In-context Quantile Regression for Multi-product Inventory Management using Time-series Transformers

Magnus J. Maichle[1*], **Sohom Mukherjee**[1*], **Kai Günder**[1], **Ivane Antonov**[1], **Nikolai Stein**[1], **Richard Pibernik**[1,2]
[1]Julius-Maximilians-Universität Würzburg, [2]Zaragoza Logistics Center
{magnus.maichle, sohom.mukherjee}@uni-wuerzburg.de
[*] *equal contribution*

Abstract

This paper proposes a novel universal quantile regression approach for solving a multi-product inventory management problem, leveraging the in-context learning (ICL) capability of time-series transformers. Our work not only provides a new meta-learning approach for multi-product inventory management, but also extends the state-of-the-art in ICL of transformers by showing how they can be used as universal quantile regressors for data that is not i.i.d. In numerical experiments using a large real-world dataset, our meta-learner consistently outperforms state-of-the-art benchmark models. Remarkably, it outperforms task-specific benchmarks, even when applied to new, unseen inventory management tasks.

1 Introduction

Many companies have to take inventory decisions for thousands of different products in the face of demand uncertainty. Typically, these products have heterogeneous demand patterns that may be correlated, both across products and in time. Conventional methods in Operations Management (OM) for single-product inventory management rely on first fitting a distribution to the historical demand data or Sample Average Approximation (SAA) [4] and then solving a stochastic optimization problem. More recently, various works utilized associated feature information to make inventory prescriptions in an integrated, data-driven manner (e.g. [3, 2, 12]). The single period, single product inventory management problem is referred to as the Newsvendor (NV) problem. The optimal solution to the NV is a conditional quantile that can be estimated using quantile regression. The main limitation of existing approaches is that their extension to multiple products requires a separate quantile regression model for each product. Such task-specific models suffer from a limited availability of task-specific historical (time series) data—especially for new products—and may induce high costs associated with training and tuning. The emergent phenomenon in modern transformer models, called in-context learning, can provide an elegant solution to these challenges. ICL can be considered as an instance of meta-learning (i.e., learning to learn), where a transformer learns a learning algorithm (e.g., linear regression [6, 24]) instead of just learning weights. However, extending ICL to learning complex function classes that can deal with non-i.i.d. data is a challenging task. Few recents works such as [10] look at ICL for non-i.i.d. time series data, but do not consider features. To the best of our knowledge, we are the first to study in-context quantile regression for non-i.i.d. data. We make the following main contributions: (1) We leverage the ICL capability of transformers to build a universal quantile regressor that can act as a meta-learner to prescribe inventory quantities for different products and service levels (target quantile levels), (2) We extend the state-of-the-art in modern transformers by showing that they are capable of learning in-context beyond simple functions accounting for auto-correlation in time-series data, and (3) We provide numerical results on a large real-world dataset to show that our ICL time-series transformers outperforms task-specific models, both for seen and unseen tasks.

NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

2 Problem setup

The objective of the decision-maker in the NV problem is to find an optimal order quantity $q^\star \in \mathcal{Q} \subseteq \mathbb{R}$ for a product with random demand $D \in \mathcal{D} \subseteq \mathbb{R}$. In real-world scenarios, the distribution of the demand may depend on associated contextual feature information $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^k$. The NV problem can be formulated as a stochastic optimization problem of minimizing the expected cost conditioned on a given feature vector

$$q^\star_\alpha = \arg\min_{q \in \mathcal{Q}} \mathbb{E}\left[c(D, q) \mid \mathbf{X} = \mathbf{x}\right], \text{with } c(d, q) := c_u(d - q)^+ + c_o(q - d)^+ \tag{1}$$

In (1), the expectation is with respect to the conditional distribution $\mathbb{P}_{D|\mathbf{X}}$, and $(\cdot)^+$ denotes the $\max\{\cdot, 0\}$ operation, c_u is the underage cost when inventory is not sufficient to fulfill demand, and c_o is overage cost for excess inventory. The optimal order quantity for (1) is $q^\star_\alpha = F^{-1}_{D|\mathbf{X}}(\alpha)$, where $F^{-1}_{D|\mathbf{X}}$ denotes the inverse of the conditional CDF of the random demand D given $\mathbf{X}$, and $\alpha := \frac{c_u}{c_u + c_o} \in (0, 1)$ is called the *service level* [13, 16]. By definition, the optimal order quantity is given by the $\alpha-$conditional quantile function that can be estimated using quantile regression [8, 19]. Let f_θ be a parameterized function that comes from a sufficiently expressive class of functions $\mathcal{F} = \{f_\theta : \mathcal{X} \to \mathcal{Q} \mid \theta \in \Theta \subseteq \mathbb{R}^p\}$, and maps our feature realizations to order quantities. To determine the optimal parameters of f_θ, we want to solve

$$f_{\theta^\star} \in \arg\min_{f_\theta \in \mathcal{F}} \mathbb{E}\left[c_u(D - f_\theta(\mathbf{X}))^+ + c_o(f_\theta(\mathbf{X}) - D)^+\right] \tag{2}$$

$$= \arg\min_{f_\theta \in \mathcal{F}} \mathbb{E}\left[\ell\left(D, f_\theta(\mathbf{X}), \alpha\right)\right], \tag{3}$$

with the pinball loss $\ell(d, \hat{d}, \alpha) := \Big(\hat{d} - d\Big)\Big(\mathbb{I}\{d \leq \hat{d}\} - \alpha\Big)$. The equality above holds because the expression in (3) differs from the expression in (2) only by a constant scaling factor $(c_u + c_o)$. Finally, we replace the expectation over the true conditional by the sample average approximation for the historical data $\{\mathbf{x}_t, d_t\}_{t=1}^T$ to estimate the optimal order quantity $\hat{q}^\star_\alpha = f_{\hat{\theta}^\star}(\mathbf{x})$

$$f_{\hat{\theta}^\star} \in \arg\min_{f_\theta \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^{T} \ell\left(d_t, f_\theta(\mathbf{x}_t), \alpha\right) \tag{4}$$

3 Multi-product Newsvendor using in-context quantile regression

Simultaneous quantile regression. The first step towards building a universal quantile regressor that can act as a meta-learner to prescribe inventory quantities across products and α-quantiles is to train our model f_θ across all possible α-quantiles. This notion has been introduced in literature [18, 5, 7] (albeit, in a different context namely, uncertainty quantification) as simultaneous quantile regression (SQR). The standard quantile regression (4) can be transformed into a simultaneous quantile regression (5), by training the model on all quantile levels $\alpha \sim \mathcal{U}(0, 1)$, and using the class of functions $\mathcal{F} = \{f_\theta : \mathcal{X} \times (0, 1) \to \mathcal{Q} \mid \theta \in \Theta \subseteq \mathbb{R}^p\}$:

$$f_{\hat{\theta}^\star} \in \arg\min_{f_\theta \in \mathcal{F}} \mathbb{E}_{\alpha \sim \mathcal{U}(0,1)}\left[\frac{1}{T} \sum_{t=1}^{T} \ell\left(d_t, f_\theta(\mathbf{x}_t, \alpha), \alpha\right)\right] \tag{5}$$

Note that (5) gives us a solution for a single product across all $\alpha-$quantiles. Therefore, if we use traditional machine learning models such as MLP to solve (5), the multi-product case would necessitate either training separate models for each product, or fine-tuning on new products. We now detail how to leverage ICL capability of transformers to build a universal quantile regressor for multiple products.

In-context quantile regression. ICL refers to the capability of a transformer trained on *prompts* consisting of input-output pairs (corresponding to some *task*), to produce an output given a new prompt (corresponding to some new task), without fine-tuning [6]. In the context of inventory management, the prescription of the order quantities $\{\hat{q}^i_t\}_{t=\tau+1}^T$ for a product $i \in [I]$ can be considered as a single task. The context length τ is a hyperparameter that is fixed during training but may take different values at inference time. At each time-step t, the model receives a prompt that includes a $\tau-$recent

history of features and demands. Unlike existing works on ICL for regression or classification, we cannot assume the demand and feature observations to be i.i.d., because there may be correlations across time as well as products. To account for these dependencies, we utilize the causal self-attention mechanism of transformers and follow a similar approach as [10]. We define the input vector at any time t as $\tilde{\mathbf{x}}_t^i := (\mathbf{x}_t^i, d_{t-1}^i, \alpha_t^i)$ and build a prompt sequence to make a prescription at time t as $P_t^i = \left\{ \left(\tilde{\mathbf{x}}_{t'}^i, d_{t'}^i\right)_{t'=t-\tau}^{t-1}, \left(\tilde{\mathbf{x}}_t^i, 0\right) \right\}$. For training, we randomly sample α_t^i from $\mathcal{U}(0,1)$ for each prompt $P_{t'}^i$ and train the model f_θ to minimize the quantile loss over all prompts

$$f_{\hat{\theta}^\star} \in \underset{f_\theta \in \mathcal{F}}{\arg\min}\, \mathbb{E}_{\alpha_t^i \sim \mathcal{U}(0,1)} \left[\frac{1}{I} \sum_{i=1}^{I} \frac{1}{T-\tau} \sum_{t=\tau+1}^{T} \ell\left(d_t^i, f_\theta(P_t^i), \alpha_t^i\right) \right], \tag{6}$$

where the expectation is with respect to the joint distribution of $\alpha-$service levels over all prompts. The backward pass involves calculating gradients for the quantile loss, which is conditioned on the randomly generated service level. For testing, a prompt has the same structure as defined for training, except that $\alpha \in (0,1)$ is now a predetermined parameter. There is no gradient-based learning during testing, even for new products that were not part of the training data.

Time-series transformer. Our parameterized model f_θ is represented by a transformer architecture. It has been proven theoretically that transformers [22] are sufficiently expressive to universally approximate sequence-to-sequence functions [1, 23], including time-series [20]. Inspired by previous work on time-series transformers such as Lag-Llama [15], we use a decoder-only transformer architecture based on GPT [14] and LLaMA [21]. Mathematically, the transformer will be a function of the form $f_\theta : \mathbb{R}^{(k+3)\times\tau} \to \mathbb{R}$, where the input for any task $i \in [I]$ is a matrix $\tilde{X}^i$ with $(k+3)$ rows (corresponding to k features, the lagged demand and the service level as input, and current demand as output) and τ columns (see Appendix A.1). The transformer processes the input-vectors (columns of $\tilde{X}^i$) for each timestep (from $t-\tau$ to t) as individual tokens. We use a simple linear layer to transform the $(k+3)-$dimensional tokens into the appropriate input dimension of the transformer. We also account for the sequential nature of our time-series data using Rotary Positional Embeddings (RoPE) introduced in [17]. Since we meta-learn a universal quantile regressor across all quantiles, we do not need sophisticated distributional heads used in other time-series transformers [15]. Therefore, our output layer consists of a linear layer that transforms the transformer output into a scalar value, directly representing $\hat{q}_t^i$, conditioned on the α_t^i-quantile.

4 Experiments

Experimental setup. We evaluate our approach using a comprehensive dataset from the Kaggle M5 Uncertainty Challenge, widely referenced in forecasting research [11, 9]. This dataset includes daily demand data for 3,049 products across 10 Walmart stores over 1,941 days. In Inventory Management, a unique combination of store and product is referred to as Stock Keeping Unit (SKU). In our terminology in section 3, each SKU corresponds to a single task, implying that each task involves solving the Newsvendor problem for that specific SKU repeatedly, on a daily basis. For evaluation, we split the dataset into the last 200 days for testing and the prior 100 days for validation. To assess the generalization capability of our model, we randomly draw two samples of 512 tasks (SKUs) that we call ICL-seen and ICL-unseen tasks. The latter are excluded from the training of our ICL models. We evaluate the performance of all models based on the total costs per task, i.e., the sum of the overage and underage costs (see Equation 1) over all 200 test days for the two samples. To compare these costs across service levels and tasks, we report the per-task improvement relative to an SAA model that computes the optimal demand quantiles without features. We denote this as relative improvement in Figure 1. We call our model ICL-SQR-Transformer (ICL-SQR-TR) as it learns across tasks and α quantiles. For benchmarking, we report results for an ICL-Transformer (ICL-TR) trained on a specific quantile, a Multi Layer Perceptron (MLP) trained similarly to our Transformer model (ICL + SQR), and to three single-task models (MLP, Extreme Gradient Boosting (XGB), and Transformer (TR)).

Results. Figure 1 compares the relative improvement over SAA between our ICL-SQR-TR model and the benchmarks, evaluated at a 70% service level based on the 512 unseen tasks (see Appendix A.4, A.5, A.6 for significance tests and additional analyses). Our model leads to substantial and statistically significant performance improvements (at a 1% significance level, based on both Wilcoxon signed

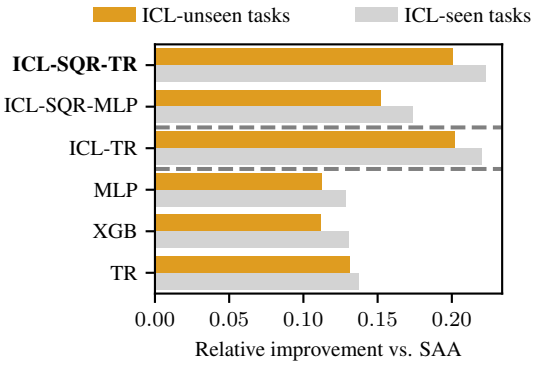

Figure 1: Average cost improvement of our ICL-SQR-TR model and benchmark models relative to SAA, evaluated at 70% service level. Averaged over total costs across the 512 unseen test tasks and test time steps.

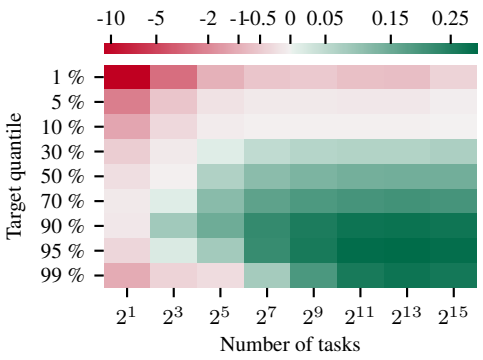

Figure 2: Average cost improvement of our ICL-SQR-TR model varied by the number of training tasks and service level, relative to SAA. Performance based on the 512 unseen tasks, averaged across multiple independent runs.

rank tests and Man-Whitney-U tests), compared to state-of-the-art benchmark models. Within the class of ICL models, there is no significant difference between our ICL-SQR-TR model and the ICL-only Transformer. Thus, our much more flexible and scalable model is superior to all available models specialized on single tasks, and it is on par with a model specialized on a single α-service level. Most importantly, these results point towards the strong generalization capability of our ICL-SQR-TR model, our model outperforms all other models—regardless of having seen or not seen the specific tasks during training.

Impact of the number of training tasks. To understand how the ICL capability depends on the number of training tasks, we conduct an additional experiment in which we successively increase the number of training tasks on a power scale from 2 to 30,000 and train various instances of our ICL-SQR-TR model. We evaluate the performance of each of these models on the 512 unseen tasks, for α-service levels ranging from 1% to 99%. To ensure the robustness of our results, we draw multiple samples of training datasets that decrease in the number of tasks according to a power scale, i.e, for 2 tasks we draw 128 samples, for 8 tasks, we draw 64 samples, etc. (details in Appendix A.2). We use hyperparameter optimization to adjust the size of the ICL-SQR-TR model and the number of training epochs for the different numbers of training tasks (see Appendix A.3). The results are shown in Figure 2. Predictably, performance improves with more training tasks, especially for extreme service levels (both high and low). While medium-sized models already perform well at moderate service levels, larger models are required for satisfactory results at extreme levels. There is a visible performance asymmetry around the 50% service level due to the highly skewed demand distributions, a common characteristic of real-world data that often features lumpy and/or intermittent demands.

5 Conclusion

Scientific considerations. Our main methodological contribution is the development of a meta-learner that combines SQR with ICL capabilities of transformers to build a universal quantile regressor that solves a practically relevant multi-variate and non-i.i.d time-series problem. Our approach can be applied to any decision-making problem that can be translated into a quantile regression problem.

Practical considerations. We propose a flexible and scalable prescriptive approach for taking inventory decisions for multiple products (tasks) with different cost characteristics (target quantiles). Numerical experiments show that our model substantially outperforms state-of-the-art models when applied to existing products, and, more importantly, to new products with limited historical data.

Limitations and outlook. In this work, we focused on a widely used (static) inventory management problem. To apply it to a broader range of real-world problems, one could extend this work to dynamic decision-making problems (where the decision in one period also affects the next period, e.g., utilizing dynamic optimization or reinforcement learning) or incorporate cross-company data.

References

[1] Silas Alberti, Niclas Dern, Laura Thesing, and Gitta Kutyniok. Sumformer: Universal approximation for efficient transformers. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 72–86. PMLR, 2023.

[2] Gah-Yi Ban and Cynthia Rudin. The Big Data Newsvendor: Practical Insights from Machine Learning. *Operations Research*, 67(1):90–108, January 2019.

[3] Dimitris Bertsimas and Nathan Kallus. From Predictive to Prescriptive Analytics. *Management Science*, 66(3):1025–1044, March 2020.

[4] Simone Buttler, Andreas Philippi, Nikolai Stein, and Richard Pibernik. A meta analysis of data-driven newsvendor approaches. In *ICLR 2022 Workshop on Setting up ML Evaluation Standards to Accelerate Progress*, 2022.

[5] Youngseog Chung, Willie Neiswanger, Ian Char, and Jeff Schneider. Beyond pinball loss: Quantile methods for calibrated uncertainty quantification. *Advances in Neural Information Processing Systems*, 34:10971–10984, 2021.

[6] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[7] Adèle Gouttes, Kashif Rasul, Mateusz Koren, Johannes Stephan, and Tofigh Naghibi. Probabilistic time series forecasting with implicit quantile networks. *arXiv preprint arXiv:2107.03743*, 2021.

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

[9] A. David Lainder and Russell D. Wolfinger. Forecasting with gradient boosted trees: augmentation, tuning, and cross-validation strategies. *International Journal of Forecasting*, 38(4):1426–1433, October 2022.

[10] Jiecheng Lu, Yan Sun, and Shihao Yang. In-context time series predictor. *arXiv preprint arXiv:2405.14982*, 2024.

[11] Spyros Makridakis, Evangelos Spiliotis, Vassilios Assimakopoulos, Zhi Chen, Anil Gaba, Ilia Tsetlin, and Robert L. Winkler. The M5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting*, 38(4):1365–1385, October 2022.

[12] Afshin Oroojlooyjadid, MohammadReza Nazari, Lawrence V. Snyder, and Martin Takáč. A Deep Q-Network for the Beer Game: Deep Reinforcement Learning for Inventory Optimization. *Manufacturing & Service Operations Management*, 24(1):285–304, January 2022.

[13] Meng Qi and Zuo-Jun Shen. Integrating prediction/estimation and optimization with applications in operations management. In *Tutorials in operations research: emerging and impactful topics in operations*, pages 36–58. INFORMS, 2022.

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[15] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

[16] Lawrence V Snyder and Zuo-Jun Max Shen. *Fundamentals of supply chain theory*. John Wiley & Sons, 2019.

[17] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[18] Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. *Advances in neural information processing systems*, 32, 2019.

[19] Ichiro Takeuchi, Quoc V Le, Timothy D Sears, Alexander J Smola, and Chris Williams. Nonparametric quantile estimation. *Journal of machine learning research*, 7(7), 2006.

[20] Josef Teichmann, Christa Cuchiero, Matteo Gambara, Florian Krach, and Hanna Wutte. Eth zürich machine learning in finance 2024 lecture 10. `https://people.math.ethz.ch/~jteichma/index.php?content=teach_mlf2024`, 2024.

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[22] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[23] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.

[24] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023.

A Appendix

A.1 Prompt details for ICL SQR Transformer

Here in (7) we provide the detailed matrix structure of a prompt $\tilde{P}_t^i$ that was defined in section 3. The first row of the matrix represents features, the second row represents lagged demands and the third row service levels for each prompt. The first three rows together which have been highlighted in blue represents the input of the prompt. The last row which has been highlighted in red represents the output of the prompt. The last column of the matrix represents the query to which we wish to find out the output using the transformer. Since one do not know the output to this query, we provide a 0 in the corresponding position of the prompt, to match the token size of the transformer.

$$\tilde{P}_t^i = \begin{bmatrix} x_{t-\tau}^i & x_{t-\tau+1}^i & \dots & x_t^i \\ d_{t-\tau-1}^i & d_{t-\tau}^i & \dots & d_{t-1}^i \\ \alpha_i^t & \alpha_i^t & \dots & \alpha_i^t \\ d_{t-\tau}^i & d_{t-\tau+1}^i & \dots & 0 \end{bmatrix} \tag{7}$$

A.2 Number of independent runs per number of training tasks

Number of training tasks	2^1	2^3	2^5	2^7	2^9	2^{11}	2^{13}	2^{15}
Number of runs	128	64	32	16	8	4	2	1

Table 1: Number of independent runs per number of training tasks

A.3 Hyperparameters for ICL-SQR-TR models by number of training tasks

Number of training tasks	$\mathbf{2^1}$	2^3	2^5	$\mathbf{2^7}$	2^9	$\mathbf{2^{11}}$	2^{13}	2^{15}
Model parameters								
Number of layers	**2**	3	4	**5**	5	**5**	5	6
Number of heads	**8**	8	8	**8**	10	**12**	12	12
Size of embedding per head	**16**	16	32	**32**	32	**32**	48	48
Training parameters								
Training epochs	**500**	400	300	**200**	100	**50**	25	10
Early stopping patience	**50**	50	30	**20**	15	**10**	5	3
Learning rate	$\mathbf{3 \times 10^{-2}}$	1×10^{-2}	1×10^{-3}	$\mathbf{7 \times 10^{-4}}$	7×10^{-4}	$\mathbf{1 \times 10^{-3}}$	1×10^{-3}	1×10^{-3}
Learning rate scheduler warmup	**500**	1000	2000	**3000**	3000	**2000**	2000	2000
Regularization								
Drop-out probability	**0.35**	0.3	0.3	**0.25**	0.25	**0.2**	0.15	0.1
L-2 regularization	$\mathbf{3 \times 10^{-3}}$	1×10^{-3}	3×10^{-4}	$\mathbf{3 \times 10^{-5}}$	1×10^{-5}	$\mathbf{3 \times 10^{-6}}$	1×10^{-6}	1×10^{-6}

Table 2: Hyperparemters used for the ICL-SQR-TR models traind on varying number of training tasks. Due to high computational cost, a specific tuning was only conducted for some dataset sizes (columns in bold). For the remaining sizes, hyperparameters were interpolated/extrapolated.

A.4 Significance tests

	Service level	**ICL-SQR-MLP**	**ICL-TR**	**MLP**	**XGB**	**TR**	**SAA**
Wilcoxon signed-rank	1%	0.0000^{***}		0.9846	0.0965^{*}	0.0000^{***}	1.0000
	5%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.9941
	10%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.3005
	30%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	50%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	70%	0.0000^{***}	0.9643	0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	90%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	95%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	99%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
Man-Whitney-U	1%	0.0000^{***}		0.7958	0.1410	0.0000^{***}	0.9998
	5%	0.0000^{***}		0.0000^{***}	0.0003^{***}	0.0000^{***}	0.7502
	10%	0.0000^{***}		0.0000^{***}	0.0006^{***}	0.0000^{***}	0.5397
	30%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	50%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	70%	0.0000^{***}	0.5586	0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	90%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	95%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}
	99%	0.0000^{***}		0.0000^{***}	0.0000^{***}	0.0000^{***}	0.0000^{***}

Significance codes: $p-$ value $< 0.01 : {}^{***}$, $p-$ value $< 0.05 : {}^{**}$, $p-$ value $< 0.1 : {}^{*}$, for a single-sided test

Table 3: Statistical differences between our ICL-SQR-TR model and all benchmarks across all service levels based on the Wilcoxon signed-rank test and Mann–Whitney U test. Calculated based on the ICL-unseen tasks. The numbers represent the p-values, highlighted with significance codes. The service-level specific ICL transformer has only been trained on the 70% quantile.

A.5 Detailed results for 70% service level

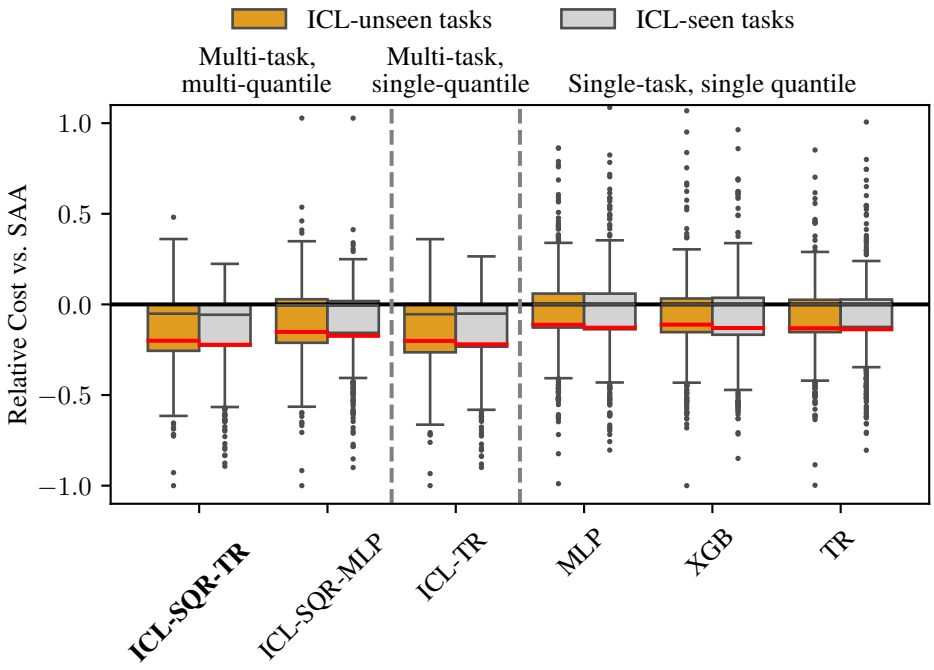

Figure 3: Relative cost of our ICL-SQR-TR model and benchmark models relative to SAA, evaluated at 70% service level. The boxplots represent the relative distribution of total cost on each of the 512 tasks in the test set. The red lines represent mean-performance.

A.6 Results for additional relevant service levels

A.6.1 Service level of 30%

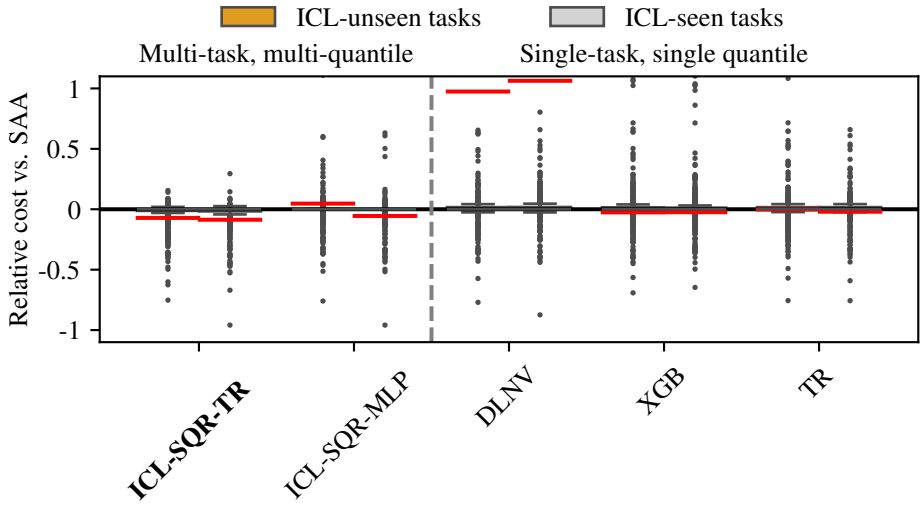

Figure 4: Relative cost of our ICL-SQR-TR model and benchmark models relative to SAA, evaluated at 30% service level. The boxplots represent the relative distribution of total cost on each of the 512 tasks in the test set. The red lines represent mean-performance.

A.6.2 Service level of 50%

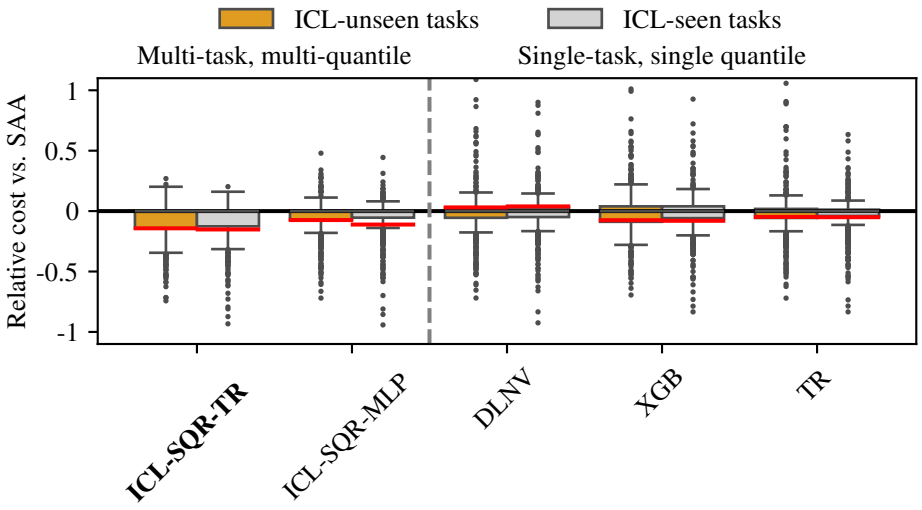

Figure 5: Relative cost of our ICL-SQR-TR model and benchmark models relative to SAA, evaluated at 50% service level. The boxplots represent the relative distribution of total cost on each of the 512 tasks in the test set. The red lines represent mean-performance.

A.6.3 Service level of 90%

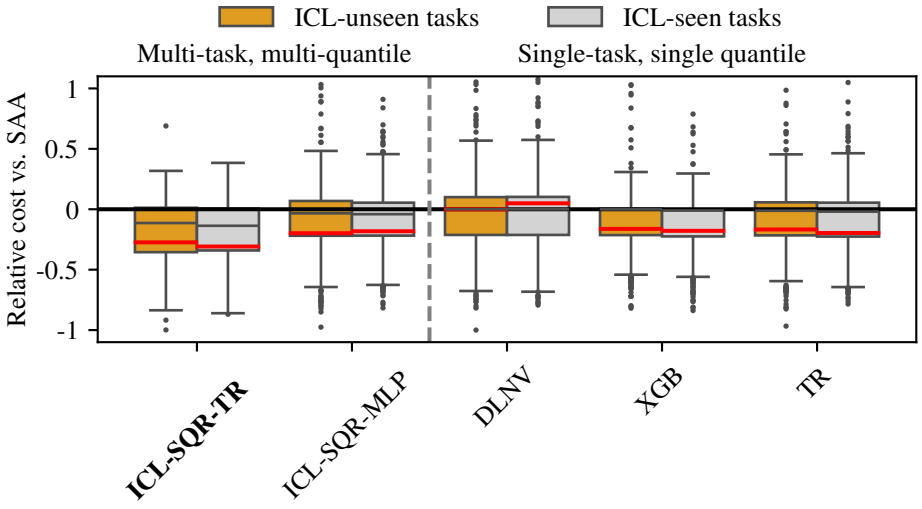

Figure 6: Relative cost of our ICL-SQR-TR model and benchmark models relative to SAA, evaluated at 90% service level. The boxplots represent the relative distribution of total cost on each of the 512 tasks in the test set. The red lines represent mean-performance.

A.6.4 Service level of 95%

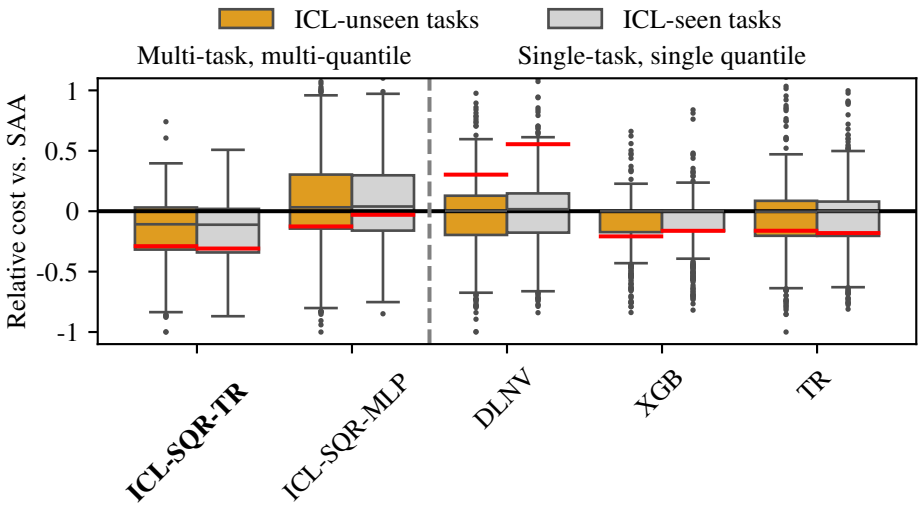

Figure 7: Relative cost of our ICL-SQR-TR model and benchmark models relative to SAA, evaluated at 95% service level. The boxplots represent the relative distribution of total cost on each of the 512 tasks in the test set. The red lines represent mean-performance.

A.6.5 Service level of 99%

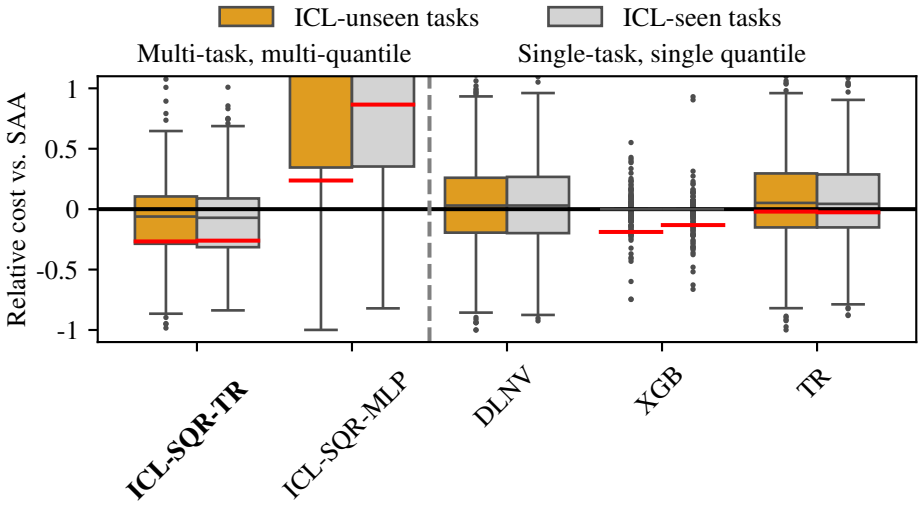

Figure 8: Relative cost of our ICL-SQR-TR model and benchmark models relative to SAA, evaluated at 99% service level. The boxplots represent the relative distribution of total cost on each of the 512 tasks in the test set. The red lines represent mean-performance.