAQUE maintains superior performance in both scenarios, with its global model outperforming its local counterpart. Furthermore, NIAQUE maintains reliable uncertainty quantification across all scenarios, with coverage staying close to the target 95% level and consistently lower CRPS values compared to baselines. These results confirm NIAQUE's capacity to leverage cross-dataset learning effectively, maintaining or even improving performance on individual tasks through robust feature representations that generalize across diverse datasets and domains. This experiment also shows the principal inability of tree-based models to operate in cross-dataset learning scenarios, emphasizing their inability to develop joint representations across heterogeneous problems.

Domain-Specific Results (Tables 5 and 6 in Appendix D.1) focus on *House Price Prediction* and *Energy and Efficiency* domains, showing the ability of the model to effectively leverage additional information from domain-specific datasets to improve on target task.

Adaptation to New Tasks on TabRegSet-101. To evaluate NIAQUE's transfer learning capabilities on unseen tasks, we randomly split our collection of 101 datasets into 80 pretraining datasets and 21 held-out test datasets. We compare two scenarios: training from scratch (NIAQUE-Scratch) and fine-tuning a pretrained model (NIAQUE-Pretrain). The pretrained model is first trained on the 80 datasets and then fine-tuned on each held-out dataset using a 10 times smaller learning rate. To assess the impact of data scarcity, we evaluate both models by varying the fine-tuning data proportion (p_s) of the held-out datasets while maintaining constant test sets. Results in Table 2 demonstrate that: 1) The pretrained model consistently outperforms training from scratch across all metrics. 2) The performance gap widens as training data becomes scarcer (smaller p_s). 3) Both models maintain reliable uncertainty estimates, as evidenced by COVERAGE @ 95 values. These results validate that NIAQUE effectively transfers pretrained knowledge to novel regression tasks, with improvements particularly pronounced in low-data scenarios. Note that these results are not directly comparable with those in Table 1, as they are based on different dataset splits (21 vs. 101 datasets).

Adaptation to New Tasks on Kaggle Competitions. To validate NIAQUE's practical effectiveness in the wild, we evaluate its performance in recent Kaggle competitions: Regression with an Abalone Dataset (Reade & Chow, 2024a), Regression with a Flood Prediction Dataset (Reade & Chow, 2024b) Our approach involves two stages: pretraining and fine-tuning. First, we pretrain NIAQUE on TabRegSet-101 using our quantile loss framework. Then, we fine-tune the pretrained model on the competition's training data, optimizing for target metric. To systematically evaluate the impact of proposed pretraining strategy and architecture, we show NIAQUE-Scratch baseline (trained only on competition data, no pretraining), a

Table 2: Transfer learning results on held-out datasets. p_s represents the proportion of training data used for fine-tuning, ranging from 0.05 (5%) to 1.0 (100%). Lower values are better for all metrics except COVERAGE @ 95 (target: 95). For BIAS, lower absolute values are better.

	NIAQUE	p_s=0.05	0.1	0.25	0.5	1.0
SMAPE	Scratch	28.0	24.7	21.7	20.8	19.4
	Pretrain	23.5	21.9	20.3	18.7	17.7
AAD	Scratch	0.71	0.60	0.56	0.54	0.49
	Pretrain	0.61	0.57	0.54	0.50	0.47
RMSE	Scratch	1.23	1.10	1.06	1.04	0.96
	Pretrain	1.11	1.08	1.04	0.97	0.94
BIAS	Scratch	-0.06	-0.04	-0.04	0.02	-0.04
	Pretrain	-0.06	-0.07	-0.06	-0.06	-0.04
CRPS	Scratch	0.488	0.423	0.392	0.383	0.351
	Pretrain	0.427	0.404	0.380	0.354	0.334
COV@95	Scratch	93.3	93.0	94.4	93.1	94.4
	Pretrain	95.3	94.4	94.2	93.9	94.6

Table 3: Performance comparison on Kaggle competition datasets: Abalone (RMSLE, lower is better), Flood Prediction (R2, higher is better).

Model	Abalone RMLSE	Flood Prediction R2 Score
XGBoost (siukeitin); Sayed (2024)	0.15019	0.842
LightGBM (dataWr3cker); Masoudi (2024)	0.14914	0.766
CatBoost Wate (2024); Milind (2024)	0.14783	0.845
TabNet (siukeitin)	0.15481	0.842
TabDPT	0.15026	0.804
TabPFN	0.15732	0.431
NIAQUE-Scratch	0.15047	0.865
NIAQUE-Pretrain	0.14808	0.867
Winner (Heller, 2024; Aldparis, 2024)	0.14374	0.869

number of tree-based baselines as well as TabDPT and TabPFN pretrained models. While TabDPT and TabPFN show promising results on small datasets, they face significant scalability challenges—TabPFN is limited to 10,000 samples and TabDPT requires substantial context size reduction for large datasets (details in Appendix 5.3). On the other hand, our approach shows very strong scalability and accuracy results on the competition datasets, outperforming vanilla tree-based models as well as TabDPT and TabPFN baselines. Additional results in Appendices D.2.1, D.2.2 show how our approach, without significant manual interventions, further benefits from advanced automatic feature engineering (OpenFE (Zhang et al., 2023)) and ensembling thereby rivalling results of human competitors. These results are particularly significant given that neural networks were generally considered ineffective for these competitions.

Learned Representations are studied qualitatively in Fig. 3 (left) showing UMAP projections (McInnes et al., 2018) of dataset row embeddings derived from NIAQUE's feature encoder. The encoder maps input features $\mathbf{x}_i$ to a fixed-dimensional latent space using a prototype-based aggregation mechanism. The resulting UMAP visualization reveals distinct dataset-specific clusters, indicating that NIAQUE learns representations that capture dataset-specific characteristics while maintaining a shared latent space that enables effective transfer learning.

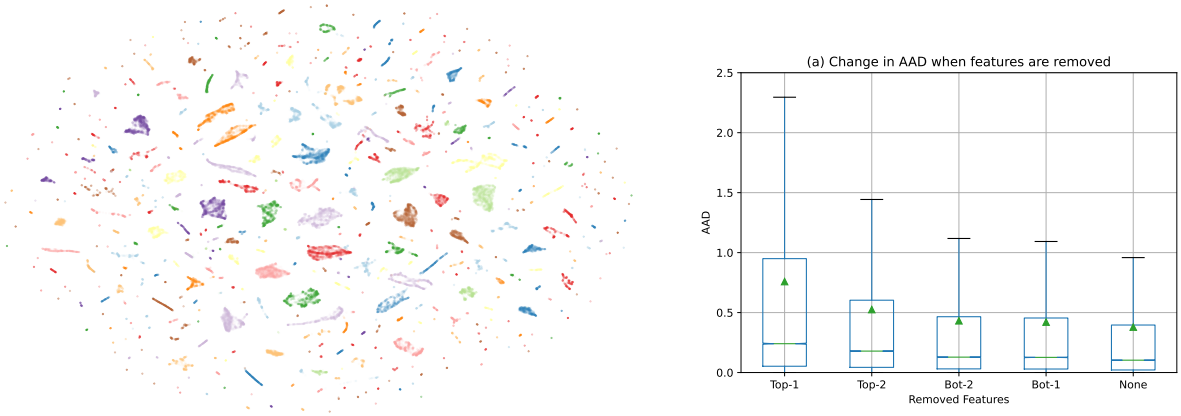


Figure 3: UMAP projections of embeddings derived from NIAQUE's feature encoder for each sample, colored by dataset (left). NIAQUE accuracy response to the removal of input features by importance (right). Top-rated features have the greatest impact on AAD degradation when removed.

Feature Importance is based on the inverse of the average confidence interval derived from feature's marginal distribution, as detailed in Equation equation 17. Qualitatively, Fig. 3 (right), indicates that features with higher weights (i.e., smaller average confidence intervals) are most critical to prediction accuracy—removing these features significantly increases the AAD metric, whereas eliminating features with lower weights has minimal impact. Quantitatively, two-sided t-test comparing the impact of removing the most vs. least important features (Top-1 vs. Bot-1 AAD values in Fig. 3) ($t = -50.24$, p-value ≈ 0) along with the effect size (Cohen's $d = 0.22$) indicate significant and practically impactful effect across datasets. Additionally, we computed SHAP values with shap.SamplingExplainer across 101 datasets and used them as a reference ranking. For each dataset, we computed the NDCG score between the SHAP ranking and our model-native importance ranking, then averaged the results. The average NDCG was 0.899, while computation time was reduced from 3.5h $\times$ 8 GPUs to 20s $\times$ 1 GPU for all 101 datasets. The NDCG score of 0.9 generally demonstrates very strong ranking alignment with established attribution method at a fraction of the cost.

Ablation Studies (Appendices H–J) support NIAQUE's design choices. We observe: (1) our encoder outperforms attention-based alternatives; (2) log-transforming inputs (Eq. equation 10) enhances both stability and accuracy; (3) the model is robust to hyperparameter changes and benefits from greater depth; and (4) incorporating single-feature samples enables interpretability without degrading performance.

5.3 Scalability Analysis of TabPFN and TabDPT

TabPFN and TabDPT mark key advances in tabular deep learning, showing strong results on small-scale datasets. However, our Kaggle experiments reveal major scalability bottlenecks, limiting their practicality in larger real-world settings. This highlights the need for scalable alternatives like NIAQUE.

Limitations of TabPFN. Despite TabPFN's impressive few-shot performance, its applicability is bounded by architectural and implementation constraints, as documented in the official repository Müller et al. (2025): (i) maximum support for 10,000 training samples, (ii) limit of 500 features per instance, (iii) assumes all features are numerical, requiring preprocessing for categorical inputs. Furthermore, to adapt TabPFN for our use cases, we employed several workarounds: (i) for large datasets (e.g., the flood prediction dataset with 1.12M samples), we applied Random Forest-based subsampling, following official guidelines Müller et al. (2025), (ii) categorical features were ordinally encoded to conform to TabPFN's numerical input requirement. These constraints, especially the aggressive downsampling, likely contributed to TabPFN's suboptimal performance on large-scale tasks (e.g., R2 score of 0.431 on flood prediction).

Scalability Challenges in TabDPT. TabDPT, based on a transformer backbone, encounters scalability issues typical of attention-based models: (i) memory consumption grows quadratically with context size due to self-attention, (ii) inference time scales poorly with dataset size, (iii) performance is sensitive to reductions in context size, making trade-offs between scalability and accuracy non-trivial. In our experiments we had to reduce the context window for the flood prediction dataset to fit within memory constraints. We reduce the context size from 8,192 in powers of 2. Context size of 1024 finally works. This reduction correlated with a decline in model performance (R2 score dropped to 0.804). The compute cost of running extensive hyperparameter tuning on large datasets proved impractical. TabDPT takes 12 hours of inference time for each hyperparameter configuration for flood prediction datatset on a V100 GPU.

6 Discussion

We believe that our results applying NIAQUE to the 101 dataset benchmark lay out the stepping stone for the development of probabilistic meta-models eventually possessing the following key properties. **Scalability**: A unified model shares computational resources to address multiple regression tasks, optimizing resource utilization and reducing the operational costs of maintaining separate models. **Data Efficiency**: Training on diverse tasks introduces strong regularization effects, and we expect existing datasets to be repurposed to solve emerging problems, promoting data reuse and recycling. **Representation and Generalization**: A model trained across multiple datasets uncovers generalizable representations of regression tasks and ways of solving them, acquiring the ability to apply this knowledge across datasets. As an example, while TabPFN and TabDPT perform well on small datasets, they face significant scalability challenges with high-dimensional, large-scale data. NIAQUE demonstrates that scalable, accurate, and interpretable neural models are viable without relying on heavy preprocessing or memory-intensive components. These results position NIAQUE not just as a practical alternative, but also as evidence that our approach offers a fruitful research path beyond current paradigms.

Limitations. While we significantly expand the scope of cross-dataset probabilistic model training by applying our neural model to a 101-dataset benchmark, this remains a limited effort. It is still unclear how many datasets are required for a regression model to be considered foundational for solving, for instance, 80% of industry problems. What level of dataset diversity is necessary? Will millions or billions of unrelated datasets be required, or would 10,000 overlapping datasets suffice? Defining and evaluating global success in this context remains an open question, necessitating further research. Furthermore, our findings have implications for designing machine learning deployments based on unified models that address multiple regression tasks. We expect that this will eventually lead to improved operational efficiency and accuracy of the models. However, this could also contribute to the centralization of power among a few large entities. In this context, risk mitigation strategies include (i) improving model computational efficiency and (ii) publicly releasing data, model training code and pretrained models. Additionally, multi-task learning on multiple datasets may introduce new biases not present in locally trained models, making interpretability and fairness research critical. We explore some interpretability aspects in this paper, and further research on interpretability and fairness in large probabilistic regression models pretrained across multiple datasets seems to be an important area for future work.

7 Conclusion

Our study demonstrates that NIAQUE enables effective transfer learning for tabular probabilistic regression, providing additional evidence to further challenge the belief that tabular data resists neural modeling and generalization. Across 101 datasets and a Kaggle case studies, NIAQUE shows strong empirical performance, delivering scalability via a unified architecture, data efficiency through cross-dataset pretraining, and robust generalization across tasks. Its probabilistic formulation further supports uncertainty quantification and feature importance estimation, useful for real-world deployment. While promising, open questions remain regarding the optimal scale of transfer, the design of universally effective feature preprocessing, and the theoretical principles underlying cross-domain generalization. Overall, our findings suggest a path forward for building interpretable tabular foundation models that scale well in large-scale downstream tasks.

References

Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, and Salvador García. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Multiple Valued Log. Soft Comput.*, 17(2–3):255–287, 2011.

Aldparis. 1st place solution for the regression with a flood prediction dataset. `https://www.kaggle.com/competitions/playground-series-s4e5/discussion/509043`, 2024. Kaggle.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Sercan Ö. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proc. AAAI*, pp. 6679–6687, May 2021.

Sourav Banerjee. House rent prediction dataset, 2022. URL `https://www.kaggle.com/datasets/iamsouravbanerjee/house-rent-prediction-dataset`.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, August 2016. doi: 10.1145/2939672.2939785. URL `http://dx.doi.org/10.1145/2939672.2939785`.

Gourab Roy (dataWr3cker). ps-s4e4: Eda | lightgbm | importance plots. `https://www.kaggle.com/code/gourabr0y555/ps-s4e4-eda-lightgbm-importance-plots`, 2024. Kaggle.

Dean De Cock. Ames, iowa: Alternative to the boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3), 2011.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *NAACL-HLT (1)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Luc Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, NY, USA, 1986.

Brijlal Dhankour. Flood prediction factors. `https://www.kaggle.com/datasets/brijlaldhankour/flood-prediction-factors`, 2024. Kaggle.

Dheeru Dua and Casey Graff. Uci machine learning repository: Abalone data set, 2019. URL `https://archive.ics.uci.edu/ml/datasets/abalone`.

Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, January 2021.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In *Proc. ICML*, volume 80, pp. 1704–1713. PMLR, Jul 2018a. URL `https://proceedings.mlr.press/v80/garnelo18a.html`.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes, 2018b. URL `https://arxiv.org/abs/1807.01622`.

Azul Garza and Max Mergenthaler-Canseco. Timegpt-1, 2023.

Tilmann Gneiting and Roopesh Ranjan. Comparing density forecasts using threshold-and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, 29(3):411–422, 2011.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Proc. NeurIPS*, pp. 507–520, 2022.

Johannes Heller. 1st place solution for the regression with an abalone dataset competition. https://www.kaggle.com/competitions/playground-series-s4e4/discussion/499174, 2024. Kaggle.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *Proc. ICLR*, 2023.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. TabTransformer: Tabular data modeling using contextual embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7671–7679. AAAI Press, 2021.

Kaggle. Kaggle datasets, 2024. URL https://www.kaggle.com/datasets.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-lojasiewicz condition. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 795–811, Cham, 2016. Springer International Publishing.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Roman Levin, Valeriia Cherepanova, Avi Schwarzschild, Arpit Bansal, C Bayan Bruss, Tom Goldstein, Andrew Gordon Wilson, and Micah Goldblum. Transfer learning with deep tabular models. In *Proc. ICLR*, 2023.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L Caterini. Tabdpt: Scaling tabular foundation models. *arXiv preprint arXiv:2410.18164*, 2024.

Ali Masoudi. Xgb|cat|lightgbm. https://www.kaggle.com/code/alimsd77/xgb-cat-lightgbm#lightgbm, 2024. Kaggle.

Julian McAuley, Rahul Pandey, and Jure Leskovec. Amazon price and product data: Electronic commerce, 2015. URL https://www.kaggle.com/datasets/skillsmuggler/amazon-products-dataset.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, February 2018.

Jay Milind. Baseline v1 | catboost. https://www.kaggle.com/code/jaymilindpadloskar/baseline-v1-catboost, 2024. Kaggle.

Felix Müller, Christian Holtz, Emir Akyürek, Lukas Lorbeer, and Frank Hutter. TabPFN: A Transformer that solves small tabular classification problems in a second. https://github.com/PriorLabs/TabPFN, 2025. URL https://github.com/PriorLabs/TabPFN.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proc. ICCV*, October 2019.

Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. Pmlb: A large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(36), 2017a.

Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10 (1):36, Dec 2017b. ISSN 1756-0381. doi: 10.1186/s13040-017-0154-4. URL https://doi.org/10.1186/s13040-017-0154-4.

OpenML. Wind dataset, n.d. URL https://www.openml.org/data/download/52615/wind.arff.

Boris N. Oreshkin, Florent Bocquelet, Félix G. Harvey, Bay Raitt, and Dominic Laflamme. Protores: Proto-residual network for pose authoring via learned inverse kinematics. In *Proc. ICLR*, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proc. AAAI*, 2018.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features, 2019.

C. Qi, Hao Su, Kaichun Mo, and L. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. CVPR*, pp. 77–85, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, volume 139, pp. 8748–8763, Jul 2021.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Marvin Sun, et al. Scalable and accurate deep learning with electronic health records. *Nature Medicine*, 24(7):1337–1340, 2018.

Walter Reade and Ashley Chow. Regression with an abalone dataset. https://kaggle.com/competitions/playground-series-s4e4, 2024a. Kaggle.

Walter Reade and Ashley Chow. Regression with a flood prediction dataset. https://www.kaggle.com/competitions/playground-series-s4e5, 2024b. Kaggle.

Joseph D Romano, Trang T Le, William La Cava, John T Gregg, Daniel J Goldberg, Praneel Chakraborty, Natasha L Ray, Daniel Himmelstein, Weixuan Fu, and Jason H Moore. Pmlb v1.0: an open source dataset collection for benchmarking machine learning methods. *arXiv preprint arXiv:2012.00058v2*, 2021.

Zeyad Sayed. Flood forecasting with xgboost. https://www.kaggle.com/code/zeyadsayedadbullah/flood-forecasting-with-xgboost, 2024. Kaggle.

Broccoli Beef (siukeitin). Why nns is better than gbds? https://www.kaggle.com/competitions/playground-series-s4e4/discussion/496471#2767909, 2024. Kaggle.

Slawek Smyl, Boris N. Oreshkin, Paweł Pełka, and Grzegorz Dudek. Any-quantile probabilistic forecasting of short-term electricity demand, 2024. URL https://arxiv.org/abs/2404.17451.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Proc. NIPS*, pp. 4080–4090, 2017.

Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, One-stage, Regression of Multiple 3D People. In *ICCV*, 2021.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.264119.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Proc. NeurIPS*, volume 30, 2017.

Suraj Wate. S4e4 | abalone | catboost. https://www.kaggle.com/code/surajwate/s4e4-abalone-catboost, 2024. Kaggle.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Proc. NeurIPS*, volume 30. Curran Associates, Inc., 2017.

Tianping Zhang, Zheyu Zhang, Zhiyuan Fan, Haoyan Luo, Fengyuan Liu, Qian Liu, Wei Cao, and Jian Li. Openfe: automated feature generation with expert-level performance. In *Proc. ICML*, 2023.

A Definitions of metrics

We employ the following metrics to evaluate the accuracy of point predictions: Symmetric Mean Absolute Percentage Error (sMAPE), Average Absolute Deviation (AAD), Root Mean Square Error (RMSE), Root Mean Square Logarithmic Error (RMSLE).

1. Symmetric Mean Absolute Percentage Error (sMAPE):

$$\text{sMAPE} = \frac{2}{S} \sum_{i=1}^{S} \frac{|y_i - \hat{y}_{i,0.5}|}{|y_i| + |\hat{y}_{i,0.5}|} \cdot 100\% \,. \tag{18}$$

2. Average Absolute Deviation (AAD):

$$\text{AAD} = \frac{1}{S} \sum_{i=1}^{S} |y_i - \hat{y}_{i,0.5}| \,. \tag{19}$$

3. Bias:

$$\text{BIAS} = \frac{1}{S} \sum_{i=1}^{S} \hat{y}_{i,0.5} - y_i \,. \tag{20}$$

4. Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{S} \sum_{i=1}^{S} (y_i - \hat{y}_{i,0.5})^2} \,. \tag{21}$$

5. Root Mean Square Logarithmic Error (RMSLE):

$$\text{RMSLE} = \sqrt{\frac{1}{S} \sum_{i=1}^{S} (\log(y_i + 1) - \log(\hat{y}_{i,0.5} + 1))^2} \,. \tag{22}$$

B Proof of Theorem 1

Theorem. *Let F be a probability measure over variable y such that inverse F^{-1} exists and let $P_{y,\mathbf{x}}$ be the joint probability measure of variables $\mathbf{x}, y$. Then the expected loss, $\mathbb{E}\,\rho(y, F^{-1}(q))$, is minimized if and only if:*

$$F = P_{y|\mathbf{x}}\,. \tag{23}$$

Additionally:

$$\min_F \mathbb{E}\,\rho(y, F^{-1}(q)) = \mathbb{E}_{\mathbf{x}} \frac{1}{2} \int_{\mathbb{R}} P_{y|\mathbf{x}}(z)(1 - P_{y|\mathbf{x}}(z))\,dz\,. \tag{24}$$

Proof. First, combining (9) with the L2 representation of CRPS equation 2 we can write:

$$\mathbb{E}\,\rho(y, F^{-1}(q)) = \mathbb{E}_{\mathbf{x},y} \frac{1}{2} \int_{\mathbb{R}} \left(F(z) - \mathbb{1}_{\{z \geq y\}}\right)^2 \mathrm{d}z \tag{25}$$

$$= \mathbb{E}_{\mathbf{x}} \mathbb{E}_{y|\mathbf{x}} \frac{1}{2} \int_{\mathbb{R}} F^2(z) - 2F(z)\mathbb{1}_{\{z \geq y\}} + \mathbb{1}_{\{z \geq y\}} \mathrm{d}z \tag{26}$$

$$= \mathbb{E}_{\mathbf{x}} \frac{1}{2} \int_{\mathbb{R}} F^2(z) - 2F(z)\mathbb{E}_{y|\mathbf{x}}\mathbb{1}_{\{z \geq y\}} + \mathbb{E}_{y|\mathbf{x}}\mathbb{1}_{\{z \geq y\}} \mathrm{d}z \tag{27}$$

$$= \mathbb{E}_{\mathbf{x}} \frac{1}{2} \int_{\mathbb{R}} F^2(z) - 2F(z)P_{y|\mathbf{x}}(z) + P_{y|\mathbf{x}}(z) \mathrm{d}z. \tag{28}$$

Here we used the law of total expectation and Fubini theorem to exchange the order of integration and then used the fact that $\mathbb{E}_{y|\mathbf{x}}\mathbb{1}_{\{z \geq y\}} = P_{y|\mathbf{x}}(z)$. Completing the square we further get:

$$\mathbb{E}\,\rho(y, F^{-1}(q)) = \mathbb{E}_{\mathbf{x}} \frac{1}{2} \int_{\mathbb{R}} F^2(z) - 2F(z)P_{y|\mathbf{x}}(z) + P_{y|\mathbf{x}}(z) + P^2_{y|\mathbf{x}}(z) - P^2_{y|\mathbf{x}}(z) \mathrm{d}z \tag{29}$$

$$= \mathbb{E}_{\mathbf{x}} \frac{1}{2} \int_{\mathbb{R}} (F(z) - P_{y|\mathbf{x}}(z))^2 + P_{y|\mathbf{x}}(z) - P^2_{y|\mathbf{x}}(z) \mathrm{d}z \tag{30}$$

$F = P_{y|\mathbf{x}}$ is clearly the unique minimizer of the last expression since $\int_{\mathbb{R}} (F(z) - P_{y|\mathbf{x}}(z))^2 \mathrm{d}z > 0, \forall F \neq P_{y|\mathbf{x}}$. □

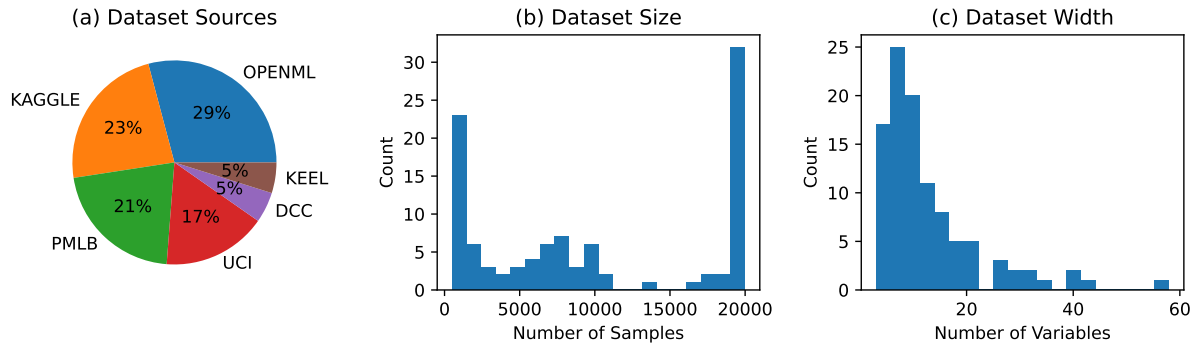


Figure 4: Statistics of the evaluation dataset: (a) distribution by source, (b) dataset sizes, and (c) feature counts.

C TabRegSet-101 Details

We introduce TabRegSet-101 (Tabular Regression Set 101), a curated collection of 101 publicly available regression datasets gathered from UCI Kelly et al. (2017), Kaggle Kaggle (2024), PMLB Romano et al. (2021); Olson et al. (2017b), OpenML Vanschoren et al. (2013), and KEEL Alcalá-Fdez et al. (2011). These datasets span diverse domains, including *Housing and Real Estate*, *Energy and Efficiency*, *Retail and Sales*, *Computer Systems*, *Physics Models* and *Medicine* and exhibit different characteristics in terms of sample size and feature dimensionality (Fig. 4). The datasets, along with their sample count, number of variables and source information are listed in Table 4.

We focus specifically on the regression task in which the target variable is continuous or, if it has limited number of levels, these are ordered such as student exam scores or wine quality. The target variable in each dataset is normalized to the [0, 10] range and the independent variables are used as is, raw. The target variable scaling is applied to equalize the contributions of the evaluation metrics from each dataset. Datasets have variable number of samples, the lowest being just below 1000. For very large datasets we limit the number of samples used in our benchmark to be 20,000 by subsampling uniformly at random. This allows us (i) to model imbalance, and at the same time (ii) avoid the situation in which a few large datasets could completely dominate the training and evaluation of the model. The distribution of datasets by source, number of samples and number of variables is shown in Figure 4.

Table 4: The list of datasets comprising TabRegSet-101 benchmark

id	Name	n_samples	n_vars	source	URL
0	Abalone	4177	7	uci	https://archive.ics.uci.edu/static/public/1/data.csv
1	Student_Performance	649	29	uci	https://archive.ics.uci.edu/static/public/320/data.csv
2	Infrared_Thermography_Temperature	1020	32	uci	https://archive.ics.uci.edu/static/public/925/data.csv
3	Parkinsons_Telemonitoring	5875	18	uci	https://archive.ics.uci.edu/static/public/189/data.csv
4	Energy_Efficiency	768	7	uci	https://archive.ics.uci.edu/static/public/242/data.csv
5	1027_ESL	488	3	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1027_ESL/1027_ESL.tsv.gz
6	1028_SWD	1000	9	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1028_SWD/1028_SWD.tsv.gz
7	1029_LEV	1000	3	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1029_LEV/1029_LEV.tsv.gz
8	1030_ERA	1000	3	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1030_ERA/1030_ERA.tsv.gz
9	1199_BNG_echoMonths	17496	8	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1199_BNG_echoMonths/1199_BNG_echoMonths.tsv.gz
10	197_cpu_act	8192	20	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/197_cpu_act/197_cpu_act.tsv.gz
11	225_puma8NH	8192	7	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/225_puma8NH/225_puma8NH.tsv.gz
12	227_cpu_small	8192	11	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/227_cpu_small/227_cpu_small.tsv.gz
13	294_satellite_image	6435	35	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/294_satellite_image/294_satellite_image.tsv.gz
14	344_mv	20000	9	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/344_mv/344_mv.tsv.gz
15	503_wind	6574	13	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/503_wind/503_wind.tsv.gz
16	529_pollen	3848	3	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/529_pollen/529_pollen.tsv.gz
17	537_houses	20000	7	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/537_houses/537_houses.tsv.gz
18	547_no2	500	6	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/547_no2/547_no2.tsv.gz
19	564_fried	20000	9	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/564_fried/564_fried.tsv.gz
20	595_fri_c0_1000_10	1000	9	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/595_fri_c0_1000_10/595_fri_c0_1000_10.tsv.gz

Continued on next pag

Table 4: The list of datasets comprising TabRegSet-101 benchmark

	name	n_samples	n_vars	source	url
21	593_fri_c1_1000_10	1000	9	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/593_fri_c1_1000_10/593_fri_c1_1000_10.tsv.gz
22	1193_BNG_lowbwt	20000	8	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1193_BNG_lowbwt/1193_BNG_lowbwt.tsv.gz
23	1201_BNG_breastTumor	20000	8	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1201_BNG_breastTumor/1201_BNG_breastTumor.tsv.gz
24	1203_BNG_pwLinear	20000	9	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/1203_BNG_pwLinear/1203_BNG_pwLinear.tsv.gz
25	215_2dplanes	20000	9	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/215_2dplanes/215_2dplanes.tsv.gz
26	218_house_8L	20000	7	pmlb	https://github.com/EpistasisLab/penn-ml-benchmarks/raw/master/datasets/218_house_8L/218_house_8L.tsv.gz
27	QsarFishToxicity	908	5	uci	https://archive.ics.uci.edu/static/public/504/qsar+fish+toxicity.zip
28	concrete_compressive_strength	1030	7	uci	https://archive.ics.uci.edu/static/public/165/concrete+compressive+strength.zip
29	PRODUCTIVITY	1197	12	uci	https://archive.ics.uci.edu/static/public/597/productivity+prediction+of+garment+employees.zip
30	CCPP	9568	3	uci	https://archive.ics.uci.edu/static/public/294/combined+cycle+power+plant.zip
31	AIRFOIL	1503	4	uci	https://archive.ics.uci.edu/static/public/291/airfoil+self+noise.zip
32	TETOUAN	20000	6	uci	https://archive.ics.uci.edu/static/public/849/power+consumption+of+tetouan+city.zip
33	BIAS_CORRECTION	7725	22	uci	https://archive.ics.uci.edu/static/public/514/bias+correction+of+numerical+prediction+model+temperature+forecast.zip
34	APARTMENTS	10000	10	uci	https://archive.ics.uci.edu/static/public/555/apartment+for+rent+classified.zip
35	MedicalCost	1338	5	kaggle	kaggledatasetsdownload-dmirichoi0218/insurance
36	Vehicle	2059	18	kaggle	kaggledatasetsdownload-dnehalbirla/vehicle-dataset-from-cardekho
37	LifeExpectancy	2928	18	kaggle	kaggledatasetsdownload-dkumarajarshi/life-expectancy-who
38	CalHousing	20000	7	dcc	https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.tgz
39	Ailerons	7154	39	dcc	https://www.dcc.fc.up.pt/~ltorgo/Regression/ailerons.tgz
40	DeltaElevators	9517	5	dcc	https://www.dcc.fc.up.pt/~ltorgo/Regression/delta_elevators.tgz
41	Pole	10000	25	dcc	https://www.dcc.fc.up.pt/~ltorgo/Regression/pol.tgz
42	Kinematics	8192	7	dcc	https://www.dcc.fc.up.pt/~ltorgo/Regression/kinematics.tar.gz
43	BigMartSales	8523	10	kaggle	kaggledatasetsdownload-dbrijbhushannanda1979/bigmart-sales-data
44	VideoGameSales	16598	3	kaggle	kaggledatasetsdownload-dgregorut/videogamesales
45	NewsPopularity	20000	58	uci	https://archive.ics.uci.edu/static/public/332/online+news+popularity.zip

Continued on next pag

Table 4: The list of datasets comprising TabRegSet-101 benchmark

	name	n_samples	n_vars	source	url
46	Wizmir	1461	8	keel	https://sci2s.ugr.es/keel/dataset/data/regression/wizmir.zip
47	Ele2	1056	3	keel	https://sci2s.ugr.es/keel/dataset/data/regression/ele-2.zip
48	Treasury	1049	14	keel	https://sci2s.ugr.es/keel/dataset/data/regression/treasury.zip
49	Mortgage	1049	14	keel	https://sci2s.ugr.es/keel/dataset/data/regression/mortgage.zip
50	Laser	993	3	keel	https://sci2s.ugr.es/keel/dataset/data/regression/laser.zip
51	SpaceGa	3107	5	openml	https://www.openml.org/data/download/52619/space_ga.arff
52	VisualizingSoil	8641	3	openml	https://www.openml.org/data/download/52988/visualizing_soil.arff
53	Diamonds	20000	8	openml	https://www.openml.org/data/download/21792853/dataset.arff
54	TitanicFare	1307	6	openml	https://www.openml.org/data/download/20649205/file277c5e2b70e8.arff
55	Sulfur	10081	5	openml	https://www.openml.org/data/download/2095629/phpBXEqg1.arff
56	Debutanizer	2394	6	openml	https://www.openml.org/data/download/2096280/phpWT77lf.arff
57	Fardamento	6277	5	openml	https://www.openml.org/data/download/21854531/fardamento_saidas_19_20a20maio.arff
58	ProteinTertiary	20000	8	openml	https://api.openml.org/data/download/22111827/file22f167620a212.arff
59	BrazilianHouses	10692	7	openml	https://api.openml.org/data/download/22111854/file22f1627e4a960.arff
60	Cps88Wages	20000	5	openml	https://api.openml.org/data/download/22111848/file22f161d4b5556.arff
61	CPMP-2015	2108	25	openml	https://www.openml.org/data/download/21377442/file16a868cf35f5.arff
62	NASA-PHM2008	20000	16	openml	https://www.openml.org/data/download/22045221/dataset.arff
63	Wind	6574	12	openml	https://www.openml.org/data/download/52615/wind.arff
64	NewFuelCar	20000	17	openml	https://www.openml.org/data/download/21230500/pruebaconvonline.csv.arff
65	MiamiHousing	13932	14	openml	https://www.openml.org/data/download/22047757/miami2016.arff
66	BlackFriday	20000	8	openml	https://www.openml.org/data/download/21230845/file639340bd9ca9.arff
67	IEEE80211aaGATS	5296	28	openml	https://www.openml.org/data/download/22101884/dataset.arff
68	Yprop41	8885	41	openml	https://api.openml.org/data/download/22111920/dataset.arff
69	Sarcos	20000	20	openml	https://api.openml.org/data/download/22111840/file22f166a1669bb.arff
70	ZurichDelays	20000	16	openml	https://www.openml.org/data/download/21854423/file86eb92864fd.arff
71	1000-Cameras	1015	13	openml	https://www.openml.org/data/download/22102539/dataset.arff
72	GridStability	10000	11	openml	https://api.openml.org/data/download/22111837/file22f1652de1c8a.arff
73	PumaDyn32nh	8192	31	openml	https://api.openml.org/data/download/22111845/file22f161b261f3b.arff

Continued on next pag

Table 4: The list of datasets comprising TabRegSet-101 benchmark

	name	n_samples	n_vars	source	url
74	Fifa	19178	27	openml	https://api.openml.org/data/download/22111894/file10aca711933d5.arff
75	WhiteWine	4898	10	openml	https://api.openml.org/data/download/22111835/file22f16150a82cd.arff
76	RedWine	1599	10	openml	https://api.openml.org/data/download/22111836/file22f162b311c38.arff
77	FpsBenchmark	20000	42	openml	https://api.openml.org/data/download/22111856/file22f1639d20997.arff
78	KingCountyHousing	20000	20	openml	https://api.openml.org/data/download/22111853/file22f167bd414f1.arff
79	AvocadoPrices	18249	12	kaggle	kaggledatasetsdownload-dneuromusic/avocado-prices
80	Transcoding	20000	18	uci	https://archive.ics.uci.edu/static/public/335/online+video+characteristics+and+transcoding+time+dataset.zip
81	house_16H	20000	15	openml	https://www.openml.org/data/download/52752/house_16H.arff
82	Sales	10738	13	openml	https://www.openml.org/data/download/21756753/dataset.arff
83	WalmartSales	6435	8	kaggle	kaggledatasetsdownload-dmikhail1681/walmart-sales
84	UsedCar	6019	11	kaggle	kaggledatasetsdownload-dnitishjolly/used-car-price-prediction
85	HouseRent	4746	11	kaggle	kaggledatasetsdownload-diamsouravbanerjee/house-rent-prediction-dataset
86	LaptopPrice	1273	15	kaggle	kaggledatasetsdownload-dehtishamsadiq/uncleaned-laptop-price-dataset
87	UberFare	20000	8	kaggle	kaggledatasetsdownload-dyasserh/uber-fares-dataset
88	Co2Emission	7385	10	kaggle	kaggledatasetsdownload-ddebajyotipodder/co2-emission-by-vehicles
89	SongPopularity	18835	12	kaggle	kaggledatasetsdownload-dyasserh/song-popularity-dataset
90	Cars	20000	8	kaggle	kaggledatasetsdownload-daishwaryamuthukumar/cars-dataset-audi-bmw-ford-hyundai-skoda-vw
91	GemstonePrice	20000	8	kaggle	kaggledatasetsdownload-dcolearninglounge/gemstone-price-prediction
92	LoanAmount	20000	20	kaggle	kaggledatasetsdownload-dphileinsophos/predict-loan-amount-data
93	SaudiArabiaCars	5507	10	kaggle	kaggledatasetsdownload-dturkibintalib/saudi-arabia-used-cars-dataset
94	GpuKernelPerformance	20000	13	kaggle	kaggledatasetsdownload-drupals/gpu-runtime
95	AmericanHousePrices	20000	10	kaggle	kaggledatasetsdownload-djeremylarcher/american-house-prices-and-demographics-of-top-cit
96	KindleBooks	20000	12	kaggle	kaggledatasetsdownload-dasaniczka/amazon-kindle-books-dataset-2023-130k-books
97	BookSales	1070	8	kaggle	kaggledatasetsdownload-dthedevastator/books-sales-and-ratings
98	CapitalGain	20000	12	kaggle	kaggledatasetsdownload-dminnieliang/adult-data
99	MarketingCampaign	2976	14	kaggle	kaggledatasetsdownload-dahmadazari/marketing-campaign-data
100	CampaignUplift	2000	9	kaggle	kaggledatasetsdownload-dhwwang98/software-usage-promotion-campaign-uplift-model

Table 5: Performance comparison on Wind dataset across point prediction accuracy (SMAPE, AAD, RMSE, BIAS) and distributional accuracy (COV@95, CRPS) metrics. Lower values are better for all metrics except COV@95, where values closer to 95 are optimal. The results with 95% confidence intervals derived from multiple random seed runs.

Model	SMAPE	AAD	RMSE	BIAS	COV@95	CRPS
XGBoost-Local	19.4 ± 0.1	0.607 ± 0.001	0.779 ± 0.004	0.044 ± 0.01	64.9 ± 0.2	0.473 ± 0.001
XGBoost-Domain	19.3 ± 0.2	0.603 ± 0.002	0.779 ± 0.005	0.016 ± 0.01	96.0 ± 0.2	0.462 ± 0.002
XGBoost-Global	20.0 ± 4.4	0.627 ± 0.100	0.811 ± 0.143	0.010 ± 0.05	97.0 ± 0.3	0.527 ± 0.165
LightGBM-Local	18.8 ± 0.2	0.589 ± 0.01	0.753 ± 0.007	0.04 ± 0.01	91.5 ± 1.0	0.429 ± 0.003
LightGBM-Domain	18.9 ± 0.1	0.592 ± 0.004	0.766 ± 0.006	0.05 ± 0.01	93.8 ± 0.6	0.446 ± 0.002
LightGBM-Global	20.8 ± 0.1	0.657 ± 0.002	0.854 ± 0.001	0.02 ± 0.01	97.8 ± 0.3	0.599 ± 0.032
CatBoost-Local	18.9 ± 0.1	0.590 ± 0.001	0.754 ± 0.003	0.046 ± 0.01	93.6 ± 0.1	0.424 ± 0.001
CatBoost-Domain	19.8 ± 0.2	0.617 ± 0.002	0.796 ± 0.004	0.038 ± 0.01	94.8 ± 0.2	0.454 ± 0.002
CatBoost-Global	26.3 ± 0.2	0.832 ± 0.006	1.068 ± 0.009	0.027 ± 0.02	98.3 ± 0.2	0.722 ± 0.004
Transformer-Local	18.5 ± 0.3	0.575 ± 0.008	0.740 ± 0.015	0.025 ± 0.01	94.2 ± 0.3	0.410 ± 0.005
Transformer-Domain	18.5 ± 0.3	0.573 ± 0.008	0.736 ± 0.015	0.022 ± 0.01	94.3 ± 0.3	0.405 ± 0.005
Transformer-Global	18.4 ± 0.3	0.572 ± 0.008	0.733 ± 0.015	0.020 ± 0.01	94.5 ± 0.3	0.401 ± 0.005
NIAQUE-Local	18.8 ± 0.4	0.582 ± 0.012	0.747 ± 0.019	0.045 ± 0.01	95.7 ± 0.4	0.407 ± 0.011
NIAQUE-Domain	18.7 ± 0.2	0.585 ± 0.006	0.760 ± 0.010	0.027 ± 0.01	95.4 ± 0.3	0.415 ± 0.006
NIAQUE-Global	19.0 ± 0.1	0.587 ± 0.002	0.755 ± 0.005	0.031 ± 0.01	93.0 ± 0.2	0.412 ± 0.002

D Supplementary Results

D.1 Domain Adaptation

For domain adaptation, we first focus on *House Price Prediction*, selecting HouseRent Banerjee (2022) (4.7K samples, 11 features) as our representative dataset. This dataset offers a balanced evaluation ground with its moderate sample size and limited feature dimensionality. Next, we focus on *Energy and Efficiency Domain*. We analyze the Wind dataset OpenML (n.d.) (6.5K samples, 12 features) as our representative case study. Similar to HouseRent, this dataset was chosen for its moderate sample size and limited feature dimensionality, providing a balanced evaluation ground. Tables 5, 6 show the performance comparison between NIAQUE and baselines across different training scenarios. The results reinforce our main findings: NIAQUE maintains consistent performance across local, domain, and global training settings (RMSE: 0.747-0.760), while traditional models like CatBoost show significant degradation in global settings (RMSE increases from 0.754 to 1.068). Notably, NIAQUE's performance in the Energy and Efficiency domain exhibits similar patterns to those observed in the House Price Prediction domain. The model maintains reliable uncertainty quantification (coverage near 95%) and demonstrates effective knowledge transfer in domain-specific training, achieving comparable or better performance than local training, showing the ability of the model to effectively leverage additional information from domain-specific datasets to improve on target task.

D.2 Adaptation to New Tasks: Kaggle Competitions

D.2.1 Abalone Competition

To validate NIAQUE's practical effectiveness, we evaluate its performance in recent Kaggle competitions. Abalone Reade & Chow (2024a), focuses on the Abalone age prediction task. This competition, with 2,700 participants and over 20,000 submissions, provides an excellent real-world benchmark, particularly as neural networks are widely reported to underperform in it, compared to traditional boosted tree methods.

Methodology: Our approach involves two stages: pretraining and fine-tuning. First, we pretrain NIAQUE on TabRegSet-101 (which includes the original UCI Abalone dataset Dua & Graff (2019)) using our quantile

Table 6: Performance comparison on HouseRent dataset across point prediction accuracy (SMAPE, AAD, RMSE, BIAS) and distributional accuracy (COV@95, CRPS) metrics. Lower values are better for all metrics except COV@95, where values closer to 95 are optimal. The results with 95% confidence intervals derived from multiple random seed runs.

Model	SMAPE	AAD	RMSE	BIAS	COV@95	CRPS
XGBoost-Local	34.1	0.028	0.066	−0.002	86.5	0.022
XGBoost-Domain	32.9	0.031	0.073	0.004	88.8	0.026
XGBoost-Global	35.4	0.033	0.075	0.011	62.5	0.033
LightGBM-Local	31.4	0.028	0.068	−0.01	92.5	0.021
LightGBM-Domain	33.6	0.029	0.07	−0.01	91.8	0.024
LightGBM-Global	37.1	0.036	0.085	−0.01	94.4	0.028
CatBoost-Local	31.5	0.028	0.068	−0.007	92.8	0.022
CatBoost-Domain	32.4	0.028	0.072	−0.011	94.1	0.022
CatBoost-Global	50.3	0.044	0.104	−0.015	88.4	0.032
Transformer-Local	33.5	0.030	0.070	−0.008	93.6	0.025
Transformer-Domain	32.8	0.029	0.069	−0.007	94.0	0.023
Transformer-Global	32.3	0.028	0.068	−0.007	93.5	0.020
NIAQUE-Local	32.0	0.028	0.067	−0.007	96.0	0.019
NIAQUE-Domain	30.4	0.026	0.063	−0.002	93.5	0.018
NIAQUE-Global	30.3	0.026	0.067	−0.005	93.3	0.019

Table 7: Performance comparison on Kaggle competition dataset. Lower values are better. Baseline results are adopted from publicly shared notebooks and discussion forums in the competition Reade & Chow (2024a). Rank represents the position of the various methods on private leaderboard.

Model	Feature Engineering	RMLSE	Rank
XGBoost (siukeitin)	None	0.15019	1615
LightGBM (dataWr3cker)	None	0.14914	1464
CatBoost Wate (2024)	None	0.14783	1064
TabNet (siukeitin)	None	0.15481	2047
TabDPT	None	0.15026	1623
TabDPT	OpenFE	0.14751	924
TabPFN	None	0.15732	2132
TabPFN	OpenFE	0.14922	1478
NIAQUE-Scratch	None	0.15047	1646
NIAQUE-Pretrain-100	None	0.14823	1232
NIAQUE-Pretrain-full	None	0.14808	1178
NIAQUE-Pretrain-full	OpenFE	0.14556	304
NIAQUE-Ensemble	OpenFE	0.14423	8
Winner Heller (2024)	Manual	0.14374	1

loss framework. Then, we fine-tune the pretrained model on the competition's training data, optimizing for RMSLE metric as defined in Section 3. The dataset size is $90,614$ training samples and $60,410$ test samples Reade & Chow (2024a). To systematically evaluate the impact of transfer learning, we implement three variants: NIAQUE-Scratch (trained only on competition data, no pretraining), NIAQUE-Pretrain-100 (pretrained on TabRegSet-101, excluding Abalone dataset), and NIAQUE-Pretrain-full (pretrained on full TabRegSet-101). Note that the competition dataset is synthetic, generated using a deep learning model trained on the original Abalone dataset, and the competition explicitly encourages the use of the original dataset. Finally, we measure the effect of automated feature engineering on our model using OpenFE Zhang

Table 8: Performance comparison on Flood Prediction competition. Higher R2-scores are better. Baseline results are from competition forums Reade & Chow (2024b). Rank represents the position on private leaderboard.

Model	R2 Score	Rank
XGBoost (siukeitin); Sayed (2024)	0.842	2304
LightGBM (dataWr3cker); Masoudi (2024)	0.766	2557
CatBoost Wate (2024); Milind (2024)	0.845	1700
TabNet (siukeitin)	0.842	2304
TabDPT	0.804	2529
TabPFN	0.431	-
NIAQUE-Scratch	0.865	1099
NIAQUE-Pretrain	0.867	935
Winner (Heller, 2024; Aldparis, 2024)	0.869	1

et al. (2023), an automated feature engineering framework, and explore ensemble strategies leveraging our model's probabilistic nature (NIAQUE-Ensemble).

Results and Analysis: The private leaderboard competition results along with their ranks are presented in Table 7, highlighting NIAQUE's capabilities against a wide range of well-established representative approaches. The effectiveness of transfer learning is evident in the performance progression: NIAQUE-Scratch (RMSLE: 0.15047), trained only on competition data, performs similarly to traditional baselines like XGBoost (0.15019). NIAQUE-Pretrain-100 (0.14823), pretrained on TabRegSet-101 excluding the Abalone dataset, shows significant improvement, demonstrating that knowledge from unrelated regression tasks can enhance performance. NIAQUE-Pretrain-full (0.14808), leveraging the complete TabRegSet-101, further improves performance by incorporating original Abalone information in the training mix and matching CatBoost (0.14783) without any feature engineering. The addition of automated feature engineering (OpenFE) into the competition dataset further improves NIAQUE's performance to 0.14556, significantly outperforming all baseline models and approaching the competition's winning score. Our best result comes from NIAQUE-Ensemble (RMSLE: 0.14423), which leverages the probabilistic nature of our model by combining predictions from different quantiles and model variants (Scratch and Pretrain-full). The ensemble benefits from quantile predictions (in addition to the median) that help correct for data imbalance, predicting higher values where models might underestimate and vice versa, achieving the 8th position on the private leaderboard, remarkably close to the winning score of 0.14374. This result is attained without extensive manual intervention—eschewing hand-engineered features in favor of transfer learning, standard ensemble techniques and automated feature extraction. Moreover, the model's probabilistic formulation is leveraged to further enhance point prediction accuracy, consistent with Bayesian estimation theory, where optimal estimators are typically derived as functions of the posterior distribution (e.g., the variance-optimal estimator is the posterior mean).

These results are particularly significant given that neural networks were generally considered ineffective for this task. Multiple participants reported neural approaches failing to achieve scores better than 0.15, with the competition's winner noting: *"I tried different neural network architectures only to observe that none of them is competitive"* Heller (2024), echoed by other participants: *"yes, I also tried different neural network architectures only to observe that they could not reach beyond .15xx"* Heller (2024). This real-world validation highlights NIAQUE's competitiveness against heavily engineered solutions and underscores the efficacy of integrating transfer learning with probabilistic modeling. Whereas the winning solution relied on an ensemble of 49 models with extensive manual tuning, our approach achieves comparable performance through principled transfer learning and uncertainty quantification, maintaining interpretability and necessitating minimal task-specific modifications.

D.2.2 FloodPrediction Competition

The flood prediction competition Reade & Chow (2024b) presents a challenging regression task aimed at predicting flood event probabilities based on environmental features including MonsoonIntensity, TopographyDrainage, RiverManagement, and Deforestation. With 2,932 participants, this competition addresses a critical real-world problem where labeled data consists of flood probabilities rather than binary outcomes, making regression models more suitable than classification approaches.

Methodology: We adapted NIAQUE for this flood probability prediction task while maintaining its core probabilistic framework. The competition dataset comprises 1.12M training samples and 745K test samples, each containing relevant features derived from Flood Prediction Factors Dhankour (2024). Similar to our Abalone experiment, we evaluate transfer learning effectiveness through two variants: NIAQUE-Scratch (trained directly on competition data) and NIAQUE-Pretrain (pretrained on TabRegSet-101). The model is pretrained using our quantile loss framework and then fine-tuned with MSE Loss on the competition dataset. Notably, we found that automated feature engineering not only failed to improve performance but also significantly increased the feature space dimensionality, making several baseline methods (TabDPT and TabPFN) computationally intractable. Therefore, we focus on comparing the fundamental algorithmic capabilities without feature engineering enhancements.

Results and Analysis: The competition results (Table 8) demonstrate NIAQUE's strong performance in flood prediction. NIAQUE-Scratch achieves an R2-score of 0.865, significantly outperforming both traditional methods like XGBoost (0.842), CatBoost (0.845), and LightGBM (0.766), as well as newer deep learning approaches such as TabNet (0.842), TabDPT (0.804), and TabPFN (0.431). NIAQUE-Pretrain further improves performance to 0.867, approaching the winning score of 0.869, and securing rank 935 on the private leaderboard.

These results are particularly noteworthy given the scale and complexity of the dataset. Unlike the Abalone competition, where feature engineering played a crucial role, this competition highlights NIAQUE's ability to learn effective representations directly from raw features. The improvement from pretraining, though modest in absolute terms (0.002 increase in R2-score), represents a significant advancement in ranking (from 1099 to 935), demonstrating the value of transfer learning even in specialized environmental prediction tasks. The strong performance of our base model without extensive modifications or feature engineering suggests that NIAQUE's probabilistic framework and architecture are well-suited for large-scale regression tasks where the target variable represents underlying probabilities.

The relatively poor performance of some specialized tabular models (particularly TabPFN with R2-score of 0.431) on this large-scale dataset underscores the importance of scalability in real-world applications. NIAQUE maintains its computational efficiency while handling over a million samples, making it practical for deployment in real-world flood prediction systems where both accuracy and computational resources are critical considerations.

D.3 Statistical Significance Analysis

To save space, we present benchmarking results with confidence intervals here. All confidence intervals are obtained by aggregating the evaluation results over 4 runs with different random seeds.

Table 9: Performance comparison across all metrics, with point prediction accuracy (SMAPE, AAD, RMSE, BIAS) and distributional accuracy (COV@95, CRPS). Lower values are better for all metrics except COV@95, where values closer to 95 are optimal. The results with 95% confidence intervals derived from 4 random seed runs.

	SMAPE	AAD	RMSE	BIAS	COV@95	CRPS
XGBoost-global	31.4 ± 4.4	0.574 ± 0.100	1.056 ± 0.143	-0.15 ± 0.05	94.6 ± 0.3	0.636 ± 0.165
XGBoost-local	25.6 ± 0.1	0.433 ± 0.001	0.883 ± 0.004	-0.03 ± 0.01	90.8 ± 0.2	0.334 ± 0.001
LightGBM-global	27.5 ± 0.1	0.475 ± 0.001	0.930 ± 0.003	-0.06 ± 0.01	94.8 ± 0.1	0.426 ± 0.017
LightGBM-local	25.7 ± 0.1	0.427 ± 0.003	0.865 ± 0.012	-0.03 ± 0.01	91.5 ± 0.2	0.327 ± 0.001
CATBOOST-global	31.3 ± 0.2	0.561 ± 0.006	1.030 ± 0.009	-0.12 ± 0.02	94.9 ± 0.2	0.443 ± 0.004
CATBOOST-local	24.3 ± 0.1	0.408 ± 0.001	0.840 ± 0.003	-0.03 ± 0.01	92.7 ± 0.1	0.315 ± 0.001
Transformer-global	23.1 ± 0.3	0.383 ± 0.008	0.806 ± 0.015	-0.01 ± 0.01	94.6 ± 0.3	0.272 ± 0.005
NIAQUE-local	22.8 ± 0.4	0.377 ± 0.012	0.797 ± 0.019	-0.03 ± 0.01	94.9 ± 0.4	0.267 ± 0.011
NIAQUE-global	22.1 ± 0.1	0.367 ± 0.002	0.787 ± 0.005	-0.02 ± 0.01	94.6 ± 0.2	0.261 ± 0.002

Table 10: Ablation study of the XGBoost model.

type	max depth	learning rate	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
global	8	0.02	31.4	0.574	-0.15	1.056	0.636	94.6
global	16	0.02	25.7	0.441	-0.07	0.864	0.484	91.5
global	32	0.02	24.1	0.402	-0.05	0.800	0.353	80.0
global	40	0.02	24.6	0.414	-0.05	0.815	0.378	78.2
global	48	0.02	24.1	0.397	-0.04	0.785	0.362	74.8
global	96	0.02	23.8	0.384	-0.03	0.769	0.346	64.9
local	16	0.02	23.0	0.367	-0.00	0.753	0.317	52.0
local	12	0.02	22.7	0.369	-0.01	0.756	0.304	66.0
local	8	0.02	22.4	0.372	-0.02	0.773	0.294	82.3
local	8	0.05	22.5	0.373	-0.02	0.773	0.291	82.4
local	6	0.02	22.7	0.382	-0.02	0.795	0.298	87.3
local	4	0.02	24.1	0.412	-0.03	0.847	0.318	90.2
local	3	0.02	25.6	0.433	-0.03	0.883	0.334	90.8

E XGBoost Baseline

Table 11: Ablation study of the CATBoost model.

type	depth	min data in leaf	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
global	16	50	31.4	0.565	-0.12	1.036	0.442	94.2
global	16	100	31.3	0.561	-0.12	1.030	0.443	94.9
global	16	200	31.6	0.569	-0.13	1.041	0.445	94.2
global	8	100	41.1	0.785	-0.26	1.324	0.602	94.3
local	3	50	24.3	0.409	-0.03	0.841	0.316	92.7
local	3	100	24.3	0.407	-0.03	0.843	0.317	92.7
local	3	200	24.3	0.408	-0.03	0.840	0.315	92.7
local	5	50	22.2	0.373	-0.02	0.785	0.285	90.7
local	5	100	22.3	0.374	-0.02	0.786	0.285	91.3
local	5	200	22.4	0.378	-0.02	0.791	0.288	91.6
local	7	50	21.5	0.359	-0.02	0.761	0.272	87.2
local	7	100	21.6	0.362	-0.02	0.765	0.273	88.6
local	7	200	21.8	0.366	-0.02	0.772	0.277	89.9

Table 12: CATBoost accuracy as a function of the number of quantiles.

type	depth	min data in leaf	num quantiles	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
global	16	100	3	31.3	0.561	-0.12	1.030	0.443	94.9
global	16	100	5	35.0	0.665	-0.13	1.183	0.482	96.2
global	16	100	7	38.5	0.746	-0.18	1.265	0.533	96.2
global	16	100	9	43.7	0.879	-0.25	1.437	0.622	96.2
global	16	100	51	68.9	1.538	-0.53	2.132	1.036	95.5
local	7	100	3	21.5	0.359	-0.02	0.761	0.272	87.2
local	7	100	9	23.9	0.399	-0.03	0.823	0.284	92.4
local	7	100	51	30.3	0.525	-0.09	1.079	0.369	92.1
local	16	100	51	30.2	0.514	-0.09	1.055	0.362	92.4

F CATBoost Baseline

The CATBoost is trained using the standard package via `pip install catboost` using `grow_policy = Depthwise`. The explored hyper-parqameter grid appears in Table 11.

Table 12 shows CATBoost accuracy as a function of the number of quantiles. Quantiles are generated using linspace grid `np.linspace(0.01, 0.99, num_quantiles)`. We recover the best overall result for the case of 3 quantiles, and increasing the number of quantiles leads to quickly deteriorating metrics. It appears that CATBoost is unfit to solve complex multi-quantile problems.

Table 13: Ablation study of the LightGBM model.

type	max_depth	num leaves	learning rate	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
global	-1	10	0.05	35.6	0.661	-0.17	1.199	0.804	95.2
global	-1	20	0.05	30.9	0.554	-0.11	1.034	0.566	95.4
global	-1	40	0.05	27.5	0.475	-0.06	0.930	0.426	94.8
global	-1	100	0.05	24.6	0.417	-0.03	0.852	0.342	93.3
global	-1	200	0.05	23.4	0.393	-0.02	0.813	0.32	92.3
global	-1	400	0.05	23.6	0.379	-0.02	0.786	0.305	90.9
global	3	10	0.05	50.7	1.084	-0.49	1.763	1.013	94.1
global	3	20	0.05	50.7	1.084	-0.49	1.763	1.013	94.1
global	3	40	0.05	50.7	1.084	-0.49	1.763	1.013	94.1
global	3	100	0.05	50.7	1.084	-0.49	1.763	1.013	94.1
global	3	200	0.05	50.7	1.084	-0.49	1.763	1.013	94.1
global	3	400	0.05	50.7	1.084	-0.49	1.763	1.013	94.1
global	5	10	0.05	39.1	0.768	-0.25	1.341	0.856	94.8
global	5	20	0.05	39.0	0.76	-0.26	1.327	0.863	94.8
global	5	40	0.05	39.0	0.759	-0.26	1.328	0.864	94.8
global	5	100	0.05	39.0	0.759	-0.26	1.328	0.864	94.8
global	5	200	0.05	39.0	0.759	-0.26	1.328	0.864	94.8
global	5	400	0.05	39.0	0.759	-0.26	1.328	0.864	94.8
global	10	10	0.05	35.6	0.661	-0.17	1.199	0.804	95.2
global	10	20	0.05	31.5	0.572	-0.14	1.054	0.59	95.4
global	10	40	0.05	29.8	0.537	-0.13	1.001	0.575	95.2
global	10	100	0.05	29.5	0.528	-0.12	0.991	0.577	95.2
global	10	200	0.05	29.2	0.522	-0.12	0.981	0.576	95.0
global	10	400	0.05	29.1	0.52	-0.12	0.975	0.582	95.1
global	20	10	0.05	35.6	0.661	-0.17	1.199	0.804	95.2
global	20	20	0.05	30.9	0.554	-0.11	1.034	0.566	95.4
global	20	40	0.05	27.1	0.468	-0.07	0.913	0.512	95.2
global	20	100	0.05	25.5	0.435	-0.06	0.864	0.496	94.9
global	20	200	0.05	25.0	0.424	-0.06	0.846	0.488	94.3
global	20	400	0.05	24.3	0.41	-0.05	0.823	0.482	93.6
global	40	10	0.05	35.6	0.661	-0.17	1.199	0.804	95.2
global	40	20	0.05	30.9	0.554	-0.11	1.034	0.566	95.4
global	40	40	0.05	27.8	0.481	-0.05	0.913	0.431	94.7
global	40	100	0.05	24.7	0.419	-0.04	0.848	0.348	93.5
global	40	200	0.05	23.5	0.395	-0.03	0.811	0.332	92.7
global	40	400	0.05	23.2	0.383	-0.03	0.791	0.322	92.0

G LightGBM Baseline

Table 14: Ablation study of the LightGBM model.

type	max_depth	num leaves	learning rate	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
local	-1	5	0.05	23.8	0.399	-0.03	0.823	0.319	90.6
local	-1	10	0.05	22.5	0.376	-0.02	0.786	0.301	88.9
local	-1	20	0.05	21.9	0.364	-0.02	0.766	0.289	86.5
local	-1	50	0.05	21.6	0.355	-0.01	0.752	0.278	82.6
local	2	5	0.05	25.7	0.427	-0.03	0.865	0.327	91.5
local	2	10	0.05	25.7	0.427	-0.03	0.865	0.327	91.5
local	2	20	0.05	25.7	0.427	-0.03	0.865	0.327	91.5
local	2	50	0.05	25.7	0.427	-0.03	0.865	0.327	91.5
local	3	5	0.05	24.3	0.404	-0.03	0.83	0.318	90.7
local	3	10	0.05	23.9	0.396	-0.03	0.818	0.304	90.4
local	3	20	0.05	23.9	0.396	-0.03	0.818	0.304	90.4
local	3	50	0.05	23.9	0.396	-0.03	0.818	0.304	90.4
local	5	5	0.05	23.8	0.399	-0.03	0.823	0.319	90.6
local	5	10	0.05	22.7	0.379	-0.02	0.79	0.3	89.1
local	5	20	0.05	22.3	0.37	-0.02	0.776	0.287	87.6
local	5	50	0.05	22.2	0.368	-0.02	0.773	0.285	87.4

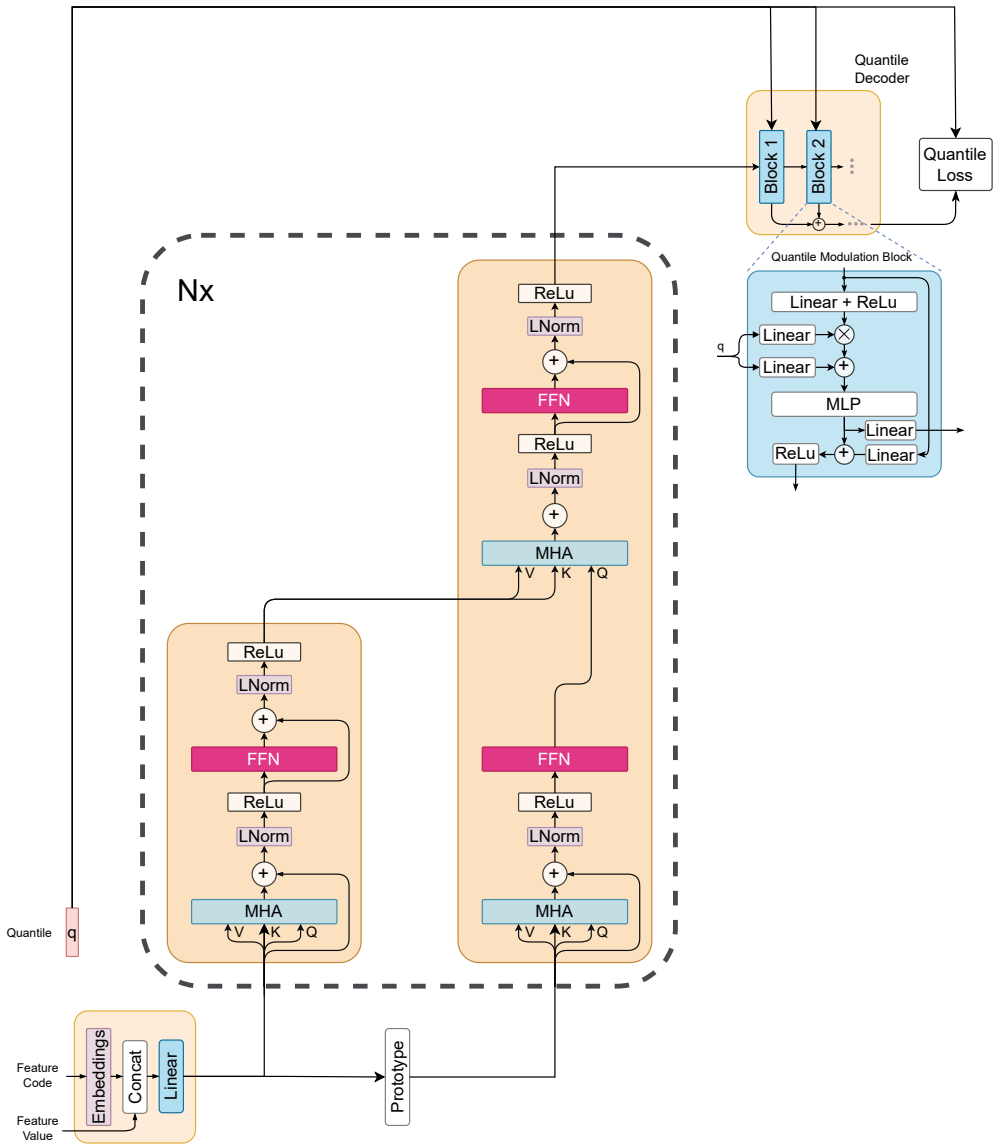


Figure 5: Transformer baseline used in our experiments. The feature encoding module is replaced with transformer block. Feature encoding is implemented via self-attention. The extraction of feature encoding is done by applying cross-attention between the prototype of input features and the output of self-attention. This operation is repeated several times corresponding to the number of blocks in transformer encoder.

H Transformer Baseline

The ablation study of the transformer architecture is presented in Table 15. It shows that in general, increasing the number of transformer blocks improves accuracy, however, at 8-10 blocks we clearly see diminishing returns. Dropout helps to gain better empirical coverage of the 95% confidence interval, but this happens at the expense of point prediction accuracy. Finally, the decoder query that is used to produce the feature embedding that is fed to the quantile decoder can be implemented in two principled ways. First, the scheme depicted in Figure 5, uses the prototype of features supplied to the encoder. We call it the prototype scheme. Second, the prototype can be replaced by a learnable embedding. Comparing the last and third rows in Table 15, we conclude that the prototype scheme is a clear winner.

Table 15: Ablation study of the Transformer architecture.

query	d_model	width	blocks	dp	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
proto	256	256	4	0.1	25.6	0.462	-0.01	0.918	0.313	95.2
proto	256	1024	4	0.1	24.5	0.414	-0.02	0.845	0.292	95.1
proto	256	256	6	0.1	23.7	0.397	-0.01	0.824	0.281	94.9
proto	256	512	6	0.2						
proto	256	1024	6	0.1	24.3	0.407	-0.01	0.840	0.287	94.9
proto	256	1024	6	0.0	26.5	0.477	-0.04	0.980	0.334	93.0
proto	256	512	8	0.0	23.3	0.388	-0.03	0.814	0.276	94.3
proto	256	1024	8	0.0	23.1	0.383	-0.02	0.806	0.272	94.6
proto	256	1024	8	0.1	23.1	0.384	-0.01	0.809	0.272	94.6
proto	256	512	10	0.0	23.0	0.384	-0.03	0.814	0.273	94.2
proto	256	1024	10	0.1	24.3	0.407	-0.01	0.840	0.287	94.9
proto	512	1024	6	0.1						
learn	256	256	6	0.2	35.0	0.722	-0.16	1.406	0.489	93.9

Table 16: Ablation study of NIAQUE-local model.

blocks	width	dp	layers	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
2	64	0.0	3	24.2	0.414	-0.03	0.848	0.292	95.1
2	128	0.0	3	22.8	0.381	-0.02	0.804	0.270	94.5
2	256	0.0	3	22.1	0.365	-0.02	0.786	0.260	94.0
2	512	0.0	3	21.9	0.360	-0.02	0.781	0.257	92.7
2	64	0.1	3	24.7	0.431	-0.07	0.855	0.305	93.3
2	128	0.1	3	23.1	0.389	-0.04	0.81	0.276	94.0
2	256	0.1	3	22.2	0.369	-0.02	0.79	0.263	94.0
2	512	0.1	3	22.0	0.361	-0.02	0.779	0.257	93.5
2	64	0.0	2	24.5	0.419	-0.03	0.852	0.296	95.0
2	128	0.0	2	23.4	0.391	-0.02	0.815	0.276	94.7
2	256	0.0	2	22.3	0.368	-0.02	0.783	0.262	94.1
2	512	0.0	2	22.1	0.363	-0.03	0.780	0.259	92.9
4	64	0.0	2	23.8	0.399	-0.02	0.828	0.282	95.1
4	128	0.0	2	22.8	0.377	-0.03	0.797	0.267	94.9
4	256	0.0	2	22.0	0.363	-0.02	0.788	0.259	93.5
4	512	0.0	2	22.0	0.359	-0.02	0.785	0.257	92.0
4	64	0.1	2	23.8	0.401	-0.03	0.829	0.284	94.3
4	128	0.1	2	22.9	0.379	-0.03	0.801	0.267	94.6
4	256	0.1	2	22.1	0.363	-0.03	0.786	0.259	93.5
4	512	0.1	2	22.0	0.360	-0.03	0.781	0.257	92.4
8	128	0.0	2	23.0	0.381	-0.02	0.798	0.27	95.7

I -Local Baseline

NIAQUE-local baseline is trained on each dataset individually using the same overall training framework as discussed in the main manuscript for the NIAQUE-global, with the following exceptions. The number of training epochs for each dataset is fixed at 1200, the batch size is set to 256, feature dropout is disabled. Finally, for each dataset we select the best model to be evaluated by monitoring the loss on validation set every epoch.

Table 17: Ablation study of NIAQUE model.

blocks	width	dp	layers	singles	log input	sMAPE	AAD	BIAS	RMSE	CRPS	COVERAGE @ 95
1	1024	0.2	2	5%	yes	25.6	0.433	-0.04	0.864	0.306	96.5
2	1024	0.2	2	5%	yes	23.1	0.384	-0.02	0.802	0.272	95.7
2	1024	0.2	3	5%	yes	22.7	0.377	-0.03	0.796	0.267	95.6
4	1024	0.2	2	5%	yes	22.1	0.367	-0.02	0.787	0.261	94.6
4	1024	0.2	3	5%	yes	22.1	0.367	-0.02	0.792	0.262	94.6
8	1024	0.2	2	5%	yes	22.0	0.366	-0.02	0.798	0.264	92.7
4	512	0.2	2	0%	yes	22.5	0.372	-0.02	0.791	0.264	95.4
4	1024	0.2	2	0%	yes	22.1	0.366	-0.02	0.791	0.261	94.2
4	1024	0.3	2	0%	yes	22.1	0.367	-0.02	0.787	0.260	94.7
4	1024	0.4	2	0%	yes	22.2	0.370	-0.02	0.791	0.263	95.1
4	2048	0.3	2	0%	yes	22.1	0.366	-0.02	0.795	0.263	93.4
4	1024	0.2	2	5%	no	31.4	0.530	-0.066	1.017	0.371	95.6

J Training Details and Ablation Studies

To train both NIAQUE and Transformer models we use feature dropout defined as follows. Given dropout probability dp, we toss a coin with probability $\sqrt{\text{dp}}$ to determine if the dropout event is going to happen at all for a given batch. If this happens, we remove each feature from the batch, again with probability $\sqrt{\text{dp}}$. This way each feature has probability dp of being removed from a given batch and there is a probability $\sqrt{\text{dp}}$ that the model will see all features intact in a given batch. The intuition behind this design is that we want to expose the model to all features most of the time, but we also want to create many situations with some feature combinations missing.

Input log transformation defined in eq. (10) is important to ensure the success of the training, as follows both from Table 17 and Figure 6. The introduction of log-transform makes learning curves well-behaved and smooth and translates into much better accuracy.

Adding samples containing only one of the features as input does not significantly affect accuracy. At the same time, the addition of single-feature training rows has very strong effect on the effectiveness of **'s interpretability mechanism. When rows with single feature input are added (Figures 7a and 7b), demonstrates very clear accuracy degradation when top features are removed and insignificant degradation when bottom features are removed. When rows with single feature input are *not* added (Figure 7c), the discrimination between strong and weak features is poor, with removal of top and bottom features having approximately the same effect across datasets.**

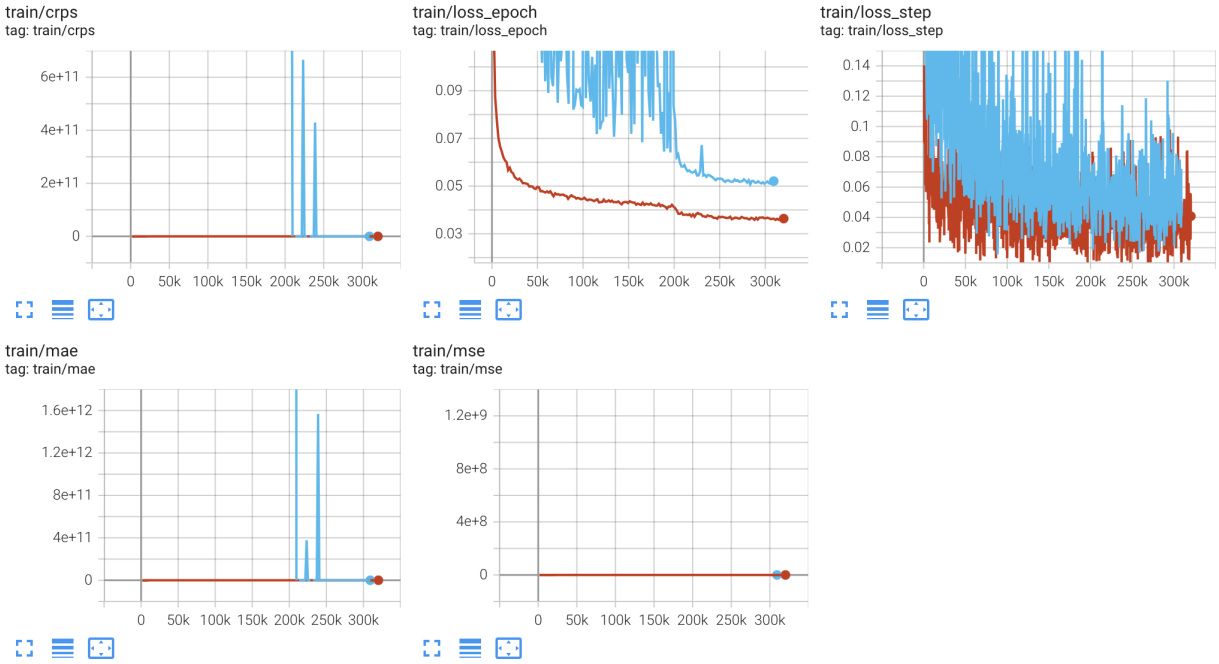
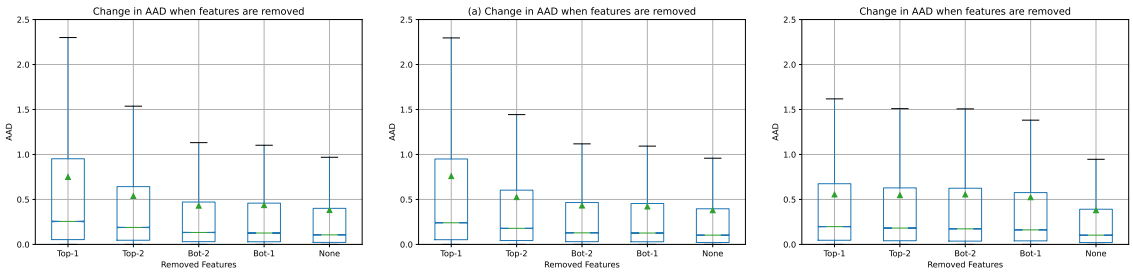


Figure 6: Training losses with (dark red) and without (blue) input value log-transform eq. (10). The introduction of log-transform makes learning curves well-behaved and smooth.



(a) Single features, 10% (b) Single features, 5% (c) Single features, 0%

Figure 7: The effect of adding training rows containing only one of the input features as **input. When rows with single feature input are added (Figures 7a and 7b), demonstrates very clear accuracy degradation when top features are removed and insignificant degradation when bottom features are removed. When rows with single feature input are *not* added (Figure 7c), the discrimination between strong and weak features is poor, with removal of top and bottom features having approximately the same effect across datasets.**