REVISITING TABULAR DEEP REGRESSION: OBJECTIVES AND THE LIMITS OF GLOBAL EARLY STOPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

While deep learning models for tabular data have recently received significant research attention, the choice of objective functions for regression tasks remains an understudied degree of freedom. Several lines of research offer alternatives to the standard Mean Squared Error (MSE), but differences in datasets, backbones, and evaluation protocols make their practical benefits difficult to compare. In this work, we address this gap by performing a systematic re-evaluation of regression objectives across a broad collection of established tabular datasets and multiple neural backbones. Our comparison reveals that many proposed objectives provide limited or inconsistent gains over a carefully tuned MSE baseline, while quantile loss yields consistent aggregate improvements over MSE across the tested backbones. We then identify a limitation that remains even with this strong objective: regions with different levels of target noise can reach their best predictive performance at different training epochs. Consequently, a single global early-stopping checkpoint can impose a compromise between uncertainty regions, even when the objective can represent heteroskedastic target distributions. Motivated by this limitation, we introduce a heteroskedasticity-aware extension of quantile loss that uses input-dependent label smoothing as a potential remedy and improves over quantile loss on several backbones. Overall, our results provide a modern empirical picture of regression objectives for tabular deep learning and highlight uncertainty-dependent stopping phenomenon as an important consideration in the design of tabular methods.

1 INTRODUCTION

Despite the growing adoption of deep learning models for tabular data, in practice, it is not entirely clear what objective function should be used to solve regression tasks with DNNs. While the design of advanced regression objectives is an active research direction (Imani & White, 2018; Stewart et al., 2025; Farebrother et al., 2024; Song & Bahri, 2025; Zhu et al., 2025; Zhang et al., 2023; Zha et al., 2023; Zhang et al., 2025), the traditional Mean Squared Error (MSE) remains widely used. Existing studies evaluate alternative objectives on different datasets, backbones, and tuning protocols, making it difficult to establish which alternatives provide reliable improvements in practice.

We revisit traditional and modern regression objectives on 41 established tabular datasets with extensive hyperparameter tuning, focusing on conventional DNNs (pretrained PFNs are outside our scope). We find that most objectives do not consistently improve over a carefully tuned MSE baseline. Two approaches stand out on MLP: Gaussian-smoothed binning objectives (Imani & White, 2018; Stewart et al., 2025), which represent targets as distributions over discrete bins, and quantile loss (Koenker & Bassett, 1978), which predicts conditional quantiles of $p(y \mid x)$ via the pinball objective. Of these, quantile loss yields more consistent gains across backbones. Related evidence comes from the PFN literature: TabICLv2 (Qu et al., 2026) adopts quantile loss for regression and reports benefits over MSE and binning; our findings complement this with a systematic comparison in conventional tabular models.

Having identified quantile loss as a strong regression objective, we investigate a limitation that remains under standard training: regions with different levels of data uncertainty can reach their best

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

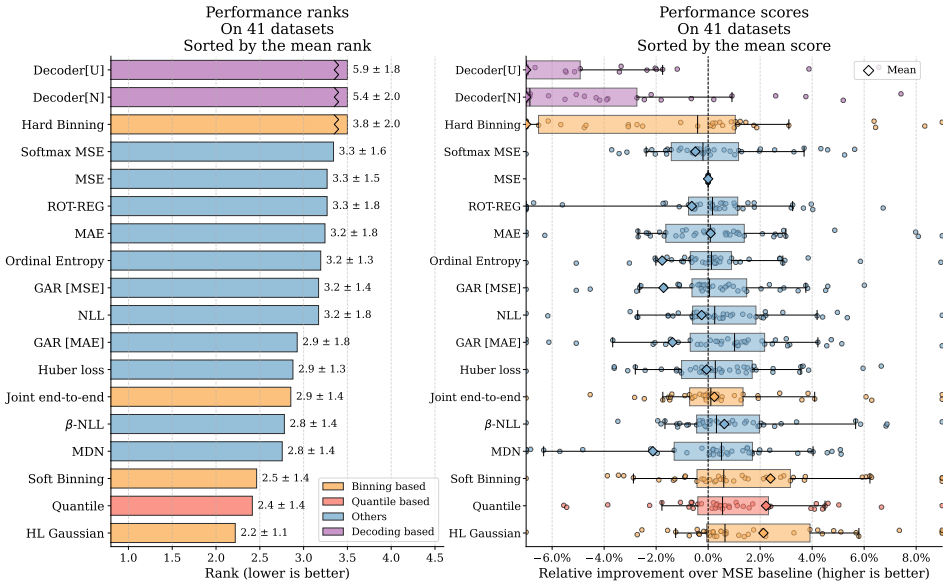


Figure 1: Performance comparison of different objective functions using an MLP backbone across 41 BeyondArena datasets. (*Left*) Average rank performance across all datasets. (*Right*) Relative RMSE improvement over the standard MSE baseline, visualizing the magnitude of performance gains or degradation. The box plots visualize the 10th, 25th, 50th, 75th, and 90th percentiles of the jitter plots. Notation details are provided in the beginning of Section 3.2.2.

predictive performance at different epochs. Consequently, a single early-stopping checkpoint can impose a compromise between uncertainty regions. This effect is particularly relevant when target noise varies across the feature space, as is common in tabular data (Kartashev et al., 2025). We examine this conflict through controlled synthetic experiments and uncertainty-based checkpoint selection on real datasets.

Motivated by this limitation, we propose *Adaptive Quantile Regression* (AdaQR), which incorporates jointly learned, input-dependent smoothing into quantile training. Effectively, by reducing the sensitivity of quantile updates to individual label realizations in high-uncertainty regions, adaptive smoothing aims to delay overfitting there while allowing learning to continue in lower-uncertainty regions. AdaQR is designed as one potential remedy for the mismatch induced by a shared stopping point, which improves upon standard quantile loss on several DNN backbones as we show in our experiments.

Our main contributions are:

- We benchmark 18 regression objectives on an MLP across 41 datasets and evaluate selected objectives on three additional tabular DNN backbones, using up to 100 hyperparameter tuning trials and the Muon optimizer (Jordan et al., 2024; Gorishniy et al., 2026b).
- We show that most objectives do not consistently beat tuned MSE. Gaussian-smoothed binning objectives and quantile loss stand out on MLPs, while quantile loss provides more consistent gains across the tested backbones.
- We characterize uncertainty-dependent stopping behavior in tabular regression: controlled synthetic experiments reveal different preferred stopping epochs across noise levels, while real-data experiments show that estimated uncertainty informs checkpoint selection.
- We propose a new AdaQR objective that provides a jointly learned smoothing mechanism for investigating how uncertainty-aware training can mitigate stopping conflicts and evaluate this objective on established tabular backbones.

2 Related Work

Regression objectives. Alternatives to MSE include robust losses (Huber, 1964), target discretization (Imani & White, 2018; Stewart et al., 2025; Farebrother et al., 2024), conditional distribution modeling (Bishop, 1994; Koenker & Bassett, 1978; Song & Bahri, 2025; Han et al., 2022), and representation regularization (Zha et al., 2023; Zhang et al., 2025; 2023). We compare representative methods under a common tabular evaluation protocol and describe the individual objectives in the following sections.

Early stopping and overfitting. Different parts of a model or subsets of the data can reach their best generalization performance at different training epochs. Heckel & Yilmaz (2020) analyze different bias–variance tradeoffs across feature directions and network layers, showing that adjusting learning rates can improve early-stopping performance. Bao et al. (2023) observe that some token groups are already overfitting while others remain underfit at the selected checkpoint in sequence-to-sequence models. Berta et al. (2025) show that calibration and refinement errors are minimized at different stages of training, including in tabular classification. We extend the uncertainty-based analysis of tabular models (Kartashev et al., 2025) to training trajectories and checkpoint selection, examining how input-dependent noise relates to the epochs at which different regions achieve their best predictive performance.

Heteroskedasticity-aware objectives and adaptive smoothing. A large body of work models input-dependent uncertainty in regression (Kendall & Gal, 2017; Immer et al., 2023). However, joint mean and variance estimation can introduce optimization difficulties (Seitzer et al., 2022; Wong-Toi et al., 2023), motivating methods that preserve the mean-prediction accuracy of a corresponding MSE-trained model (Stirn et al., 2023). Related work on quantile regression investigates smoothing and local scale adaptation. Tan et al. (2022) study convolution smoothing of the pinball loss to improve optimization and statistical estimation. Fasiolo et al. (2021) derive covariate-dependent smoothing scales from a preliminary location–scale fit, while Jung et al. (2015) estimate local scales using preliminary quantile or absolute-residual regressions. In distributional reinforcement learning, Malekzadeh et al. (2024) propose a noise-motivated generalization of quantile-Huber loss with adaptive parameter adjustment. AdaQR combines an energy-distance objective with an input-dependent smoothing head learned jointly with the neural quantile predictor. We investigate this formulation through its effects on uncertainty-dependent training dynamics and tabular point prediction.

3 Revisiting regression objectives

Notation. We consider supervised regression on tabular data. Let $D = \{(x_i, y_i)\}_{i=1}^{n}$ denote a dataset, where $x_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \mathbb{R}$ is the target. We denote the neural model by $f_\theta : \mathbb{R}^d \to \mathbb{R}^{d'}$, where the output dimension d' depends on the objective and need not equal one. For pointwise objectives, the loss on an observation is $l(f_\theta(x_i), y_i)$.

3.1 Losses

We compare the following regression objectives.

Traditional losses. MSE penalizes squared errors, MAE penalizes absolute errors, and Huber loss (Huber, 1992) combines quadratic penalties for small errors with linear penalties for large ones.

Binning-based losses. These methods discretize the target range into bins and train a categorical prediction head to match a target distribution. Hard Binning assigns each target to its nearest bin, while Soft Binning distributes probability mass according to Gaussian weights evaluated at bin centers (Stewart et al., 2025). HL-Gaussian instead integrates a truncated Gaussian density over each bin (Imani & White, 2018). Joint end-to-end encoding learns the target distribution using an auxiliary encoder and a reconstruction objective (Stewart et al., 2025). All four methods return the predicted expectation over bin centers. Softmax-MSE uses the same prediction head but directly minimizes the squared error of this expectation (Stewart et al., 2025).

Quantile regression. The model predicts a grid of conditional quantiles using the pinball loss (Koenker & Bassett, 1978); their average serves as the scalar prediction.

Table 1: Objective and optimizer comparison on MLP across 41 datasets. Gain is the median relative RMSE improvement over the common MSE+AdamW baseline; positive values indicate improvement. W/T/L counts wins, ties, and losses under the standard-deviation-based comparison rule in Appendix H.2. Bold denotes the largest median gain within each optimizer.

	AdamW		Muon	
Objective	**Gain (%)**	**W / T / L**	**Gain (%)**	**W / T / L**
MSE	—	—	+1.50	18 / 19 / 4
HL-Gaussian	**+0.78**	19 / 19 / 3	**+2.34**	26 / 11 / 4
quantile loss	+0.67	15 / 23 / 3	+2.18	24 / 13 / 4

Likelihood-based objectives. Gaussian NLL jointly learns a conditional mean and variance. β-NLL reweights this objective by a detached power of the predicted variance to address optimization difficulties (Seitzer et al., 2022). Mixture Density Networks model the conditional distribution as a Gaussian mixture (Bishop, 1994). The autoregressive approach of Song & Bahri (2025) instead predicts tokenized targets using next-token cross-entropy. We evaluate both its normalized and unnormalized target encodings.

Additional objectives and regularizers. GAR combines pointwise and pairwise terms to capture target values and their differences (Zhu et al., 2025). Ordinal Entropy (Zhang et al., 2023) and ROT-REG (Zhang et al., 2025) augment regression training with representation regularization, using entropy-based and optimal-transport-based terms, respectively.

We provide a detailed description of all objectives in Appendix A.1.

3.2 Experiments

Datasets. Our benchmark consists of 41 regression datasets from BeyondArena (Purucker et al., 2026). Further details are provided in Appendix G.

3.2.1 Implementation details

Model backbone. We first compare regression objectives on the simplest MLP backbone, then evaluate selected objectives on additional architectures. Depending on the objective, the prediction head outputs either a scalar, $\hat{y}_i = f_\theta(x_i) \in \mathbb{R}$, or a vector parameterizing the corresponding predictive output.

Experimental setup. We follow the evaluation protocol of Gorishniy et al. (2025), detailed in Appendix H.1. Unless stated otherwise, we use Muon-based training (Jordan et al., 2024), following recent evidence of its effectiveness on tabular data (Gorishniy et al., 2026b). For each dataset and objective, we tune the model, objective, and optimizer hyperparameters on the validation set. We then retrain the selected configuration from scratch across 15 random seeds and report test RMSE. Method comparisons use the mean and standard deviation across these seeds, as described in Appendix H.2.

3.2.2 Results

We evaluate all methods following the protocol above and report the results in Figure 1. For readability, Decoder[U] and Decoder[N] denote the unnormalized and normalized decoders from Song & Bahri (2025), respectively; GAR[MSE] and GAR[MAE] denote GAR variants with MSE- and MAE-based formulations (Zhu et al., 2025), respectively.

Key takeaways.

1. Most objectives do not provide consistent improvements over a carefully tuned MSE.
2. Soft Binning (Stewart et al., 2025), HL-Gaussian (Imani & White, 2018), and quantile loss (Koenker & Bassett, 1978) stand out in the aggregate results, improving upon MSE in terms of both ranks and relative RMSE improvement.

Table 2: Quantile loss yields consistent aggregate improvements across the tested backbones. Each objective is compared against MSE on the same backbone. Gain is the median relative RMSE improvement; positive values indicate improvement. W/T/L follows the comparison rule in Appendix H.2. Bold denotes the larger median gain within each backbone, with both values highlighted when equal at the reported precision.

	HL-Gaussian		quantile loss	
Backbone	**Gain (%)**	**W / T / L**	**Gain (%)**	**W / T / L**
MLP	**+0.65**	19 / 17 / 5	+0.55	21 / 15 / 5
TabPack	+0.06	8 / 20 / 13	**+0.46**	18 / 21 / 2
RealMLP	+0.12	13 / 17 / 11	**+0.40**	20 / 16 / 5
TabM	+0.54	17 / 19 / 5	**+0.58**	20 / 16 / 5

These findings also hold under AdamW (Loshchilov & Hutter, 2019). Table 1 shows that HL-Gaussian and quantile loss improve over MSE in aggregate with this optimizer. The Muon configurations achieve larger median gains over the common MSE+AdamW reference, highlighting the importance of optimizer choice when evaluating regression objectives.

3.2.3 Advanced backbones

Having identified promising objectives on MLP, we examine whether their benefits transfer to other state-of-the-art backbone architectures. We retain HL-Gaussian as a representative binning-based objective and compare it and quantile loss against MSE.

Following the same evaluation protocol, we train TabPack with periodic feature embeddings (Gorishniy et al., 2026a), TabM-mini with PLE-Q embeddings (Gorishniy et al., 2025), and RealMLP (Holzmüller et al., 2024). Implementation details are provided in Appendix H.4.

Table 2 shows that quantile loss provides more consistent gains across the tested backbones than HL-Gaussian. On TabPack, HL-Gaussian has more losses than wins against MSE, despite a small positive median improvement; on RealMLP, its median gain is also limited. In contrast, quantile loss achieves positive median improvements and more wins than losses on every tested backbone. These results support quantile loss as a practical alternative to MSE, while showing that its benefits remain dataset-dependent.

4 Quantile regression under heteroskedasticity

The benchmark in Section 3.2 identifies several regression objectives that consistently outperform the MSE baseline. While HL-Gaussian and quantile regression (QR) perform comparably on a plain MLP, QR is more robust across backbones (Table 2). Therefore, we focus on QR for the remainder of our analysis.

QR predicts conditional quantiles without assuming a parametric family for $p(y \mid x)$, allowing the predicted distribution's shape and scale to vary across inputs. This flexibility is particularly beneficial for tabular data, where the conditional variance $\mathrm{Var}[y \mid x]$, often termed *data uncertainty*, changes substantially with x (i.e., the data is *heteroskedastic*). Kartashev et al. (2025) identify regions of low and high data uncertainty in several established tabular datasets, and demonstrate that dataset-wise aggregate scores often obscure how methods perform across these different regions.

However, modeling input-dependent distributions does not guarantee that regions with different noise levels achieve their optimal predictive performance at the same epoch. Standard global early stopping selects a single checkpoint based on average validation error, applying it uniformly to all inputs. We first examine how prediction errors evolve during training across different uncertainty levels (Section 4.1). We then test whether selecting checkpoints separately for different uncertainty cohorts improves upon doing so for random cohorts of the same size (Section 4.2). Together, these experiments reveal an optimization challenge that distributional flexibility alone cannot resolve, motivating the adaptive objective introduced in Section 5.

Both experiments in this section require a per-point estimate of data uncertainty and are most interpretable when this estimate is exact and test targets are noise-free. To demonstrate the claims

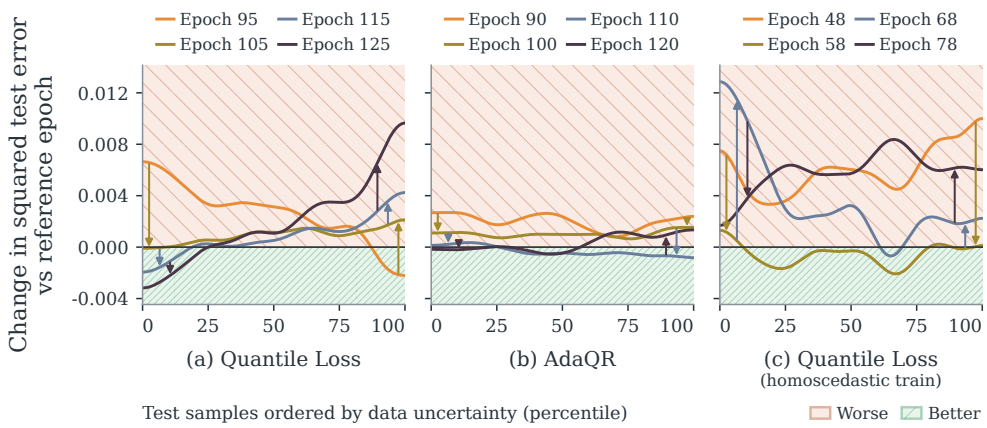


Figure 2: *Heteroskedastic noise creates a mismatch between the stopping epochs preferred by low- and high-uncertainty regions, while AdaQR mitigates this effect.* Uncertainty plots (Kartashev et al., 2025) for an MLP trained with (a) quantile loss and (b) AdaQR on the heteroskedastic dataset, and (c) quantile loss on the homoskedastic counterpart. Under heteroskedastic noise, earlier epochs perform relatively better on high-uncertainty points and later epochs on low-uncertainty points, indicating that different uncertainty regions favor different stopping times. This separation is reduced with AdaQR and disappears when the noise is homoskedastic. Each curve shows the point-wise difference in squared error relative to the reference epoch with the best aggregate validation MSE; negative values indicate improvement.

of this section in a controlled environment that has these traits, we construct a synthetic dataset $y = f(x) + \varepsilon$, with noise added only to training targets, so test RMSE directly measures recovery of $f = \mathbb{E}[y \mid x]$. We use two variants sharing exactly the same f and x: *heteroskedastic*, with $\varepsilon \sim \mathcal{N}(0, \sigma(x)^2)$, and *homoskedastic*, with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, serving as a control. Importantly, for this dataset, levels of true data uncertainty σ are known to us exactly. See Appendix C for details.

4.1 Optimal stopping epochs vary by uncertainty region

In this subsection we investigate whether models require different stopping epochs for regions with varying data uncertainty. This requires analyzing the test error disaggregated by uncertainty level across epochs.

The *uncertainty plot* (Kartashev et al., 2025) provides this perspective. Figure 2(a) shows an MLP trained with the quantile loss on the heteroskedastic synthetic dataset. Each curve corresponds to one epoch and displays the point-wise change in squared error relative to a reference epoch (the one with the best aggregate validation MSE), plotted against data uncertainty. Before the reference epoch, the model performs better on high-uncertainty points and worse on low-uncertainty ones; afterward, this trend reverses. In other words, the model starts overfitting high-uncertainty regions before it finishes fitting low-uncertainty ones, and the reference epoch is merely a compromise between these competing dynamics.

To verify that this pattern stems from varying noise levels rather than intrinsic properties of x or $f(x)$, Figure 2(c) repeats the experiment on the homoskedastic variant (same f, constant noise scale, with samples sorted as in the heteroskedastic version). The effect disappears: performance changes between epochs are largely unrelated to data uncertainty.

Ideally, a model would retain its early-epoch error on high-uncertainty points while achieving its late-epoch error on low-uncertainty points, improving upon the QR baseline. However, Figure 2(a) only illustrates this misalignment on synthetic data and does not quantify the resulting performance loss. Section 4.2 closes both gaps.

4.2 Measuring the impact of epoch misalignment

To quantify the performance penalty implied by the epoch misalignment in Figure 2(a), we evaluate a hypothetical strategy of *per-cohort early stopping*. This approach requires dividing the validation

Table 3: Cohort-based vs. global checkpoint selection for an MLP trained with quantile loss. Improv. (%) is the relative reduction in mean test RMSE using uncertainty (U/G) or random (R/G) cohorts compared with global selection. All checkpoints are selected on validation data. Bottom rows summarize available comparisons among the seven largest benchmark datasets, excluding synthetic data; W/T/L follows Appendix H.2. Results are computed from mean test RMSEs over 15 random seeds.

	$K = 5$		$K = 10$	
Dataset / summary	**U/G**	**R/G**	**U/G**	**R/G**
Synthetic (heteroskedastic)	$+0.318$	-0.041	$+0.286$	-0.118
Synthetic (homoskedastic)	-0.019	-0.021	-0.048	-0.078
Seven largest benchmark datasets				
Median Improv. (%)	$+0.180$	-0.029	$+0.114$	-0.003
W / T / L	3 / 4 / 0	0 / 7 / 0	3 / 4 / 0	0 / 7 / 0

and test sets into groups of comparable data uncertainty, termed *cohorts*. To construct these, we compute the $K - 1$ empirical $\frac{k}{K}$-quantiles of the estimated uncertainty $\hat{\sigma}(x)$ on the validation set. These thresholds are used to split both the validation and test sets into K cohorts, ordered from lowest (cohort 1) to highest (cohort K) uncertainty. Deriving these thresholds strictly from the validation set prevents data leakage, yielding equal-sized validation cohorts and test cohorts that are balanced up to sampling noise. With the cohorts established, we save a model checkpoint at every epoch during training. For each uncertainty cohort k, we select the checkpoint e_k^* that minimizes the validation MSE restricted specifically to that group. During inference, each test point is routed to the checkpoint e_k^* corresponding to its uncertainty cohort. Because this approach assigns exactly one checkpoint to each test example, the resulting aggregated test RMSE reflects the empirical benefit of tailoring stopping times to local uncertainty levels.

To accurately isolate this benefit, we cannot simply compare the resulting RMSE to standard global early stopping. While per-cohort selection allows the model to choose K different epochs, each choice relies on only a fraction $(1/K)$ of the validation data. To create a fair baseline, we apply the exact same procedure to **random cohorts** of matching size, created by randomly permuting the points before splitting. These random groups suffer the same penalty of smaller validation sets but contain no uncertainty information. Therefore, comparing uncertainty cohorts directly to random cohorts strictly isolates the advantage of grouping by local noise levels.

Table 3 presents the relative improvements in test RMSE when using uncertainty-based (U) and random (R) cohorts compared to standard global (G) early stopping. To minimize the negative effects of smaller validation sets, we restrict this experiment to the seven largest datasets from our benchmark, reporting results for $K = 5$ and $K = 10$. The results demonstrate that uncertainty-based selection improves median RMSE over global selection by 0.18% and 0.114% for five and ten cohorts, respectively. Under our comparison rule, both configurations record three wins, four ties, and no losses across the seven datasets. Furthermore, the drop in overall performance when increasing from $K = 5$ to $K = 10$ cohorts is consistent with increased checkpoint-selection noise from smaller validation subsets.

The table also includes the homoskedastic variant of our synthetic dataset, which shares the same features x and noiseless target function $f(x)$, but features a constant noise scale σ. This allows us to apply the exact same cohort splits used for the heteroskedastic variant. If $\sigma(x)$ were merely acting as a proxy for partitioning the feature space into regions of similar inputs, both variants would show similar gains. Instead, performance on the homoskedastic version worsens when applying uncertainty-based splits, while the opposite is true for the heteroskedastic variant. Thus, the benefit stems specifically from adapting to how varying label noise alters training dynamics, not from the spatial structure of the inputs.

Per-cohort early stopping, however, is not a practical solution: it requires storing numerous checkpoints, assigning test points to cohorts during inference, and merely works around the epoch misalignment rather than fixing it during optimization. Section 5 aims to reduce this mismatch by proposing a modified quantile loss that adapts to the local noise level. This allows a single model, stopped at a single epoch, to perform robustly across the entire uncertainty spectrum.

Table 4: Predictive performance on 28 BeyondArena datasets with at least 1,000 training samples. Gain is the median relative RMSE improvement over MSE on the same backbone; positive values indicate improvement. W/T/L summarizes comparisons with MSE using the rule in Appendix H.2. Bold denotes the largest median gain within each backbone. AdaQR† includes auxiliary scale regularization (Appendix E).

MLP		
Objective	**Gain (%)**	**W / T / L**
NLL	+0.02	9 / 12 / 7
β-NLL	+0.25	10 / 11 / 7
quantile loss	+0.47	16 / 10 / 2
AdaQR	**+0.66**	16 / 10 / 2

Backbone	**Objective**	**Gain (%)**	**W / T / L**
TabPack	quantile loss	+0.22	12 / 14 / 2
	AdaQR	**+0.25**	14 / 14 / 0
RealMLP	quantile loss	+0.35	15 / 11 / 2
	AdaQR†	**+0.82**	19 / 6 / 3
TabM	quantile loss	+0.39	12 / 12 / 4
	AdaQR	**+0.97**	14 / 11 / 3

5 Adaptive quantile regression

Training with the quantile loss can be interpreted, up to quantile discretization, as minimizing the Energy Distance between a predicted distribution and a target Dirac δ distribution. We detail this connection through the Continuous Ranked Probability Score (CRPS) in Appendix B.

To explicitly model potentially different levels of target noise, we propose to use the idea of Adaptive Label Smoothing (Krothapalli & Abbott, 2020), which applies different degrees of smoothing to different samples. To incorporate this logic into the quantile loss, we use its Energy Distance reformulation and replace the deterministic target distribution with a Gaussian distribution $\mathcal{N}(y, \sigma_l^2)$ centered around the target, where σ_l^2 is a learned variance predicted from the input x. We design our loss function such that a larger conditional variance $\mathrm{Var}[y \mid x]$ yields a larger predicted σ_l^2. Specifically, the loss function we use is $\mathcal{L}(F, y, \sigma_l) = \frac{1}{2}\mathcal{D}^2\big(F, \mathcal{N}(y, \sigma_l^2)\big) + \lambda\sigma_l$, where $\mathcal{D}$ denotes the Energy Distance (Székely, 2003). The first term is half of the squared Energy Distance, where the predicted distribution F is estimated through the predicted quantiles q_i. The second term acts as a regularization penalty, scaling the predicted smoothing parameter σ_l by a tunable hyperparameter λ. As shown in Appendix D.1, when expressed in terms of specific predicted quantiles q_i, our loss becomes:

$$\frac{1}{N}\sum_{i=1}^{N}\left((q_i - y)\left(2\Phi\left(\tfrac{q_i - y}{\sigma_l}\right) - 1\right) + \frac{2\sigma_l}{\sqrt{2\pi}}e^{-\frac{(q_i-y)^2}{2\sigma_l^2}}\right) - \frac{1}{N}\sum_{i=1}^{N}(2\tau_i - 1)q_i - \frac{\sigma_l}{\sqrt{\pi}} + \lambda\sigma_l, \quad (1)$$

where N is the number of predicted quantiles, Φ is the CDF of the standard normal distribution, and τ_i represents the probability level of the i-th quantile. Assuming Gaussian label noise, Gaussian model predictions, sufficient quantile resolution, and convergence to the Bayes optimum, Proposition 1 shows that the optimal smoothing variance satisfies $\sigma_l^2 \propto \sigma_t^2$, where $\sigma_t^2 = \mathrm{Var}[y \mid x]$. The full statement and proof are provided in Appendix D.2.

These assumptions provide a theoretical framework for AdaQR's uncertainty adaptation but are not strict prerequisites for empirical success. In practice, AdaQR adds a scalar output that parameterizes σ_l alongside the predicted quantiles. This output is used only during training; at inference, the point prediction is obtained by averaging the quantiles, as in standard quantile regression. Thus, AdaQR introduces no additional test-time cost. Implementation details are provided in Appendix H.5.

5.1 AdaQR results

Figure 2 shows that AdaQR behaves differently from QR during optimization, addressing the misalignment of optimal stopping epochs across uncertainty regions. To test whether this translates to better downstream accuracy, we evaluate AdaQR on 28 datasets with at least 1,000 training samples, excluding the Tiny category of BeyondArena (Purucker et al., 2026). Since AdaQR trains an additional σ_l logit jointly with the main predictions and requires tuning λ, which can complicate optimization on very small datasets, we we exclude such datasets from the main comparison.

We follow our standard evaluation protocol with Muon-based training on MLP, TabPack (Gorishniy et al., 2026a), RealMLP (Holzmüller et al., 2024), and TabM-mini with PLE-Q embeddings (Gor-

ishniy et al., 2025). On MLP, we additionally evaluate Gaussian NLL and β-NLL as heteroskedastic baselines. For RealMLP, we use a scale-regularized variant, denoted by AdaQR†, motivated by the uncertainty-estimation analysis and described in Appendix E.

Table 4 shows that AdaQR achieves positive median RMSE improvements over MSE on all four backbones. Its median gains over MSE are also larger than those of quantile loss, with the clearest differences on TabM and RealMLP and a small difference on TabPack. These results support AdaQR as a useful training alternative, while showing that its benefits vary across backbones.

5.2 Importance of Adaptive Smoothing

To highlight the importance of adaptivity in AdaQR, we conduct an ablation experiment in which the per-sample $\sigma_l(x)$ is replaced with a single global σ_l shared across all instances. Rather than being learned jointly with the model, the global σ_l is fixed and tuned as a hyperparameter. We perform this ablation on RealMLP, a strong-performing tabular model. To limit the additional computational cost, we evaluate on the standard dataset collection excluding the Large subset. We also exclude the Tiny subset as described in Section 5.1. All other settings follow the standard experimental protocol described in Section 3.2.

Gain denotes the median relative RMSE improvement over MSE. W/T/L counts datasets on which an objective is better by at least one standard deviation, comparable, or worse than MSE (Appendix H.2). The substantial degradation with a global σ_l highlights the importance of instance-wise adaptivity in AdaQR.

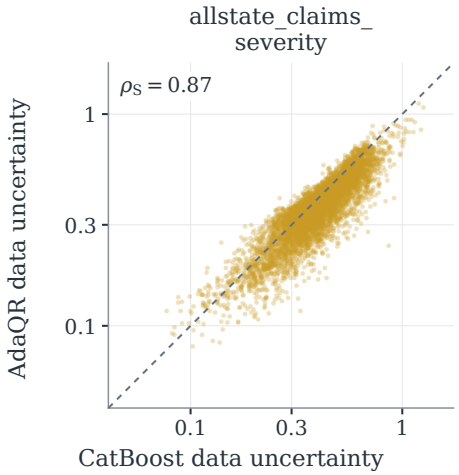


Figure 3: AdaQR versus CatBoost data uncertainty on the `allstate_claims_severity` test set (logarithmic axes). AdaQR estimates are averaged across TabM branches after conversion to variances using equation 3. The dashed line indicates $y = x$; ρ_S is Spearman's rank correlation over all 62,773 test inputs. A random subset of 8,000 inputs is displayed.

5.3 Uncertainty Estimation

We examine whether AdaQR learns meaningful data uncertainty on real-world tasks. Since ground-truth uncertainty is unavailable, we use CatBoost (Prokhorenkova et al., 2018) as a trusted reference. Figure 3 shows strong correlations between their uncertainty estimates, suggesting AdaQR captures meaningful input-dependent variation. See Appendix F for additional examples and implementation details.

Table 5: Ablation of adaptive scale prediction on RealMLP.

Objective	**Gain (%)**	**W / T / L**
quantile loss	+0.54	14 / 4 / 2
AdaQR†	+0.93	16 / 1 / 3
Global σ_l	+0.24	10 / 4 / 6

6 Conclusion

In this paper, we systematically revisit regression objectives for tabular deep learning, comparing 18 objectives on 41 datasets and evaluating leading approaches across multiple backbones. Quantile loss emerges as a consistently strong alternative to carefully tuned MSE. Yet a strong objective does not eliminate the limitations of global early stopping: regions with different noise levels can peak at different training epochs, making a shared checkpoint a compromise. Motivated by this, we introduce AdaQR, which uses jointly learned, input-dependent smoothing as one possible remedy and improves predictions in several settings, with benefits varying across backbones and datasets. Our findings highlight that progress in tabular regression requires attention not only to what models optimize, but also to how different parts of the data are learned over time.

AI USE STATEMENT

We used generative AI tools to assist with implementing experimental code, identifying relevant literature, discussing research methodology and experimental design, and editing the manuscript. The authors reviewed all AI-assisted material and verified AI-generated code. All final methodological decisions and scientific conclusions were made by the authors, who take responsibility for the manuscript, code, and reported results.

REPRODUCIBILITY STATEMENT

The appendix describes the datasets, preprocessing, hyperparameter search spaces, and evaluation protocol. Code and experiment configurations are available in our anonymized repository: `https://anonymous.4open.science/r/tabreg-iclr-DD6E/`. The repository README provides instructions for downloading the datasets and reproducing the experiments.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Guangsheng Bao, Zhiyang Teng, and Yue Zhang. Token-level fitting issues of seq2seq models. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pp. 276–288, 2023.

Eugène Berta, David Holzmüller, Michael I Jordan, and Francis Bach. Rethinking early stopping: Refine, then calibrate. *arXiv preprint arXiv:2501.19195*, 2025.

Christopher M Bishop. Mixture density networks. Technical Report NCRG/94/004, Neural Computing Research Group, Aston University, 1994.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep RL. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=dVpFKfqF3R`.

Matteo Fasiolo, Simon N Wood, Margaux Zaffran, Raphaël Nedellec, and Yannig Goude. Fast calibrated additive quantile regression. *Journal of the American Statistical Association*, 116(535): 1402–1412, 2021.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35:24991–25004, 2022.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations (ICLR)*, 2025.

Yury Gorishniy, Akim Kotelnikov, Ivan Rubachev, and Artem Babenko. Tabpack: Efficient hyperparameter ensembles for tabular deep learning. In *Forty-third International Conference on Machine Learning*, 2026a. URL `https://openreview.net/forum?id=8qHjrpxrQn`.

Yury Gorishniy, Ivan Rubachev, Dmitrii Feoktistov, and Artem Babenko. Benchmarking optimizers for mlps in tabular deep learning. *arXiv preprint arXiv:2604.15297*, 2026b.

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022.

Reinhard Heckel and Fatih Furkan Yilmaz. Early stopping in deep networks: Double descent and how to eliminate it. *arXiv preprint arXiv:2007.10099*, 2020.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *International conference on machine learning*, pp. 2157–2166. PMLR, 2018.

Alexander Immer, Emanuele Palumbo, Alexander Marx, and Julia Vogt. Effective bayesian heteroscedastic regression with deep neural networks. *Advances in Neural Information Processing Systems*, 36:53996–54019, 2023.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Yoonsuh Jung, Yoonkyung Lee, and Steven N MacEachern. Efficient quantile regression for heteroscedastic models. *Journal of Statistical Computation and Simulation*, 85(13):2548–2568, 2015.

Nikolay Kartashev, Ivan Rubachev, and Artem Babenko. Unveiling the role of data uncertainty in tabular deep learning. *arXiv preprint arXiv:2509.04430*, 2025.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Roger Koenker and Gilbert Bassett, Jr. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. ISSN 00129682. doi: 10.2307/1913643. URL http://www.jstor.org/stable/1913643.

Ujwal Krothapalli and A Lynn Abbott. Adaptive label smoothing. *arXiv preprint arXiv:2009.06432*, 2020.

Francesco Laio and Stefania Tamea. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, 11(4):1267–1277, 2007.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Parvin Malekzadeh, Konstantinos N Plataniotis, Zissis Poulos, and Zeyu Wang. A robust quantile huber loss with interpretable parameter adjustment in distributional reinforcement learning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6120–6124. IEEE, 2024.

James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python, 2018. URL https://arxiv.org/abs/1201.0490.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David Holzmüller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, Gaël Varoquaux, Frank Hutter, et al. Beyond iid: How general are tabular foundation models, really? *arXiv preprint arXiv:2606.30410*, 2026.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model. In *Forty-third International Conference on Machine Learning*, 2026.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aPOpXlnV1T.

Xingyou Song and Dara Bahri. Decoding-based regression. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=avUQ8jguxg.

Lawrence Stewart, Francis Bach, and Quentin Berthet. Building bridges between regression, clustering, and classification. *arXiv preprint arXiv:2502.02996*, 2025.

Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. Faithful heteroscedastic regression with neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 5593–5613. PMLR, 2023.

Gábor J Székely. E-statistics: The energy of statistical samples. *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 3(05):1–18, 2003.

Kean Ming Tan, Lan Wang, and Wen-Xin Zhou. High-dimensional quantile regression: Convolution smoothing and concave regularization. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):205–233, 2022.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Eliot Wong-Toi, Alex Boyd, Vincent Fortuin, and Stephan Mandt. Understanding pathologies of deep heteroskedastic regression. *arXiv preprint arXiv:2306.16717*, 2023.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36: 17882–17903, 2023.

Shihao Zhang, Linlin Yang, Michael Bi Mi, Xiaoxu Zheng, and Angela Yao. Improving deep regression with ordinal entropy. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=raU07GpP0P.

Shihao Zhang, Yuguang Yan, and Angela Yao. Improving deep regression with tightness. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dkoiAGjZV9.

Dixian Zhu, Tianbao Yang, and Livnat Jerby. Gradient aligned regression via pairwise losses. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=1bnH3Q3zL6.

A Detailed revisiting objectives description

Notation. We consider supervised regression tasks on tabular data. Let $D = \{(x_i, y_i)\}_{i=1}^n$ denote the dataset, where $x_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \mathbb{R}$ is the label for the i-th datapoint. We denote the deep learning model by $f_\theta(x) : \mathbb{R}^d \to \mathbb{R}^{d'}$, where d' is the output dimension, which is not necessarily equal to 1. Let $l(\cdot, \cdot)$ be the loss function. The loss on a given sample is then defined as $l = l(f_\theta(x), y)$.

A.1 LOSSES

A.1.1 TRADITIONAL LOSSES

The first group of objectives in our comparison includes canonical regression losses. Here, any values denoted by a hat $\hat{\cdot}$ correspond to predictions produced by our model.

- **MSE** (Mean Squared Error, or l_2 loss) is the standard objective for most regression tasks:
$$l = l_2(\hat{y}, y) = (\hat{y} - y)^2$$
- **MAE** (Mean Absolute Error, or l_1 loss) provides robustness against outliers by penalizing errors linearly:
$$l = l_1(\hat{y}, y) = |\hat{y} - y|$$
- **Huber loss** (Huber, 1992) is a middle ground between MSE and MAE:
$$l = l_\delta(\hat{y}, y) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{if } |\hat{y} - y| \le \delta \\ \delta \cdot |y - \hat{y}| - \frac{1}{2}\delta^2, & \text{if } |\hat{y} - y| > \delta \end{cases}$$

A.1.2 BINNING-BASED LOSSES

Among various regression objectives, the binning-based losses (Van Den Oord et al., 2017; Imani & White, 2018; Stewart et al., 2025), stand out as a promising option for regression tasks and are reported to be beneficial across diverse domains (Farebrother et al., 2024). The core idea is to discretize scalar targets using a grid of consecutive bins. Formally, consider a sample $(x, y) \in D$. The regression label y is mapped to a probability vector $\mathbf{p} = (p_1, \ldots, p_{d'})$ over the d' bins on the support $[a, b]$. Here, the i-th bin is the interval $[l_i, r_i]$ with center c_i. Given the model's logits $f_\theta(x)$, let $\mathbf{q} = \text{softmax}(f_\theta(x))$. The model is trained to approximate the target distribution $\mathbf{p}$ via the KL divergence:
$$l = l_{\text{KL}}(f_\theta(x), y) = D_{\text{KL}}(\mathbf{p} \,\|\, \mathbf{q}).$$
At inference, model predictions are computed by taking the expectation over the bins: $\hat{y} = \sum_i \mathbf{q}_i \cdot c_i$.

The particular form of the target distribution is typically the same for all datapoints and using different forms of the target distributions results in different regression objectives. Below, we describe several binning-based objectives that we include in the comparison.

Hard Binning (Van Den Oord et al., 2017; Stewart et al., 2025) maps each y to a one-hot vector $\mathbf{p}$ by assigning it to the nearest bin center.

Soft Binning (Stewart et al., 2025) improves Hard Binning by performing soft mapping to the dense probability vector:
$$\mathbf{p} = \text{softmax}\left(-\frac{|c_1 - y|^2}{2\sigma^2}, \ldots, -\frac{|c_{d'} - y|^2}{2\sigma^2}\right),$$
where σ is a global (shared across the dataset) hyperparameter controlling the spread of probability mass over the bins.

HL-Gaussian (Imani & White, 2018) constructs $\mathbf{p}$ by discretizing the truncated Gaussian distribution with mean y and fixed variance σ^2 over $[a, b]$. Specifically, given the density
$$\rho(t \mid y) = \frac{1}{Z\sqrt{2\pi\sigma^2}} e^{-\frac{(t-y)^2}{2\sigma^2}}$$
the bin probabilities are
$$\mathbf{p}_i = \frac{1}{Z}\left(\Phi\left(\frac{r_i - y}{\sigma}\right) - \Phi\left(\frac{l_i - y}{\sigma}\right)\right)$$
where Φ is the cdf-function for $\mathcal{N}(0, 1)$ and Z is the normalization constant. Similarly to Soft Binning, σ is a global hyperparameter that effectively determines the smoothness of $\mathbf{p}$.

Joint end-to-end encoding (Stewart et al., 2025) is a binning-based objective where the target distribution $\mathbf{p}$ itself is learned. Instead of using a fixed rule (as in the methods above), the mapping $y \mapsto \mathbf{p}$ is parameterized by an auxiliary autoencoder-style model and trained end-to-end along with the main model. This is achieved by adding a reconstruction term (recovering y from $\mathbf{p}$) to the $D_{\text{KL}}(\mathbf{p} \,\|\, \mathbf{q})$ loss.

A.1.3 QUANTILE LOSS

Quantile loss (Koenker & Bassett, 1978) offers an alternative approach to modeling conditional distributions $p(y \mid x)$ by directly predicting specific quantiles of the target variable. Specifically, for a given input x, the model predicts N quantiles $\hat{y}_{\tau_1}, \hat{y}_{\tau_2}, \ldots, \hat{y}_{\tau_N}$ corresponding to quantile levels $\tau_1, \tau_2, \ldots, \tau_N \in (0, 1)$.

Each quantile prediction $\hat{y}_{\tau_i}$ is trained using the pinball loss function:

$$\rho_{\tau_i}(y, \hat{y}_{\tau_i}) = \begin{cases} \tau_i(y - \hat{y}_{\tau_i}), & \text{if } y \geq \hat{y}_{\tau_i} \\ (1 - \tau_i)(\hat{y}_{\tau_i} - y), & \text{if } y < \hat{y}_{\tau_i} \end{cases}$$

where $\tau_i \in (0, 1)$ represents the target quantile level. The overall loss function is then computed as the average pinball loss across all N quantiles:

$$l = l_{\text{QR}}(f_\theta(x), y) = \frac{1}{N} \sum_{i=1}^{N} \rho_{\tau_i}(y, f_{\theta,i}(x))$$

where $f_{\theta,i}(x)$ denotes the model's prediction for the i-th quantile.

During inference, the final prediction $\hat{y}$ is typically obtained by taking the average of all predicted quantiles: $\hat{y} = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_{\tau_i}$.

This approach provides a more direct characterization of the conditional distribution compared to binning methods, while avoiding the discretization of the output space.

A.1.4 MODELING FULL CONDITIONAL DISTRIBUTIONS IN THE ARBITRARY FORM

An alternative paradigm to solve regression problem has been employed in several recent works (Han et al., 2022; Song & Bahri, 2025), that model the conditional distributions $p(y|x)$ without any assumptions about their parametrical form. In our experimental comparison, we include the most recent approach proposed by Song & Bahri (2025) that relies on the autoregressive decoder. Specifically, the approach uses a small autoregressive decoding head on top of an MLP backbone. Both MLP and decoder are trained jointly for the next-token (where tokens represent digits) prediction with a cross-entropy loss.

The decoder predicts the sequence of tokens $t_1, \ldots, t_k$ that represent the target y in an autoregressive manner. Formally, let $\phi(x)$ be the representation produced by the MLP encoder. The model estimates the conditional distribution $p(y \mid x)$ via the following factorization: $p(y \mid x) = \prod_{i=1}^{k} p(t_i \mid t_1, \ldots, t_{i-1}, \phi(x))$.

In our study, we evaluate two versions of this decoder described by the method's authors. In the *normalized* variant, targets are mapped to the $[0, 1]$ interval using min-max transformation before tokenization. In the *unnormalized* variant, targets are represented as floating-point numbers in a base-B system: $y = \pm B^e \cdot m$, where the exponent e and mantissa $m \in [1, B)$ are predicted.

A.1.5 OTHER LOSSES

In this section we list several additional regression objectives that are included in our evaluation.

- **GAR** (Gradient aligned regression) (Zhu et al., 2025) that combines point-wise and pairwise terms to better match the gradient and magnitude of the target function.
- **Softmax-MSE** (Stewart et al., 2025) uses softmax binning head followed by computing the expectation and MSE loss:

  $$l = \left(\sum_i \mathbf{q}_i \cdot c_i - y\right)^2$$

  where $c_i, \mathbf{q}$ follows the binning losses notation from Section A.1.2.
- **Ordinal Entropy loss** (Zhang et al., 2023) is an objective that enables NNs to learn better feature representations Z (specifically, to maximize mutual information $\mathcal{I}(Z, Y) = \mathcal{H}(Z) - \mathcal{H}(Z \mid Y)$), by adding two additional terms to initial regression loss.

- **ROT-REG** (Zhang et al., 2025) is an optimal transport based regularizer that similarly allows to enhance learned representations by tightening them.
- **NLL** assumes $p(y|x) = \mathcal{N}(\mu(x), \sigma^2(x))$ and minimizes the negative log-likelihood:

$$l = \frac{1}{2}\log\hat{\sigma}^2 + \frac{(\hat{\mu} - y)^2}{2\hat{\sigma}^2}.$$

- β**-NLL** (Seitzer et al., 2022) extends **NLL** by reweighting each sample by $\lfloor\hat{\sigma}^{2\beta}\rfloor$ to improve optimization stability, where $\lfloor\cdot\rfloor$ denotes the stop-gradient operation.
- **Mixture Density Networks** (Bishop, 1994) is an objective that models conditional distribution $p(y|x)$ as a Gaussian mixture.

B Quantile loss and Energy Distance

There exists a well-known relation between the quantile loss and the Continuous Ranked Probability Score (CRPS) (Matheson & Winkler, 1976; Laio & Tamea, 2007):

$$\begin{aligned}\mathrm{CRPS}(F, y) &= \int_{-\infty}^{+\infty} \big(F(z) - \mathbb{1}\{z \geq y\}\big)^2 \, dz \\ &= 2\int_0^1 \rho_\tau(y, Q_F(\tau)) \, d\tau \approx 2 \cdot l_{\mathrm{QR}}(Q_F(\cdot), y),\end{aligned}$$

where F is a CDF, Q_F is its quantile function, and ρ_τ is the pinball loss (Section A.1.3). The approximation replaces the integral over quantile levels with an average over the finite uniform grid used by our models. Thus, up to this discretization and a constant factor, minimizing the quantile loss is equivalent to minimizing CRPS.

For real-valued random variables with finite first moments, the squared Energy Distance is (Székely, 2003)

$$\mathcal{D}^2(F, G) = 2\mathbb{E}|X - Y| - \mathbb{E}|X - X'| - \mathbb{E}|Y - Y'|,$$

where $X, X' \sim F$ and $Y, Y' \sim G$ are mutually independent. When $G = \delta_y$, the Dirac distribution concentrated at y, this gives

$$\mathrm{CRPS}(F, y) = \mathbb{E}|X - y| - \frac{1}{2}\mathbb{E}|X - X'| = \frac{1}{2}\mathcal{D}^2(F, \delta_y).$$

This identity provides the Energy Distance interpretation of quantile loss used in Section 5: the predicted distribution is compared with a point mass at the observed target. AdaQR replaces this point mass with an input-dependent Gaussian smoothing distribution.

C Detailed description of the dataset used in Section 4

To evaluate the robustness of our method under strictly controlled noise conditions, we construct both homoskedastic and heteroskedastic synthetic datasets. Below are the details regarding the process of generation.

Dataset Size and Splits:

The total dataset size is $N = 50,000$. We split this into 20,000 training examples, 15,000 validation examples, and 15,000 test examples. Importantly, while the training targets contain injected noise, the validation and test targets are completely noise-free (y_{clean}). This allows us to accurately evaluate how well the models uncover the true underlying conditional expectation.

Base Network Architecture:

We use a standardized Multi-Layer Perceptron (MLP) generator to create non-linear mappings. To ensure stable scales across different initializations, the outputs of this generator are explicitly standardized to have zero mean and unit variance using statistics calculated over an initial batch of $10,000$ Gaussian noise samples. All MLPs use hidden dimensions of 256 and ReLU activations.

Feature Generation (X):

1. We sample a 40-dimensional latent from the standard multivariate Gaussian distribution: $z \sim \mathcal{N}(0, I_{40})$.
2. We map z through a 3-layer MLP generator to produce a 100-dimensional informative feature vector $x_{\text{real}} \in \mathbb{R}^{100}$. We use this step to imitate the complex relationships between features that are often found in tabular data.
3. To introduce irrelevant features, we sample 16-dimensional standard Gaussian noise $x_{\text{noisy}} \sim \mathcal{N}(0, I_{16})$.
4. The final feature vector is the concatenation of the informative and noisy features: $x = [x_{\text{real}}, x_{\text{noisy}}] \in \mathbb{R}^{116}$.

Target Generation (Y): The clean target values, $f(x)$, are generated by passing the informative features x_{real} through a 2-layer MLP f_θ. To create the noisy training targets y_{noisy}, we inject Gaussian noise $\epsilon \sim \mathcal{N}(0, 1)$ according to one of two settings:

- **Homoskedastic Dataset:** The variance is constant across the entire input space.
$$y_{\text{noisy}} = y_{\text{clean}} + \sigma\epsilon, \quad \text{where } \sigma = 4.0$$
- **Heteroskedastic Dataset:** The standard deviation $\sigma(x)$ depends non-linearly on the informative features. We pass x_{real} through another 2-layer MLP g_ϕ to obtain a scalar. This scalar is pushed through a Standard Normal Cumulative Distribution Function Φ to bound it in $(0, 1)$. We then scale, shift, and exponentiate it to create a smooth, variable standard deviation profile. To prevent extreme outliers, the pre-exponentiated value is clamped at 3.0:
$$\sigma(x) = \exp\Big(\min\big(\Phi(g_\phi(x_{\text{real}})) \times 4.5 - 2.0,\ 3.0\big)\Big)$$
$$y_{\text{noisy}} = y_{\text{clean}} + \sigma(x)\epsilon$$

To ensure exact reproducibility, the process uses manual random seeds before any generation.

D AdaQR: Full Objective and Proof of Proposition 1

D.1 Full objective in terms of quantiles

Denoting the predicted quantiles as q_i at levels $0 \le \tau_i \le 1$, $i = 1, \dots, N$, the AdaQR loss expands as follows:

$$\begin{aligned}
&\frac{1}{2}\mathcal{D}^2\big(F, \mathcal{N}(y, \sigma_l^2)\big) + \lambda\sigma_l \\
&= \mathop{\mathbb{E}}_{X,X'\sim F}\left[\mathop{\mathbb{E}}_{Y,Y'\sim\mathcal{N}(y,\sigma_l^2)}\left[\|X-Y\| - \frac{1}{2}\|X-X'\| - \frac{1}{2}\|Y-Y'\|\right]\right] + \lambda\sigma_l \\
&= \mathop{\mathbb{E}}_{Y,Y'\sim\mathcal{N}(y,\sigma_l^2)}\left[\frac{1}{N}\sum_{i=1}^N \|q_i - Y\| - \frac{2}{N}\sum_{i=1}^N \tau_i q_i + \frac{1}{N}\sum_{i=1}^N q_i - \frac{1}{2}\mathbb{E}\|Y-Y'\|\right] + \lambda\sigma_l \\
&= \frac{1}{N}\sum_{i=1}^N \left((q_i - y)\left(2\Phi\Big(\tfrac{q_i-y}{\sigma_l}\Big) - 1\right) + \frac{2\sigma_l}{\sqrt{2\pi}} e^{-\frac{(q_i-y)^2}{2\sigma_l^2}}\right) - \frac{2}{N}\sum_{i=1}^N \tau_i q_i + \frac{1}{N}\sum_{i=1}^N q_i - \frac{\sigma_l}{\sqrt{\pi}} + \lambda\sigma_l,
\end{aligned} \tag{2}$$

where $\Phi(\cdot)$ is the standard normal cdf.

D.2 Proof of Proposition 1

Proposition 1 (AdaQR learns proportional uncertainty). *Under the following assumptions:*

(A1) *the number of quantiles is large enough to ignore discretization effects;*

(A2) *both the predicted distribution F and the target distribution are Gaussian;*

(A3) *the optimizer converges to Bayes-optimal predictions;*

(A4) *the label noise is Gaussian: $y \mid y_{\text{true}} \sim \mathcal{N}(y_{\text{true}},\, \sigma_t^2)$,*

(A5) *$0 < \lambda < 1/\sqrt{\pi}$ and $\sigma_t > 0$*

the AdaQR objective (1) has a unique optimum in σ_l, and the optimal smoothing parameter satisfies:

$$\sigma_l^2 = \frac{(1-\sqrt{\pi}\lambda)^2}{1-(1-\sqrt{\pi}\lambda)^2}\,\sigma_t^2. \tag{3}$$

In particular, $\sigma_l^2 \propto \sigma_t^2$, so AdaQR correctly captures the relative heteroskedasticity of the data.

Proof. Fix an input x and define the expected objective

$$\overline{\mathcal{L}}(\sigma_q, \sigma_l) = \mathbb{E}_{y|x}[\mathcal{L}(F, y, \sigma_l)]\,.$$

By **(A2)** and **(A3)**, we model $F = \mathcal{N}(y_{\text{true}}, \sigma_q^2)$, and by **(A4)**, for each observed label y the target distribution is $G_y = \mathcal{N}(y, \sigma_l^2)$ with $y \sim \mathcal{N}(y_{\text{true}}, \sigma_t^2)$.

The three terms of the Energy Distance simplify via the Half-Normal expectation formula $\mathbb{E}|Z| = \sqrt{2/\pi}\,\sigma$ for $Z \sim \mathcal{N}(0, \sigma^2)$. Note that the relevant quantities are the *centered* differences $X - y_{\text{true}}$, $Y - y$, and $y - y_{\text{true}}$, which are independent and zero-mean (by **(A2)** and **(A4)**):

- $X - Y = (X - y_{\text{true}}) - (y - y_{\text{true}}) - (Y - y)$ is $\mathcal{N}(0,\, \sigma_q^2 + \sigma_t^2 + \sigma_l^2)$;
- $X - X' = (X - y_{\text{true}}) - (X' - y_{\text{true}})$ is $\mathcal{N}(0,\, 2\sigma_q^2)$, since X, X' are i.i.d. samples from F;
- conditionally on a shared label y, the samples $Y, Y' \mid y$ are i.i.d. from $\mathcal{N}(y, \sigma_l^2)$, so $Y - Y' \mid y$ is $\mathcal{N}(0,\, 2\sigma_l^2)$; crucially, this conditional law does not depend on y, hence it also holds marginally after averaging over $y \mid x$.

Combining these and taking the expectation over $y \mid x$, the first term of the expected objective equals:

$$\mathbb{E}_{y|x}\left[\tfrac{1}{2}\mathcal{D}^2(F, G_y)\right] = \frac{1}{\sqrt{\pi}}\Big(\sqrt{2(\sigma_q^2 + \sigma_t^2 + \sigma_l^2)} - \sigma_q - \sigma_l\Big),$$

so, under **(A1)–(A5)**, the expected objective is:

$$\overline{\mathcal{L}}(\sigma_q, \sigma_l) = \frac{1}{\sqrt{\pi}}\Big(\sqrt{2(\sigma_q^2 + \sigma_t^2 + \sigma_l^2)} - \sigma_q - \sigma_l\Big) + \lambda\sigma_l.$$

Setting partial derivatives to zero:

$$\begin{cases} \dfrac{\partial \overline{\mathcal{L}}}{\partial \sigma_q} = \dfrac{1}{\sqrt{\pi}}\left(\dfrac{2\sigma_q}{\sqrt{2(\sigma_q^2 + \sigma_t^2 + \sigma_l^2)}} - 1\right) = 0, \\[2ex] \dfrac{\partial \overline{\mathcal{L}}}{\partial \sigma_l} = \dfrac{1}{\sqrt{\pi}}\left(\dfrac{2\sigma_l}{\sqrt{2(\sigma_q^2 + \sigma_t^2 + \sigma_l^2)}} - 1\right) + \lambda = 0. \end{cases}$$

The first equation gives $\sigma_q^2 = \sigma_l^2 + \sigma_t^2$. Substituting into the second:

$$\frac{4}{\pi}\sigma_l^2 = \left(\frac{1}{\sqrt{\pi}} - \lambda\right)^2 \cdot 4(\sigma_l^2 + \sigma_t^2) \implies \sigma_l^2 = (1 - \sqrt{\pi}\lambda)^2(\sigma_l^2 + \sigma_t^2),$$

and solving for σ_l^2 yields equation 3.

It remains to verify that this stationary point is the unique minimum. For $\sigma_t > 0$, the function $\sqrt{2(\sigma_q^2 + \sigma_l^2 + \sigma_t^2)}$ is strictly convex in (σ_q, σ_l), since its Hessian is positive definite away from the origin and $\sigma_t > 0$ rules out the degenerate case $\sigma_q = \sigma_l = 0$. The remaining terms, $-\sigma_q/\sqrt{\pi} + (1/\sqrt{\pi} - \lambda)\sigma_l$, are linear. Hence $\overline{\mathcal{L}}$ is strictly convex, and under **(A5)** the stationary point found above — which is positive — is its unique global minimum. □

E SCALE REGULARIZATION FOR REALMLP

For RealMLP, we augment AdaQR with an auxiliary scale regularizer, denoting the resulting objective by AdaQR†. Below, we describe its motivation, formulation, and effect on uncertainty estimation.

E.1 UNCERTAINTY ESTIMATION IN REALMLP

Proposition 1 characterizes the population optimum under Gaussian assumptions. In practice, however, there is no guarantee that training reaches this optimum. In particular, a model can correctly rank inputs by uncertainty while producing inaccurate absolute scales.

We compare TabM and RealMLP on our heteroskedastic synthetic dataset, where the true data uncertainty is known, converting smoothing scales into noise-scale estimates via equation 3. TabM already recovers the underlying data uncertainty reasonably well and consistently across ensemble members (Figure 4(a)). RealMLP shows larger scale deviations and greater variation across members (Figure 4(b)), suggesting that these difficulties limit the gains of vanilla AdaQR on this backbone. This motivates an auxiliary regularizer for the scales inferred from the quantile spread and the smoothing output.

E.2 AUXILIARY SCALE REGULARIZATION

We augment AdaQR with a regularizer that provides a direct training signal for the predicted scales. Following the notation of the proof in Appendix D, let $r = 1 - \sqrt{\pi}\lambda$, with $0 < r < 1$. Under the assumptions of Proposition 1, the optimal scales satisfy

$$\sigma_t = \sqrt{1 - r^2}\,\sigma_q = \frac{\sqrt{1 - r^2}}{r}\,\sigma_l.$$

This motivates two estimates of the noise standard deviation:

$$s_q = \sqrt{1 - r^2}\,\widehat{\sigma}_q, \qquad s_\sigma = \frac{\sqrt{1 - r^2}}{r}\,\sigma_l,$$

where, for N predicted quantiles q_j at levels τ_j, we estimate the quantile-distribution scale as

$$\widehat{\sigma}_q^2 = \frac{\frac{1}{N}\sum_{j=1}^{N}(q_j - \bar{q})^2}{\frac{1}{N}\sum_{j=1}^{N}[\Phi^{-1}(\tau_j)]^2}, \qquad \bar{q} = \frac{1}{N}\sum_{j=1}^{N} q_j.$$

Here, Φ is the standard normal CDF. The denominator accounts for the finite, symmetric quantile grid, recovering σ_q^2 exactly for Gaussian quantiles.

We supervise both estimates using CRPS evaluated at Gaussian CDFs

$$F_{\mu,s}(z) = \Phi\left(\frac{z - \mu}{s}\right), \qquad \mu = \mathrm{stopgrad}(\bar{q}),$$

where $\mathrm{stopgrad}$ preserves its argument's value but blocks gradients through it. The Gaussian CRPS has the closed form

$$\mathrm{CRPS}(F_{\mu,s}, y) = e\big(2\Phi(e/s) - 1\big) + 2s\phi(e/s) - \frac{s}{\sqrt{\pi}}, \qquad e = y - \mu,$$

where ϕ is the standard normal density. The resulting objective, denoted by AdaQR†, is

$$\mathcal{L}_{\mathrm{AdaQR}^\dagger} = \mathcal{L}_{\mathrm{AdaQR}} + \frac{\gamma}{2}\left[\mathrm{CRPS}(F_{\mu,s_q}, y) + \mathrm{CRPS}(F_{\mu,s_\sigma}, y)\right],$$

averaged over the training batch, with $\gamma \geq 0$ controlling the regularization strength.

Detaching $\bar{q}$ blocks the auxiliary gradient through the predicted mean, while centering the quantiles makes the spread estimate invariant to a common shift. The regularizer thus directly trains the quantile spread and the smoothing scale, although it can still affect point predictions through the shared backbone.

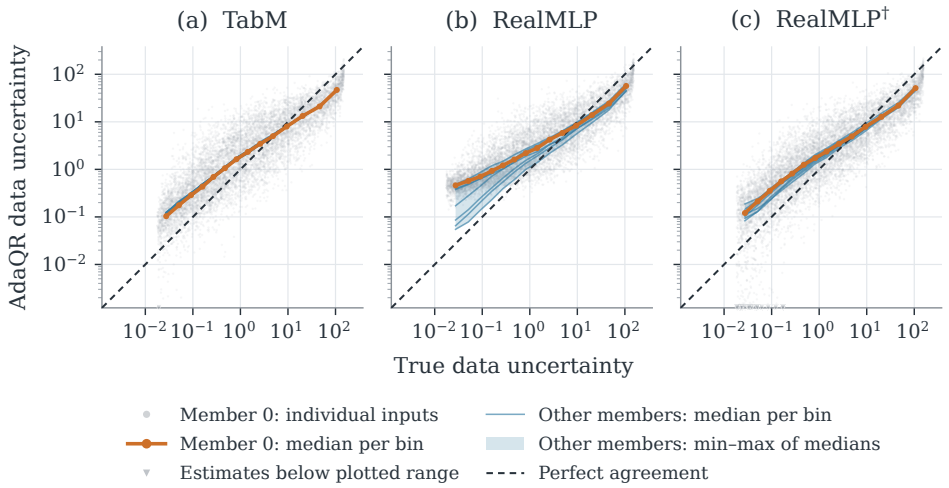


Figure 4: Data uncertainty estimated by AdaQR versus the true data uncertainty on a heteroskedastic synthetic dataset, shown on logarithmic axes. Both uncertainties are expressed as variances. AdaQR estimates are obtained using equation 3, following the theoretical results in Section 5. Panels compare (a) TabM, (b) RealMLP, and (c) RealMLP with scale-regularized AdaQR†. Gray points show individual test inputs for one ensemble member; the orange line connects this member's median estimates in equal-count bins of true uncertainty. Each blue line shows the corresponding bin medians for another ensemble member. The shaded band spans the minimum and maximum of these medians, rather than a confidence interval. Dashed diagonals indicate perfect agreement; downward triangles mark estimates below the displayed range. All three models capture relative variation in data uncertainty. Scale regularization reduces RealMLP's deviations at low uncertainty and improves consistency across ensemble members.

Under the assumptions of Proposition 1, both inferred scales equal σ_t at the population optimum. With the conditional mean correctly specified, $s = \sigma_t$ also minimizes the expected Gaussian CRPS. The auxiliary terms therefore share the same idealized scale optimum as AdaQR.

The regularizer requires no architectural changes: we retain AdaQR's scalar smoothing-scale output. At inference time, the point prediction remains the average of the predicted quantiles.

E.3 Effect on RealMLP

Comparing Figure 4(b) and (c), AdaQR† produces noise-scale estimates that more closely match the true data uncertainty and are more consistent across ensemble members. The improvement primarily concerns the absolute scales, as vanilla AdaQR already ranks inputs by uncertainty reasonably well. These results suggest that the auxiliary regularizer helps RealMLP recover the scale relationship predicted by our theoretical analysis.

F Uncertainty Estimation

Estimating data uncertainty. Following Kartashev et al. (2025), we use a CatBoost model with dual outputs ($\mu(x)$ and $\sigma(x)$) trained via Gaussian negative log-likelihood on the real datasets, utilizing the predicted $\sigma(x)$ as our uncertainty estimate. To prevent data leakage, this estimator is fit exclusively on the training split and applied to validation and test points during inference. For the synthetic data, we use the ground-truth $\sigma(x)$.

Figure 5 extends the comparison with CatBoost to nine additional datasets. We use a single TabM (Gorishniy et al., 2025) checkpoint with 32 branches. For each test input, we convert each branch's learned scale to a data variance using equation 3 and average the resulting variances. Both meth-

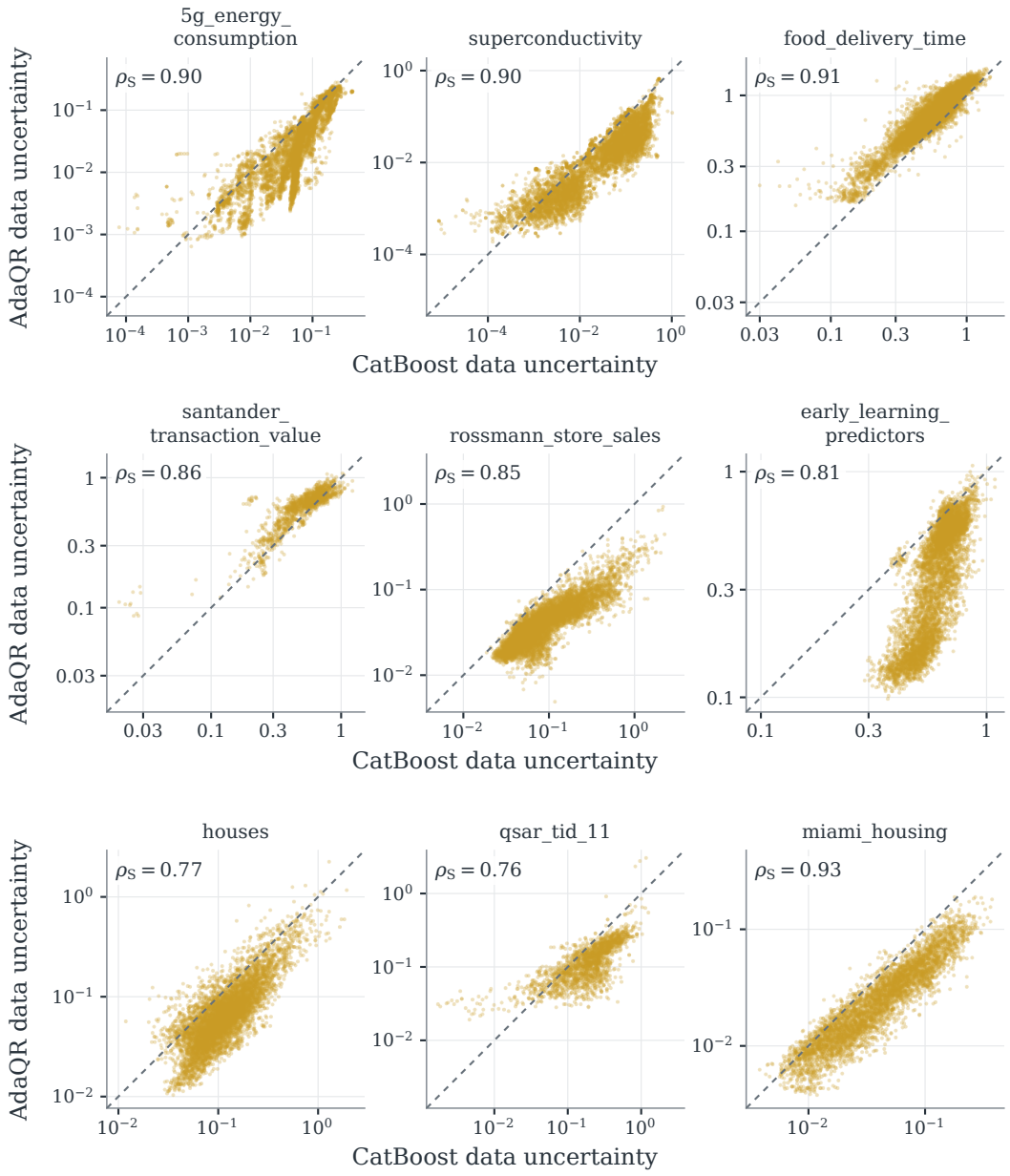


Figure 5: AdaQR versus CatBoost data uncertainty on nine additional real-world datasets. Points represent test inputs; AdaQR estimates are averaged across TabM branches after conversion to variances. Both axes are logarithmic, and dashed lines indicate equal estimates ($y = x$). ρ_S denotes Spearman's rank correlation, computed over all test inputs; at most 8,000 points are displayed per dataset.

ods' estimates are expressed in standardized-target variance units. The positive correlations provide further evidence that AdaQR learns meaningful input-dependent uncertainty.

G DATASETS

Our benchmark comprises 41 datasets from BeyondArena (Purucker et al., 2026) . We exclude `maps_router_eta_1m`, `delivery_eta_1m` and `cooking_time_1m` from the evaluation

due to their combination of high numerical dimensionality (992 for `maps_router_eta_1m`), high categorical cardinality (135131 for `cooking_time_1m`) and large sample size (1.25 million for all three of them). Under our chosen preprocessing and training pipeline, these characteristics make exhaustive hyperparameter tuning infeasible within the 24-hour computational budget imposed for each dataset. We use the lite subset of the benchmark (i.e., the first split of each dataset).

Importantly, GAR[MSE], GAR[MAE] and Ordinal Entropy on MLP show abnormal behavior and fail to converge on `airfoil_self_noise` and on `telemonitoring_parkinsons_biomedical_voice_measurements`. To retain these datasets in the aggregate comparisons, we replace each failed result with the worst result among the methods that converged on the same dataset using the MLP backbone.

Batch sizes. For non-RealMLP models we determine the batch size from the number of training examples n_{train}:

$$B(n_{\text{train}}) = \begin{cases} 32, & n_{\text{train}} \leq 2{,}800, \\ 64, & 2{,}800 < n_{\text{train}} \leq 4{,}500, \\ 128, & 4{,}500 < n_{\text{train}} \leq 6{,}400, \\ 256, & 6{,}400 < n_{\text{train}} \leq 32{,}000, \\ 512, & 32{,}000 < n_{\text{train}} \leq 108{,}000, \\ 1024, & n_{\text{train}} > 108{,}000. \end{cases}$$

For RealMLP we use default batch sizes from PyTabkit - 256 for training and 1024 for prediction.

H Implementation Details

H.1 Experimental protocol

Data preprocessing. For each dataset, the same preprocessing is used. First, we treat binary features as categorical. Both binary and categorical features are then transformed using ordinal encoding. By default, numerical features are processed via a slightly modified version of quantile normalization from `Scikit-learn` (Pedregosa et al., 2018) (see the code). Finally, targets are standardized to zero mean and unit variance, though all final metrics are computed on the original scale. The only exception is the evaluation of decoders (Song & Bahri, 2025), where we follow the authors' protocol by either mapping targets to $[0, 1]$ (for the normalized decoder) or applying no transformations (for the unnormalized version).

Training Process. We use Muon (Jordan et al., 2024) to optimize all layer weight matrices, while AdamW (Loshchilov & Hutter, 2019) is used for bias parameters and the output-layer weight matrix. We do not apply learning rate schedulers. We apply global gradient clipping to 1.0. For each dataset, we use the predefined batch size (see Appendix G) and continue training until there are `patience` consecutive epochs without improvements on the validation set; we set `patience` $= 16$.

Hyperparameter tuning. For each experiment, we carefully tune optimizer, objective and backbone hyperparameters using Optuna's TPE sampler (Akiba et al., 2019), for 100 trials with 24 hour tune time limitation. We use RMSE computed on the validation split as the optimization objective. We provide full parameter spaces in Appendix H.6.

Evaluation. After the best hyperparameters are found, we run the model with those parameters on 15 different random seeds and report the performance on test part of the dataset. For each dataset, we use mean and standard deviation of test metrics (RMSE) over these random seeds to compare methods as described in Appendix H.2).

H.2 Ranks and metrics

As we focus exclusively on regression tasks, we report Root Mean Squared Error (RMSE) as a primary metric. As mentioned in Appendix H.1, all models and objectives are evaluated on 15 random seeds. We compare methods using two aggregation approaches.

Relative Improvement. For each dataset, we compute the mean RMSE over 15 seeds and calculate its relative improvement (in %) over the baseline (typically the standard MSE objective). These

values are shown in the right panel of Figure 1, meanwhile, in the "Improv. (%)" columns of tables, we further aggregate them and report their median over the benchmark datasets.

Ranks. For each dataset, we assign ranks based on a statistically conservative order relation. Method A (with mean RMSE A_μ and standard deviation A_σ) is considered significantly better than method B if $A_\mu + A_\sigma < B_\mu$ (We use this approach for finding W/T/L with A being the reference method). Intuitively, this approach groups methods into "tiers," ensuring that negligible gains are not counted as wins on individual datasets.

Specifically, ranks are computed independently for each dataset using a top-down procedure. We first sort the methods in ascending order of their mean RMSE. The best model is assigned rank 1 and serves as the initial reference. We then iterate through the remaining models: all methods that are not significantly worse than the current reference (according to the aforementioned rule) also receive rank 1. The first model that fails this criterion is assigned rank 2 and becomes the new reference for subsequent comparisons. This process continues until all models are ranked.

The left panel of Figure 1 illustrates the mean and standard deviation of these ranks across the benchmark.

H.3 Support construction for distributional losses

For every binning-based distributional objective we construct the support $[a, b]$ as follows: (i) we compute the range between the 0.005- and 0.995-quantiles of the training labels distribution (we employ quantiles rather than min-max values to mitigate the impact of outliers while capturing the effective support of the distribution); (ii) we expand this range by adding the same interval on both the left and right sides; (iii) we then uniformly partition it into bins.

The uniform binning is the most common in distributional regression research (Imani & White, 2018; Stewart et al., 2025) and performs stably in our experiments.

H.4 Advanced models from Section 3.2.3

Here, we provide implementation details for the experiments in Section 3.2.3. We use TabM-mini with PLE-Q embeddings (Gorishniy et al., 2022), following the configuration and implementation of Gorishniy et al. (2025).

We use Tabpack equipped with numerical embeddings from Gorishniy et al. (2026a). For all objectives on TabPack we do not add any new sampling parameters: We fix num_quantiles and num_bins as 384 (the same as the width) for all probabilistic objectives. Additionally, for AdaQR we fix regulizer as 0.2.

We use the RealMLP (Holzmüller et al., 2024) implementation from PyTabkit[1] and add HL Gaussian, quantile regression, AdaQR to that implementation. We use 100 optuna TPE sampler tuning iterations as for other models and stop tuning at the 24 hour time limit. We use the "tabarena-new" hyperparameter search-space.

H.5 AdaQR implementation

The implementation of AdaQR includes the following details:

1. Alongside the quantile predictions, the model outputs a scalar logit $a_\theta(x)$ for the smoothing scale:
$$\sigma_l(x) = |a_\theta(x) + b_0| + \varepsilon, \qquad \varepsilon = 10^{-5}.$$
Before training, we compute a fixed offset b_0 so that $a_{\theta_0}(x) + b_0$ has mean 0.5 over the training set, separately for each ensemble member. All scales are expressed in standardized target units.
2. The parameter `regulizer` in our implementation corresponds to $1 - \lambda\sqrt{\pi}$, rather than λ itself in the AdaQR objective (Section 5). Its tuning space is provided in Appendix H.6.

[1] `https://github.com/dholzmueller/pytabkit`

3. We use equally spaced quantile levels from 0.001 to 0.999. At inference, we average the predicted quantiles, as in standard quantile regression; the smoothing-scale output is not used.

H.6 Hyperparameter Tuning Spaces

In this section, we provide tuning spaces for the evaluated objectives and methods. Tables 6–8 summarize the optimizer and architecture tuning spaces for different models; Objective-specific hyperparameter tuning spaces are provided in Table 9. Parameter names correspond to their implementation in the source code.

In case of RealMLP backbone we use the "tabarena-new" hyperparameter search-space for RealMLP-specific parameters.

Table 6: The hyperparameter tuning space for MLP. Here, (A) is default space and (B) is used when evaluating Decoder[U] and Decoder[N] (see Section 3.2.2).

Parameter	Distribution
# layers	UniformInt[1, 6]
Width (hidden size)	UniformInt[64, 1024]
Dropout rate	{0.0, Uniform[0.0, 0.5]}
Learning rate (Muon)	LogUniform[1e-4, 3e-2]
Learning rate (AdamW)	(A) LogUniform[3e-5, 1e-3], (B) LogUniform[3e-6, 5e-4]
Weight decay (AdamW)	LogUniform[5e-3, 5]

Table 7: The hyperparameter tuning space for TabM.

Parameter	Distribution
# layers	UniformInt[1, 4]
Width (hidden size)	UniformInt[64, 1024]
Dropout rate	{0.0, Uniform[0.0, 0.5]}
Learning rate (Muon)	LogUniform[1e-3, 1e-1]
Learning rate (AdamW)	LogUniform[5e-5, 3e-3]
Weight decay (AdamW)	LogUniform[1e-3, 1]
d_embedding	UniformInt[8, 32]
n_bins	UniformInt[2, 128]

Table 8: The sampling space for TabPack

Parameter	Distribution
# layers	UniformInt[1, 3]
Width (hidden size)	384
Dropout rate	{0.0, Uniform[0.0, 0.5]}
Learning rate (Muon)	LogUniform[1e-3, 1e-1]
Learning rate (AdamW)	LogUniform[1e-4, 5e-3]
Weight decay (AdamW)	LogUniform[1e-3, 1]
d_embedding	UniformInt[8, 32]
init_scale	LogUniform[1e-2, 1e2]

I Extended Results

In this section we present results from Section 5.1 with Tiny subset from BeyondArena (Purucker et al., 2026) included (Table 10) and all per-dataset results used in this paper (Table 11).

Table 9: Hyperparameter tuning space for different regression objectives. The notation follows Section 3.2.2; refer to the source code for any naming ambiguities.

Objective	Parameter	Distribution
MSE	—	—
MAE	—	—
Huber	delta	LogUniform[1e-2, 100]
NLL	—	—
β-NLL	beta	Uniform[0, 1]
MDN	mixture_size	UniformInt[2, 64]
Hard Binning	n_bins	UniformInt[32, 1024]
Soft Binning	n_bins	UniformInt[32, 1024]
	soft_sigma	LogUniform[1e-2, 5]
HL-Gaussian	hl_num_bins	UniformInt[32, 1024]
	sigma	LogUniform[1e-2, 5]
Joint End-To-End	alpha	{0.0, LogUniform[1e-5, 0.5]}
	lambda_auto	LogUniform[1e-4, 1]
	lambda_kl	LogUniform[1e-4, 1]
GAR[MAE]	alpha	LogUniform[1e-2, 100]
GAR[MSE]	alpha	LogUniform[1e-2, 100]
Decoder[U]	n_blocks	UniformInt[1, 6]
	d_block	UniformInt[64, 512]
	dropout	{0.0, Uniform[0.0, 0.5]}
	n_heads	Categorical[{1, 2, 4, 8}]
	temperature	LogUniform[0.01, 0.5]
	top_k	UniformInt[1, 9]
	base	10
	num_exponent_digits	1
	num_mantissa_digits	5
Decoder[N]	n_blocks	UniformInt[1, 6]
	d_block	UniformInt[64, 512]
	dropout	{0.0, Uniform[0.0, 0.5]}
	n_heads	Categorical[{1, 2, 4, 8}]
	temperature	LogUniform[0.01, 0.5]
	top_k	UniformInt[1, 9]
	base	Categorical[{2, 4, 8, 10}]
	length	UniformInt[4, 16]
Softmax-MSE	min_value_relative	Uniform[-0.1, 0.1]
	max_value_relative	Uniform[0.9, 1.1]
	num_bins	UniformInt[4, 256]
Ordinal Entropy	lambda_d	LogUniform[1e-5, 0.1]
	ordinal_entropy_samples	UniformInt[10, 1000]
ROT-REG	lambda_ot	LogUniform[1e-3, 100]
	opt_transport_samples	UniformInt[2, 200]
	gamma	LogUniform[0.1, 100]
	y_dim	UniformInt[2, 32]
Quantile Loss	num_quantiles	UniformInt[32, 1024]
AdaQR	num_quantiles	UniformInt[32, 1024]
	regulizer	Uniform[1e-3, 1.0]
AdaQR†	num_quantiles	UniformInt[32, 1024]
	regulizer	Uniform[1e-3, 1.0]
	scale_anchor_weight	LogUniform[1e-3, 2]
AdaQR w. global σ_l	sigma	LogUniform[1e-4, 8]

Table 10: Improv. (%) denotes the median relative improvement in RMSE compared to the standard MSE objective. W/T/L counts datasets where the objective is significantly better (by at least one standard deviation, see Appendix H.2), comparable, or worse than MSE.

Model	**Objective**	**Improv. (%)**	**W / T / L**
MLP	Quantile Loss	+0.55	21 / 15 / 5
	AdaQR	+0.74	19 / 17 / 5
	NLL	+0.25	10 / 23 / 8
	beta-NLL	+0.32	12 / 21 / 8
TabPack	Quantile Loss	+0.46	18 / 21 / 2
	AdaQR	+0.45	19 / 21 / 1
RealMLP	Quantile Loss	+0.40	20 / 16 / 5
	AdaQR†	+0.90	25 / 9 / 7
TabM	Quantile Loss	+0.58	20 / 16 / 5
	AdaQR	+0.93	20 / 17 / 4

Table 11: Extended Beyond results with the Muon optimizer. Single denotes one model; E5 denotes an ensemble of five.

5g_energy_consumption

Model	Objective	Single	E5
MLP	MSE	4.07 ± 0.03	4.02 ± 0.01
	MAE	4.17 ± 0.03	4.13 ± 0.01
	Huber Loss	4.15 ± 0.05	4.09 ± 0.02
	Soft Binning	4.13 ± 0.03	4.09 ± 0.02
	Hard Binning	4.06 ± 0.03	4.054 ± 0.007
	Joint End-To-End	4.10 ± 0.04	4.06 ± 0.02
	Decoder[N]	4.48 ± 0.06	4.22 ± 0.05
	HL-Gaussian	4.13 ± 0.04	4.080 ± 0.004
	Decoder[U]	2753 ± 10647	2752 ± 4759
	Softmax MSE	4.14 ± 0.03	4.08 ± 0.02
	Ordinal Entropy	4.11 ± 0.05	4.056 ± 0.008
	ROT-REG	4.06 ± 0.04	4.00 ± 0.03
	GAR [MAE]	4.05 ± 0.04	4.01 ± 0.01
	GAR [MSE]	4.08 ± 0.05	4.020 ± 0.007
	MDN	4.04 ± 0.03	4.008 ± 0.010
	NLL	4.09 ± 0.04	4.03 ± 0.01
	beta-NLL	4.08 ± 0.03	4.039 ± 0.005
	AdaQR	4.04 ± 0.01	4.017 ± 0.002
	Quantile Loss	4.06 ± 0.03	4.026 ± 0.006
TabPack	MSE	4.03 ± 0.02	–
	HL-Gaussian	4.06 ± 0.03	–
	Quantile Loss	4.01 ± 0.02	–
	AdaQR	4.01 ± 0.02	–
RealMLP	MSE	4.04 ± 0.02	4.027 ± 0.008
	HL-Gaussian	4.12 ± 0.03	4.11 ± 0.02
	Quantile Loss	4.06 ± 0.02	4.04 ± 0.01
	AdaQR†	4.03 ± 0.03	4.01 ± 0.02
TabM	MSE	4.01 ± 0.03	3.99 ± 0.01
	HL-Gaussian	3.99 ± 0.02	3.98 ± 0.01
	Quantile Loss	4.01 ± 0.04	3.98 ± 0.02
	AdaQR	4.00 ± 0.02	3.986 ± 0.006

airfoil_self_noise

Model	Objective	Single	E5
MLP	MSE	1.85 ± 0.05	1.79 ± 0.02
	MAE	1.89 ± 0.07	1.82 ± 0.02
	Huber Loss	1.91 ± 0.05	1.85 ± 0.01
	Soft Binning	1.89 ± 0.06	1.804 ± 0.008
	Hard Binning	2.19 ± 0.02	2.172 ± 0.008
	Joint End-To-End	1.93 ± 0.04	1.86 ± 0.01
	Decoder[N]	2.11 ± 0.08	1.944 ± 0.005
	HL-Gaussian	1.90 ± 0.05	1.84 ± 0.03
	Decoder[U]	4 ± 2	3 ± 1
	Softmax MSE	1.86 ± 0.06	1.81 ± 0.04
	Ordinal Entropy	–	–
	ROT-REG	1.82 ± 0.05	1.77 ± 0.02
	GAR [MAE]	–	–
	GAR [MSE]	–	–
	MDN	1.89 ± 0.07	1.825 ± 0.007
	NLL	1.91 ± 0.06	1.81 ± 0.01
	beta-NLL	1.88 ± 0.06	1.84 ± 0.03
	AdaQR	1.86 ± 0.05	1.83 ± 0.03
	Quantile Loss	1.86 ± 0.05	1.815 ± 0.007
TabPack	MSE	1.70 ± 0.03	–
	HL-Gaussian	1.77 ± 0.02	–
	Quantile Loss	1.71 ± 0.03	–
	AdaQR	1.69 ± 0.02	–
RealMLP	MSE	1.55 ± 0.02	1.53 ± 0.01
	HL-Gaussian	1.50 ± 0.01	1.491 ± 0.007
	Quantile Loss	1.43 ± 0.02	1.43 ± 0.01
	AdaQR†	1.47 ± 0.02	1.460 ± 0.007
TabM	MSE	1.64 ± 0.03	1.61 ± 0.01
	HL-Gaussian	1.63 ± 0.02	1.60 ± 0.01
	Quantile Loss	1.68 ± 0.03	1.64 ± 0.01
	AdaQR	1.62 ± 0.02	1.60 ± 0.01

Table 12: Extended Beyond results with the Muon optimizer (continued).

allstate_claims_severity

Model	Objective	Single	E5
MLP	MSE	0.5400 ± 0.0002	0.5388 ± 0.0001
	MAE	0.5408 ± 0.0001	0.5402 ± 0.0000
	Huber Loss	0.5393 ± 0.0002	0.5388 ± 0.0000
	Soft Binning	0.5378 ± 0.0001	0.5373 ± 0.0000
	Hard Binning	0.5393 ± 0.0001	0.5389 ± 0.0000
	Joint End-To-End	0.5394 ± 0.0002	0.5387 ± 0.0000
	Decoder[N]	0.550 ± 0.002	0.5426 ± 0.0010
	HL-Gaussian	0.5386 ± 0.0001	0.5379 ± 0.0001
	Decoder[U]	0.549 ± 0.001	0.5428 ± 0.0007
	Softmax MSE	0.5400 ± 0.0001	0.5392 ± 0.0001
	Ordinal Entropy	0.5389 ± 0.0001	0.5384 ± 0.0001
	ROT-REG	0.5392 ± 0.0001	0.5386 ± 0.0000
	GAR [MAE]	0.5395 ± 0.0001	0.5389 ± 0.0000
	GAR [MSE]	0.5397 ± 0.0001	0.5391 ± 0.0000
	MDN	0.5389 ± 0.0002	0.5381 ± 0.0001
	NLL	0.5400 ± 0.0002	0.5391 ± 0.0000
	beta-NLL	0.5395 ± 0.0002	0.5383 ± 0.0001
	AdaQR	0.5392 ± 0.0001	0.5386 ± 0.0001
	Quantile Loss	0.5392 ± 0.0002	0.5384 ± 0.0000
TabPack	MSE	0.5364 ± 0.0002	–
	HL-Gaussian	0.5357 ± 0.0002	–
	Quantile Loss	0.5357 ± 0.0002	–
	AdaQR	0.5358 ± 0.0002	–
RealMLP	MSE	0.5343 ± 0.0003	0.5333 ± 0.0000
	HL-Gaussian	0.5329 ± 0.0002	0.5320 ± 0.0001
	Quantile Loss	0.5331 ± 0.0002	0.5318 ± 0.0001
	AdaQR†	0.5335 ± 0.0001	0.5327 ± 0.0001
TabM	MSE	0.5344 ± 0.0002	0.5335 ± 0.0001
	HL-Gaussian	0.5339 ± 0.0002	0.5330 ± 0.0001
	Quantile Loss	0.5346 ± 0.0003	0.5336 ± 0.0000
	AdaQR	0.5341 ± 0.0003	0.5325 ± 0.0000

blood_tests_drink_prediction

Model	Objective	Single	E5
MLP	MSE	0.667 ± 0.009	0.666 ± 0.006
	MAE	0.68 ± 0.01	0.682 ± 0.004
	Huber Loss	0.68 ± 0.02	0.677 ± 0.005
	Soft Binning	0.667 ± 0.005	0.666 ± 0.003
	Hard Binning	0.70 ± 0.02	0.693 ± 0.003
	Joint End-To-End	0.671 ± 0.006	0.670 ± 0.002
	Decoder[N]	0.86 ± 0.04	0.76 ± 0.01
	HL-Gaussian	0.652 ± 0.008	0.6506 ± 0.0008
	Decoder[U]	0.74 ± 0.03	0.699 ± 0.009
	Softmax MSE	0.669 ± 0.006	0.668 ± 0.004
	Ordinal Entropy	0.69 ± 0.02	0.678 ± 0.003
	ROT-REG	0.662 ± 0.010	0.660 ± 0.005
	GAR [MAE]	0.68 ± 0.02	0.674 ± 0.005
	GAR [MSE]	0.674 ± 0.008	0.672 ± 0.005
	MDN	0.66 ± 0.01	0.655 ± 0.007
	NLL	0.674 ± 0.008	0.671 ± 0.005
	beta-NLL	0.674 ± 0.008	0.668 ± 0.008
	AdaQR	0.687 ± 0.008	0.685 ± 0.002
	Quantile Loss	0.678 ± 0.006	0.678 ± 0.004
TabPack	MSE	0.66 ± 0.01	–
	HL-Gaussian	0.668 ± 0.009	–
	Quantile Loss	0.654 ± 0.005	–
	AdaQR	0.66 ± 0.01	–
RealMLP	MSE	0.74 ± 0.03	0.693 ± 0.005
	HL-Gaussian	0.666 ± 0.005	0.665 ± 0.002
	Quantile Loss	0.670 ± 0.008	0.668 ± 0.002
	AdaQR†	0.68 ± 0.01	0.668 ± 0.006
TabM	MSE	0.671 ± 0.008	0.670 ± 0.001
	HL-Gaussian	0.67 ± 0.01	0.666 ± 0.004
	Quantile Loss	0.675 ± 0.007	0.672 ± 0.003
	AdaQR	0.67 ± 0.01	0.672 ± 0.006

Table 13: Extended Beyond results with the Muon optimizer (continued).

body_density_prediction

Model	Objective	Single	E5
MLP	MSE	0.013 ± 0.002	0.0127 ± 0.0002
	MAE	0.0121 ± 0.0002	0.0119 ± 0.0001
	Huber Loss	0.0125 ± 0.0004	0.0123 ± 0.0000
	Soft Binning	0.0137 ± 0.0002	0.0137 ± 0.0001
	Hard Binning	0.0153 ± 0.0008	0.0143 ± 0.0004
	Joint End-To-End	0.0136 ± 0.0002	0.0133 ± 0.0001
	Decoder[N]	0.014 ± 0.001	0.0127 ± 0.0004
	HL-Gaussian	0.0127 ± 0.0002	0.0125 ± 0.0002
	Decoder[U]	0.016 ± 0.002	0.0137 ± 0.0004
	Softmax MSE	0.0136 ± 0.0004	0.0131 ± 0.0002
	Ordinal Entropy	0.0130 ± 0.0006	0.0123 ± 0.0002
	ROT-REG	0.0131 ± 0.0007	0.0127 ± 0.0005
	GAR [MAE]	0.0127 ± 0.0005	0.0121 ± 0.0002
	GAR [MSE]	0.0127 ± 0.0005	0.0124 ± 0.0001
	MDN	0.0127 ± 0.0003	0.0126 ± 0.0001
	NLL	0.0130 ± 0.0006	0.0126 ± 0.0003
	beta-NLL	0.0131 ± 0.0004	0.0125 ± 0.0001
	AdaQR	0.0129 ± 0.0004	0.0126 ± 0.0001
	Quantile Loss	0.0125 ± 0.0002	0.0124 ± 0.0002
TabPack	MSE	0.0125 ± 0.0005	–
	HL-Gaussian	0.0129 ± 0.0005	–
	Quantile Loss	0.0116 ± 0.0003	–
	AdaQR	0.0123 ± 0.0003	–
RealMLP	MSE	0.0113 ± 0.0001	0.0113 ± 0.0001
	HL-Gaussian	0.0112 ± 0.0001	0.0112 ± 0.0000
	Quantile Loss	0.0114 ± 0.0002	0.0114 ± 0.0001
	AdaQR†	0.0115 ± 0.0001	0.0114 ± 0.0001
TabM	MSE	0.0126 ± 0.0002	0.0126 ± 0.0001
	HL-Gaussian	0.0129 ± 0.0001	0.0128 ± 0.0001
	Quantile Loss	0.0120 ± 0.0004	0.0119 ± 0.0001
	AdaQR	0.0124 ± 0.0002	0.0123 ± 0.0000

california_house_prices_2020

Model	Objective	Single	E5
MLP	MSE	0.274 ± 0.002	0.2710 ± 0.0008
	MAE	0.260 ± 0.002	0.2573 ± 0.0008
	Huber Loss	0.267 ± 0.002	0.258 ± 0.001
	Soft Binning	0.257 ± 0.002	0.2531 ± 0.0006
	Hard Binning	0.306 ± 0.003	0.294 ± 0.002
	Joint End-To-End	0.275 ± 0.002	0.2718 ± 0.0009
	Decoder[N]	0.312 ± 0.004	0.280 ± 0.002
	HL-Gaussian	0.2622 ± 0.0009	0.2597 ± 0.0002
	Decoder[U]	0.306 ± 0.005	0.2895 ± 0.0009
	Softmax MSE	0.274 ± 0.004	0.268 ± 0.002
	Ordinal Entropy	0.273 ± 0.005	0.266 ± 0.002
	ROT-REG	0.255 ± 0.003	0.253 ± 0.002
	GAR [MAE]	0.261 ± 0.003	0.254 ± 0.002
	GAR [MSE]	0.275 ± 0.002	0.2658 ± 0.0009
	MDN	0.263 ± 0.004	0.2573 ± 0.0007
	NLL	0.262 ± 0.003	0.260 ± 0.001
	beta-NLL	0.268 ± 0.003	0.265 ± 0.002
	AdaQR	0.245 ± 0.002	0.2406 ± 0.0009
	Quantile Loss	0.268 ± 0.004	0.262 ± 0.001
TabPack	MSE	0.2009 ± 0.0006	–
	HL-Gaussian	0.202 ± 0.001	–
	Quantile Loss	0.202 ± 0.002	–
	AdaQR	0.2002 ± 0.0006	–
RealMLP	MSE	0.2010 ± 0.0004	0.2001 ± 0.0003
	HL-Gaussian	0.2022 ± 0.0009	0.2003 ± 0.0007
	Quantile Loss	0.2004 ± 0.0004	0.1989 ± 0.0004
	AdaQR†	0.2007 ± 0.0008	0.1993 ± 0.0003
TabM	MSE	0.212 ± 0.002	0.2078 ± 0.0004
	HL-Gaussian	0.206 ± 0.002	0.2040 ± 0.0006
	Quantile Loss	0.202 ± 0.001	0.1997 ± 0.0007
	AdaQR	0.206 ± 0.002	0.2027 ± 0.0003

Table 14: Extended Beyond results with the Muon optimizer (continued).

cirrhosis_patient_survival_prediction

Model	Objective	Single	E5
MLP	MSE	0.72 ± 0.02	0.705 ± 0.005
	MAE	0.66 ± 0.02	0.655 ± 0.008
	Huber Loss	0.67 ± 0.01	0.668 ± 0.005
	Soft Binning	0.69 ± 0.01	0.693 ± 0.006
	Hard Binning	0.93 ± 0.07	0.83 ± 0.02
	Joint End-To-End	0.74 ± 0.01	0.734 ± 0.002
	Decoder[N]	1.0 ± 0.2	0.80 ± 0.08
	HL-Gaussian	0.78 ± 0.03	0.779 ± 0.008
	Decoder[U]	0.83 ± 0.06	0.76 ± 0.02
	Softmax MSE	0.73 ± 0.01	0.725 ± 0.002
	Ordinal Entropy	0.72 ± 0.02	0.712 ± 0.005
	ROT-REG	0.76 ± 0.04	0.73 ± 0.01
	GAR [MAE]	0.68 ± 0.01	0.678 ± 0.003
	GAR [MSE]	0.67 ± 0.02	0.671 ± 0.007
	MDN	0.68 ± 0.02	0.680 ± 0.006
	NLL	0.70 ± 0.02	0.69 ± 0.01
	beta-NLL	0.64 ± 0.01	0.636 ± 0.002
	AdaQR	0.680 ± 0.010	0.679 ± 0.002
	Quantile Loss	0.68 ± 0.01	0.681 ± 0.005
TabPack	MSE	0.77 ± 0.03	–
	HL-Gaussian	0.91 ± 0.08	–
	Quantile Loss	0.75 ± 0.02	–
	AdaQR	0.72 ± 0.03	–
RealMLP	MSE	0.70 ± 0.06	0.68 ± 0.04
	HL-Gaussian	0.78 ± 0.05	0.74 ± 0.03
	Quantile Loss	0.658 ± 0.009	0.656 ± 0.001
	AdaQR†	0.70 ± 0.05	0.69 ± 0.01
TabM	MSE	0.649 ± 0.007	0.646 ± 0.001
	HL-Gaussian	0.67 ± 0.01	0.671 ± 0.006
	Quantile Loss	0.68 ± 0.01	0.676 ± 0.005
	AdaQR	–	–

climate_model_weather_forecasting_1m

Model	Objective	Single	E5
MLP	MSE	1.474 ± 0.009	1.468 ± 0.006
	MAE	1.454 ± 0.008	1.440 ± 0.006
	Huber Loss	1.47 ± 0.01	1.458 ± 0.004
	Soft Binning	1.470 ± 0.008	1.460 ± 0.001
	Hard Binning	1.448 ± 0.007	1.434 ± 0.003
	Joint End-To-End	1.46 ± 0.01	1.443 ± 0.007
	Decoder[N]	1.51 ± 0.01	1.455 ± 0.008
	HL-Gaussian	1.467 ± 0.008	1.455 ± 0.004
	Decoder[U]	1.49 ± 0.01	1.444 ± 0.005
	Softmax MSE	1.49 ± 0.01	1.473 ± 0.005
	Ordinal Entropy	1.50 ± 0.01	1.481 ± 0.002
	ROT-REG	1.470 ± 0.008	1.463 ± 0.005
	GAR [MAE]	1.451 ± 0.008	1.442 ± 0.003
	GAR [MSE]	1.476 ± 0.009	1.468 ± 0.002
	MDN	1.45 ± 0.01	1.437 ± 0.004
	NLL	1.45 ± 0.01	1.436 ± 0.005
	beta-NLL	1.436 ± 0.008	1.4259 ± 0.0008
	AdaQR	1.46 ± 0.01	1.444 ± 0.008
	Quantile Loss	1.45 ± 0.01	1.444 ± 0.005
TabPack	MSE	1.376 ± 0.004	–
	HL-Gaussian	1.373 ± 0.004	–
	Quantile Loss	1.365 ± 0.003	–
	AdaQR	1.368 ± 0.003	–
RealMLP	MSE	1.402 ± 0.005	1.389 ± 0.003
	HL-Gaussian	1.372 ± 0.003	1.365 ± 0.002
	Quantile Loss	1.379 ± 0.005	1.371 ± 0.002
	AdaQR†	1.380 ± 0.006	1.372 ± 0.002
TabM	MSE	1.394 ± 0.009	1.388 ± 0.002
	HL-Gaussian	1.382 ± 0.005	1.375 ± 0.005
	Quantile Loss	1.368 ± 0.009	1.356 ± 0.003
	AdaQR	–	1.354 ± 0.001

Table 15: Extended Beyond results with the Muon optimizer (continued).

coffee_rating_prediction

Model	Objective	Single	E5
MLP	MSE	1.48 ± 0.02	1.4771 ± 0.0003
	MAE	1.48 ± 0.05	1.46 ± 0.02
	Huber Loss	1.54 ± 0.07	1.51 ± 0.02
	Soft Binning	1.43 ± 0.04	1.406 ± 0.008
	Hard Binning	1.53 ± 0.04	1.53 ± 0.02
	Joint End-To-End	1.51 ± 0.02	1.503 ± 0.008
	Decoder[N]	1.41 ± 0.05	1.36 ± 0.02
	HL-Gaussian	1.40 ± 0.02	1.390 ± 0.005
	Decoder[U]	1.5 ± 0.1	1.44 ± 0.03
	Softmax MSE	1.44 ± 0.02	1.424 ± 0.005
	Ordinal Entropy	1.50 ± 0.03	1.495 ± 0.007
	ROT-REG	1.48 ± 0.02	1.465 ± 0.006
	GAR [MAE]	1.54 ± 0.06	1.52 ± 0.02
	GAR [MSE]	1.53 ± 0.05	1.50 ± 0.03
	MDN	1.56 ± 0.02	1.553 ± 0.003
	NLL	1.49 ± 0.05	1.47 ± 0.02
	beta-NLL	1.54 ± 0.03	1.5312 ± 0.0010
	AdaQR	1.50 ± 0.07	1.48 ± 0.03
	Quantile Loss	1.57 ± 0.05	1.56 ± 0.01
TabPack	MSE	1.34 ± 0.02	–
	HL-Gaussian	1.36 ± 0.02	–
	Quantile Loss	1.33 ± 0.02	–
	AdaQR	1.34 ± 0.02	–
RealMLP	MSE	1.31 ± 0.04	1.291 ± 0.010
	HL-Gaussian	1.28 ± 0.02	1.276 ± 0.009
	Quantile Loss	1.27 ± 0.01	1.269 ± 0.007
	AdaQR†	1.259 ± 0.007	1.252 ± 0.003
TabM	MSE	1.33 ± 0.02	1.311 ± 0.006
	HL-Gaussian	1.34 ± 0.03	1.33 ± 0.01
	Quantile Loss	1.29 ± 0.04	1.28 ± 0.02
	AdaQR	1.30 ± 0.02	1.289 ± 0.009

concrete_compressive_strength

Model	Objective	Single	E5
MLP	MSE	4.8 ± 0.1	4.59 ± 0.03
	MAE	4.7 ± 0.1	4.52 ± 0.06
	Huber Loss	4.59 ± 0.08	4.54 ± 0.02
	Soft Binning	4.65 ± 0.08	4.58 ± 0.02
	Hard Binning	5.64 ± 0.06	5.61 ± 0.03
	Joint End-To-End	4.63 ± 0.04	4.59 ± 0.01
	Decoder[N]	5.1 ± 0.2	4.73 ± 0.08
	HL-Gaussian	4.88 ± 0.09	4.791 ± 0.005
	Decoder[U]	6.2 ± 0.3	5.21 ± 0.08
	Softmax MSE	4.67 ± 0.03	4.63 ± 0.01
	Ordinal Entropy	4.7 ± 0.1	4.62 ± 0.09
	ROT-REG	4.73 ± 0.06	4.62 ± 0.01
	GAR [MAE]	4.69 ± 0.06	4.63 ± 0.04
	GAR [MSE]	4.58 ± 0.04	4.544 ± 0.005
	MDN	4.7 ± 0.1	4.66 ± 0.02
	NLL	4.8 ± 0.1	4.57 ± 0.05
	beta-NLL	4.63 ± 0.07	4.57 ± 0.02
	AdaQR	4.9 ± 0.1	4.72 ± 0.01
	Quantile Loss	4.57 ± 0.04	4.54 ± 0.02
TabPack	MSE	4.14 ± 0.09	–
	HL-Gaussian	4.12 ± 0.10	–
	Quantile Loss	3.99 ± 0.08	–
	AdaQR	3.95 ± 0.09	–
RealMLP	MSE	4.00 ± 0.06	3.94 ± 0.02
	HL-Gaussian	4.06 ± 0.05	4.020 ± 0.009
	Quantile Loss	3.92 ± 0.06	3.858 ± 0.008
	AdaQR†	4.16 ± 0.05	4.111 ± 0.006
TabM	MSE	4.08 ± 0.07	3.964 ± 0.009
	HL-Gaussian	3.80 ± 0.05	3.76 ± 0.01
	Quantile Loss	3.84 ± 0.05	3.80 ± 0.03
	AdaQR	3.94 ± 0.08	3.86 ± 0.02

Table 16: Extended Beyond results with the Muon optimizer (continued).

diamonds

Model	Objective	Single	E5
MLP	MSE	0.0892 ± 0.0008	0.0856 ± 0.0001
	MAE	0.0869 ± 0.0008	0.0857 ± 0.0005
	Huber Loss	0.0879 ± 0.0005	0.0852 ± 0.0001
	Soft Binning	0.0879 ± 0.0006	0.0876 ± 0.0002
	Hard Binning	0.0879 ± 0.0006	0.0873 ± 0.0004
	Joint End-To-End	0.0875 ± 0.0004	0.0870 ± 0.0002
	Decoder[N]	0.094 ± 0.002	0.0868 ± 0.0001
	HL-Gaussian	0.0873 ± 0.0004	0.0867 ± 0.0002
	Decoder[U]	0.097 ± 0.002	0.0908 ± 0.0001
	Softmax MSE	0.0890 ± 0.0005	0.0852 ± 0.0002
	Ordinal Entropy	0.0867 ± 0.0002	0.0863 ± 0.0001
	ROT-REG	–	–
	GAR [MAE]	0.0883 ± 0.0004	0.0856 ± 0.0003
	GAR [MSE]	0.0888 ± 0.0005	0.0869 ± 0.0002
	MDN	0.0877 ± 0.0008	0.0844 ± 0.0001
	NLL	0.0875 ± 0.0004	0.0858 ± 0.0000
	beta-NLL	0.0879 ± 0.0006	0.0859 ± 0.0003
	AdaQR	0.0878 ± 0.0006	0.0855 ± 0.0003
	Quantile Loss	0.0877 ± 0.0006	0.0855 ± 0.0001
TabPack	MSE	0.0826 ± 0.0002	–
	HL-Gaussian	0.0823 ± 0.0001	–
	Quantile Loss	0.0820 ± 0.0001	–
	AdaQR	0.0819 ± 0.0001	–
RealMLP	MSE	0.0817 ± 0.0002	0.0808 ± 0.0000
	HL-Gaussian	0.0820 ± 0.0002	0.0812 ± 0.0000
	Quantile Loss	0.0818 ± 0.0003	0.0813 ± 0.0000
	AdaQR†	0.0817 ± 0.0003	0.0811 ± 0.0001
TabM	MSE	0.0830 ± 0.0003	0.0823 ± 0.0001
	HL-Gaussian	0.0829 ± 0.0004	0.0820 ± 0.0001
	Quantile Loss	0.0826 ± 0.0005	0.0817 ± 0.0003
	AdaQR	0.0821 ± 0.0003	0.0813 ± 0.0001

early_learning_predictors

Model	Objective	Single	E5
MLP	MSE	13.38 ± 0.04	13.32 ± 0.02
	MAE	13.23 ± 0.02	13.212 ± 0.006
	Huber Loss	13.23 ± 0.01	13.207 ± 0.003
	Soft Binning	13.25 ± 0.02	13.21 ± 0.01
	Hard Binning	13.26 ± 0.02	13.237 ± 0.004
	Joint End-To-End	13.33 ± 0.05	13.25 ± 0.04
	Decoder[N]	15 ± 1	13.7 ± 0.4
	HL-Gaussian	13.192 ± 0.009	13.182 ± 0.005
	Decoder[U]	15 ± 1	13.9 ± 0.5
	Softmax MSE	13.24 ± 0.02	13.21 ± 0.01
	Ordinal Entropy	13.27 ± 0.03	13.227 ± 0.002
	ROT-REG	13.24 ± 0.01	13.225 ± 0.008
	GAR [MAE]	13.23 ± 0.02	13.220 ± 0.008
	GAR [MSE]	13.22 ± 0.01	13.202 ± 0.005
	MDN	13.26 ± 0.04	13.21 ± 0.01
	NLL	13.37 ± 0.03	13.31 ± 0.01
	beta-NLL	13.24 ± 0.02	13.215 ± 0.006
	AdaQR	13.31 ± 0.03	13.26 ± 0.02
	Quantile Loss	13.29 ± 0.04	13.257 ± 0.009
TabPack	MSE	13.05 ± 0.02	–
	HL-Gaussian	13.04 ± 0.02	–
	Quantile Loss	13.04 ± 0.02	–
	AdaQR	13.05 ± 0.03	–
RealMLP	MSE	13.02 ± 0.03	12.951 ± 0.009
	HL-Gaussian	13.11 ± 0.03	12.993 ± 0.004
	Quantile Loss	13.01 ± 0.02	12.953 ± 0.007
	AdaQR†	13.09 ± 0.04	13.025 ± 0.009
TabM	MSE	13.06 ± 0.02	13.033 ± 0.008
	HL-Gaussian	13.12 ± 0.03	13.08 ± 0.03
	Quantile Loss	13.09 ± 0.03	13.052 ± 0.009
	AdaQR	13.07 ± 0.03	13.046 ± 0.004

Table 17: Extended Beyond results with the Muon optimizer (continued).

electric_motor_temperature_prediction

Model	Objective	Single	E5
MLP	MSE	2.62 ± 0.03	2.61 ± 0.02
	MAE	2.96 ± 0.08	2.92 ± 0.03
	Huber Loss	2.65 ± 0.06	2.63 ± 0.04
	Soft Binning	2.66 ± 0.05	2.64 ± 0.02
	Hard Binning	3.7 ± 0.2	3.53 ± 0.09
	Joint End-To-End	2.6 ± 0.1	2.55 ± 0.05
	Decoder[N]	3.7 ± 0.2	3.31 ± 0.09
	HL-Gaussian	2.66 ± 0.03	2.652 ± 0.007
	Decoder[U]	270 ± 1005	266 ± 445
	Softmax MSE	2.7 ± 0.2	2.5 ± 0.1
	Ordinal Entropy	2.68 ± 0.05	2.65 ± 0.02
	ROT-REG	2.9 ± 0.2	2.81 ± 0.09
	GAR [MAE]	2.61 ± 0.07	2.56 ± 0.02
	GAR [MSE]	2.6 ± 0.1	2.49 ± 0.05
	MDN	2.88 ± 0.10	2.83 ± 0.09
	NLL	2.62 ± 0.03	2.60 ± 0.02
	beta-NLL	2.64 ± 0.05	2.618 ± 0.005
	AdaQR	2.7 ± 0.2	2.60 ± 0.03
	Quantile Loss	2.6 ± 0.1	2.49 ± 0.06
TabPack	MSE	2.5 ± 0.1	–
	HL-Gaussian	2.40 ± 0.06	–
	Quantile Loss	2.5 ± 0.1	–
	AdaQR	2.37 ± 0.10	–
RealMLP	MSE	2.36 ± 0.04	2.35 ± 0.02
	HL-Gaussian	2.42 ± 0.02	2.41 ± 0.02
	Quantile Loss	2.23 ± 0.06	2.20 ± 0.02
	AdaQR†	2.29 ± 0.04	2.258 ± 0.007
TabM	MSE	2.4 ± 0.1	2.33 ± 0.02
	HL-Gaussian	2.28 ± 0.05	2.27 ± 0.02
	Quantile Loss	2.50 ± 0.05	2.49 ± 0.01
	AdaQR	2.27 ± 0.07	2.24 ± 0.03

fiat_500

Model	Objective	Single	E5
MLP	MSE	861 ± 9	859 ± 2
	MAE	854 ± 7	852 ± 2
	Huber Loss	830 ± 5	828.0 ± 0.4
	Soft Binning	807 ± 5	803.8 ± 0.9
	Hard Binning	806 ± 3	805 ± 1
	Joint End-To-End	828 ± 4	827 ± 3
	Decoder[N]	1237 ± 632	1019 ± 178
	HL-Gaussian	811 ± 1	811.0 ± 0.9
	Decoder[U]	890 ± 18	822 ± 6
	Softmax MSE	839 ± 7	836 ± 3
	Ordinal Entropy	838 ± 9	835 ± 2
	ROT-REG	834 ± 8	831.7 ± 0.7
	GAR [MAE]	1056 ± 477	992 ± 292
	GAR [MSE]	829 ± 7	826 ± 5
	MDN	828 ± 5	823 ± 2
	NLL	844 ± 13	841 ± 4
	beta-NLL	832 ± 7	829 ± 6
	AdaQR	831 ± 5	830 ± 2
	Quantile Loss	826 ± 7	821 ± 3
TabPack	MSE	794 ± 10	–
	HL-Gaussian	776 ± 5	–
	Quantile Loss	807 ± 19	–
	AdaQR	801 ± 10	–
RealMLP	MSE	789 ± 9	782 ± 2
	HL-Gaussian	791 ± 6	783 ± 2
	Quantile Loss	787 ± 5	783 ± 2
	AdaQR†	781 ± 6	774 ± 2
TabM	MSE	809 ± 7	800 ± 1
	HL-Gaussian	805 ± 4	803 ± 2
	Quantile Loss	813 ± 12	809 ± 4
	AdaQR	819 ± 8	810 ± 3

Table 18: Extended Beyond results with the Muon optimizer (continued).

food_delivery_time

Model	Objective	Single	E5
MLP	MSE	7.73 ± 0.01	7.715 ± 0.008
	MAE	7.81 ± 0.03	7.750 ± 0.008
	Huber Loss	7.75 ± 0.01	7.714 ± 0.007
	Soft Binning	7.748 ± 0.008	7.742 ± 0.002
	Hard Binning	7.65 ± 0.01	7.636 ± 0.002
	Joint End-To-End	7.73 ± 0.01	7.725 ± 0.005
	Decoder[N]	8.3 ± 0.2	7.93 ± 0.04
	HL-Gaussian	7.69 ± 0.01	7.680 ± 0.004
	Decoder[U]	8.24 ± 0.09	8.03 ± 0.03
	Softmax MSE	7.78 ± 0.03	7.738 ± 0.005
	Ordinal Entropy	7.72 ± 0.02	7.692 ± 0.008
	ROT-REG	7.76 ± 0.02	7.74 ± 0.01
	GAR [MAE]	7.82 ± 0.02	7.791 ± 0.008
	GAR [MSE]	7.75 ± 0.02	7.739 ± 0.004
	MDN	7.71 ± 0.01	7.703 ± 0.002
	NLL	7.77 ± 0.01	7.744 ± 0.009
	beta-NLL	7.75 ± 0.01	7.734 ± 0.004
	AdaQR	7.74 ± 0.02	7.707 ± 0.008
	Quantile Loss	7.70 ± 0.01	7.671 ± 0.003
TabPack	MSE	7.514 ± 0.005	–
	HL-Gaussian	7.476 ± 0.006	–
	Quantile Loss	7.497 ± 0.007	–
	AdaQR	7.497 ± 0.007	–
RealMLP	MSE	7.416 ± 0.006	7.409 ± 0.002
	HL-Gaussian	7.395 ± 0.006	7.385 ± 0.003
	Quantile Loss	7.394 ± 0.004	7.381 ± 0.002
	AdaQR†	7.399 ± 0.006	7.381 ± 0.003
TabM	MSE	7.56 ± 0.01	7.540 ± 0.006
	HL-Gaussian	7.463 ± 0.009	7.444 ± 0.005
	Quantile Loss	7.512 ± 0.007	7.501 ± 0.002
	AdaQR	7.44 ± 0.01	7.428 ± 0.003

forest_fires

Model	Objective	Single	E5
MLP	MSE	1.37 ± 0.01	1.362 ± 0.002
	MAE	1.356 ± 0.009	1.344 ± 0.008
	Huber Loss	1.349 ± 0.001	1.3488 ± 0.0007
	Soft Binning	1.362 ± 0.002	1.3614 ± 0.0002
	Hard Binning	1.355 ± 0.002	1.3549 ± 0.0002
	Joint End-To-End	1.351 ± 0.002	1.3505 ± 0.0004
	Decoder[N]	1.7 ± 0.1	1.50 ± 0.02
	HL-Gaussian	1.353 ± 0.002	1.3531 ± 0.0004
	Decoder[U]	1.57 ± 0.06	1.45 ± 0.07
	Softmax MSE	1.3519 ± 0.0008	1.3516 ± 0.0000
	Ordinal Entropy	1.37 ± 0.02	1.359 ± 0.004
	ROT-REG	1.367 ± 0.009	1.364 ± 0.003
	GAR [MAE]	1.352 ± 0.004	1.350 ± 0.001
	GAR [MSE]	1.378 ± 0.009	1.3736 ± 0.0002
	MDN	1.349 ± 0.004	1.348 ± 0.001
	NLL	1.37 ± 0.01	1.358 ± 0.003
	beta-NLL	1.376 ± 0.008	1.374 ± 0.003
	AdaQR	1.3563 ± 0.0006	1.3563 ± 0.0002
	Quantile Loss	1.3586 ± 0.0008	1.3585 ± 0.0003
TabPack	MSE	1.362 ± 0.009	–
	HL-Gaussian	1.339 ± 0.009	–
	Quantile Loss	1.355 ± 0.001	–
	AdaQR	1.360 ± 0.003	–
RealMLP	MSE	1.36 ± 0.01	1.348 ± 0.002
	HL-Gaussian	1.359 ± 0.005	1.357 ± 0.001
	Quantile Loss	1.34 ± 0.01	1.34 ± 0.01
	AdaQR†	1.338 ± 0.006	1.337 ± 0.005
TabM	MSE	1.374 ± 0.009	1.373 ± 0.004
	HL-Gaussian	1.362 ± 0.005	1.3601 ± 0.0008
	Quantile Loss	1.355 ± 0.003	1.3545 ± 0.0009
	AdaQR	1.354 ± 0.001	1.3537 ± 0.0004

Table 19: Extended Beyond results with the Muon optimizer (continued).

garments_worker_productivity

Model	Objective	Single	E5
MLP	MSE	0.099 ± 0.004	0.097 ± 0.001
	MAE	0.101 ± 0.004	0.100 ± 0.002
	Huber Loss	0.099 ± 0.004	0.098 ± 0.002
	Soft Binning	0.102 ± 0.006	0.099 ± 0.001
	Hard Binning	0.105 ± 0.005	0.104 ± 0.004
	Joint End-To-End	0.099 ± 0.007	0.096 ± 0.002
	Decoder[N]	0.098 ± 0.010	0.092 ± 0.005
	HL-Gaussian	0.092 ± 0.002	0.0910 ± 0.0005
	Decoder[U]	0.11 ± 0.01	0.096 ± 0.005
	Softmax MSE	0.102 ± 0.007	0.101 ± 0.004
	Ordinal Entropy	0.096 ± 0.003	0.094 ± 0.002
	ROT-REG	–	–
	GAR [MAE]	0.096 ± 0.006	0.094 ± 0.003
	GAR [MSE]	0.103 ± 0.007	0.101 ± 0.003
	MDN	0.097 ± 0.005	0.095 ± 0.001
	NLL	0.102 ± 0.006	0.096 ± 0.005
	beta-NLL	0.098 ± 0.003	0.097 ± 0.002
	AdaQR	0.100 ± 0.005	0.0985 ± 0.0005
	Quantile Loss	0.100 ± 0.005	0.098 ± 0.002
TabPack	MSE	0.099 ± 0.003	–
	HL-Gaussian	0.095 ± 0.003	–
	Quantile Loss	0.093 ± 0.003	–
	AdaQR	0.094 ± 0.003	–
RealMLP	MSE	0.095 ± 0.004	0.094 ± 0.003
	HL-Gaussian	0.092 ± 0.009	0.089 ± 0.006
	Quantile Loss	0.087 ± 0.003	0.0856 ± 0.0009
	AdaQR†	0.089 ± 0.002	0.0884 ± 0.0007
TabM	MSE	0.099 ± 0.003	0.098 ± 0.001
	HL-Gaussian	0.088 ± 0.003	0.088 ± 0.002
	Quantile Loss	0.097 ± 0.003	0.096 ± 0.002
	AdaQR	0.098 ± 0.003	0.097 ± 0.002

healthcare_insurance_expenses

Model	Objective	Single	E5
MLP	MSE	0.401 ± 0.002	0.400 ± 0.001
	MAE	0.398 ± 0.004	0.394 ± 0.001
	Huber Loss	0.396 ± 0.005	0.393 ± 0.002
	Soft Binning	0.404 ± 0.002	0.4037 ± 0.0005
	Hard Binning	0.400 ± 0.005	0.3985 ± 0.0009
	Joint End-To-End	0.399 ± 0.002	0.398 ± 0.001
	Decoder[N]	0.400 ± 0.007	0.394 ± 0.002
	HL-Gaussian	0.400 ± 0.001	0.3995 ± 0.0001
	Decoder[U]	0.43 ± 0.02	0.415 ± 0.003
	Softmax MSE	0.396 ± 0.002	0.3955 ± 0.0007
	Ordinal Entropy	0.403 ± 0.004	0.4003 ± 0.0008
	ROT-REG	0.407 ± 0.006	0.404 ± 0.004
	GAR [MAE]	0.404 ± 0.006	0.4001 ± 0.0009
	GAR [MSE]	0.397 ± 0.002	0.3955 ± 0.0007
	MDN	0.391 ± 0.004	0.390 ± 0.001
	NLL	0.397 ± 0.003	0.395 ± 0.001
	beta-NLL	0.400 ± 0.003	0.395 ± 0.001
	AdaQR	0.391 ± 0.002	0.389 ± 0.001
	Quantile Loss	0.3912 ± 0.0010	0.3902 ± 0.0001
TabPack	MSE	0.395 ± 0.003	–
	HL-Gaussian	0.393 ± 0.005	–
	Quantile Loss	0.387 ± 0.004	–
	AdaQR	0.384 ± 0.003	–
RealMLP	MSE	0.404 ± 0.005	0.402 ± 0.002
	HL-Gaussian	0.391 ± 0.002	0.3898 ± 0.0006
	Quantile Loss	0.391 ± 0.006	0.389 ± 0.004
	AdaQR†	0.392 ± 0.007	0.389 ± 0.003
TabM	MSE	0.397 ± 0.005	0.394 ± 0.001
	HL-Gaussian	0.389 ± 0.004	0.388 ± 0.003
	Quantile Loss	0.379 ± 0.002	0.3787 ± 0.0006
	AdaQR	0.378 ± 0.001	0.3777 ± 0.0003

Table 20: Extended Beyond results with the Muon optimizer (continued).

houses

Model	Objective	Single	E5
MLP	MSE	0.2273 ± 0.0009	0.2226 ± 0.0001
	MAE	0.228 ± 0.001	0.2217 ± 0.0003
	Huber Loss	0.2242 ± 0.0007	0.2225 ± 0.0003
	Soft Binning	0.226 ± 0.001	0.2233 ± 0.0004
	Hard Binning	0.225 ± 0.001	0.2234 ± 0.0005
	Joint End-To-End	0.2250 ± 0.0010	0.2243 ± 0.0005
	Decoder[N]	0.234 ± 0.002	0.224 ± 0.001
	HL-Gaussian	0.2238 ± 0.0008	0.2226 ± 0.0003
	Decoder[U]	0.251 ± 0.004	0.2315 ± 0.0006
	Softmax MSE	0.230 ± 0.001	0.2228 ± 0.0005
	Ordinal Entropy	0.2266 ± 0.0008	0.2238 ± 0.0004
	ROT-REG	–	–
	GAR [MAE]	0.2241 ± 0.0009	0.2233 ± 0.0002
	GAR [MSE]	0.2242 ± 0.0005	0.2225 ± 0.0001
	MDN	0.2228 ± 0.0008	0.2215 ± 0.0003
	NLL	0.2257 ± 0.0010	0.2220 ± 0.0004
	beta-NLL	0.223 ± 0.001	0.2214 ± 0.0002
	AdaQR	0.2231 ± 0.0009	0.2213 ± 0.0003
	Quantile Loss	0.2237 ± 0.0009	0.2223 ± 0.0004
TabPack	MSE	0.2030 ± 0.0009	–
	HL-Gaussian	0.204 ± 0.002	–
	Quantile Loss	0.2015 ± 0.0010	–
	AdaQR	0.2015 ± 0.0009	–
RealMLP	MSE	0.1949 ± 0.0005	0.1926 ± 0.0003
	HL-Gaussian	0.195 ± 0.001	0.1929 ± 0.0002
	Quantile Loss	0.1917 ± 0.0005	0.1902 ± 0.0003
	AdaQR†	0.1958 ± 0.0009	0.1934 ± 0.0004
TabM	MSE	0.202 ± 0.001	0.1995 ± 0.0007
	HL-Gaussian	0.200 ± 0.002	0.197 ± 0.001
	Quantile Loss	0.200 ± 0.001	0.198 ± 0.001
	AdaQR	0.1994 ± 0.0008	0.1986 ± 0.0004

immoscout_german_house_prices

Model	Objective	Single	E5
MLP	MSE	0.407 ± 0.002	0.4018 ± 0.0010
	MAE	0.404 ± 0.002	0.399 ± 0.001
	Huber Loss	0.404 ± 0.003	0.398 ± 0.002
	Soft Binning	0.406 ± 0.002	0.4013 ± 0.0004
	Hard Binning	0.440 ± 0.003	0.428 ± 0.002
	Joint End-To-End	0.407 ± 0.002	0.403 ± 0.001
	Decoder[N]	0.423 ± 0.008	0.409 ± 0.001
	HL-Gaussian	0.409 ± 0.002	0.4041 ± 0.0002
	Decoder[U]	0.429 ± 0.005	0.409 ± 0.001
	Softmax MSE	0.409 ± 0.003	0.402 ± 0.001
	Ordinal Entropy	0.410 ± 0.003	0.4035 ± 0.0010
	ROT-REG	0.409 ± 0.003	0.402 ± 0.001
	GAR [MAE]	0.404 ± 0.002	0.400 ± 0.001
	GAR [MSE]	0.406 ± 0.003	0.4003 ± 0.0010
	MDN	0.401 ± 0.002	0.3976 ± 0.0007
	NLL	0.405 ± 0.003	0.4001 ± 0.0001
	beta-NLL	0.404 ± 0.002	0.3977 ± 0.0003
	AdaQR	0.405 ± 0.002	0.3987 ± 0.0003
	Quantile Loss	0.404 ± 0.002	0.3995 ± 0.0007
TabPack	MSE	0.369 ± 0.001	–
	HL-Gaussian	0.371 ± 0.001	–
	Quantile Loss	0.366 ± 0.001	–
	AdaQR	0.3656 ± 0.0009	–
RealMLP	MSE	0.368 ± 0.002	0.3650 ± 0.0006
	HL-Gaussian	0.368 ± 0.001	0.3629 ± 0.0007
	Quantile Loss	0.367 ± 0.001	0.3641 ± 0.0003
	AdaQR†	0.365 ± 0.001	0.3620 ± 0.0001
TabM	MSE	0.3726 ± 0.0007	0.3709 ± 0.0003
	HL-Gaussian	0.369 ± 0.002	0.3669 ± 0.0005
	Quantile Loss	0.371 ± 0.001	0.3694 ± 0.0006
	AdaQR	0.369 ± 0.001	0.3684 ± 0.0004

Table 21: Extended Beyond results with the Muon optimizer (continued).

mercari_price_suggestion_1m

Model	Objective	Single	E5
MLP	MSE	0.4746 ± 0.0006	0.4664 ± 0.0008
	MAE	0.479 ± 0.002	0.472 ± 0.002
	Huber Loss	0.4741 ± 0.0003	0.4666 ± 0.0003
	Soft Binning	0.472 ± 0.002	0.463 ± 0.001
	Hard Binning	0.4742 ± 0.0008	0.4649 ± 0.0009
	Joint End-To-End	0.4770 ± 0.0005	0.4694 ± 0.0004
	Decoder[N]	0.507 ± 0.005	0.486 ± 0.001
	HL-Gaussian	0.4730 ± 0.0002	0.4625 ± 0.0005
	Decoder[U]	0.513 ± 0.005	0.477 ± 0.003
	Softmax MSE	0.4764 ± 0.0007	0.4696 ± 0.0002
	Ordinal Entropy	0.474 ± 0.001	0.4675 ± 0.0007
	ROT-REG	0.4782 ± 0.0002	0.4705 ± 0.0001
	GAR [MAE]	0.476 ± 0.002	0.4668 ± 0.0006
	GAR [MSE]	0.4748 ± 0.0003	0.4687 ± 0.0001
	MDN	0.474 ± 0.002	0.4661 ± 0.0010
	NLL	0.4767 ± 0.0005	0.4679 ± 0.0004
	beta-NLL	0.476 ± 0.001	0.4685 ± 0.0005
	AdaQR	0.4723 ± 0.0007	0.464 ± 0.001
	Quantile Loss	0.473 ± 0.003	0.464 ± 0.001
TabPack	MSE	0.4630 ± 0.0002	–
	HL-Gaussian	0.4652 ± 0.0001	–
	Quantile Loss	0.4621 ± 0.0001	–
	AdaQR	0.4621 ± 0.0002	–
RealMLP	MSE	0.4633 ± 0.0002	0.4577 ± 0.0002
	HL-Gaussian	0.4572 ± 0.0003	0.4512 ± 0.0004
	Quantile Loss	0.4584 ± 0.0003	0.4534 ± 0.0002
	AdaQR†	0.4617 ± 0.0002	0.4560 ± 0.0002
TabM	MSE	0.4682 ± 0.0004	0.4654 ± 0.0002
	HL-Gaussian	0.4647 ± 0.0005	0.4616 ± 0.0002
	Quantile Loss	0.4676 ± 0.0003	0.4649 ± 0.0002
	AdaQR	0.4703 ± 0.0007	0.4681 ± 0.0002

mercedes_benz_greener_manufacturing

Model	Objective	Single	E5
MLP	MSE	8.6 ± 0.1	8.570 ± 0.007
	MAE	8.38 ± 0.04	8.34 ± 0.02
	Huber Loss	8.40 ± 0.05	8.37 ± 0.01
	Soft Binning	9.4 ± 0.5	9.0 ± 0.1
	Hard Binning	8.51 ± 0.08	8.46 ± 0.02
	Joint End-To-End	8.45 ± 0.06	8.365 ± 0.002
	Decoder[N]	8.7 ± 0.2	8.47 ± 0.03
	HL-Gaussian	8.68 ± 0.10	8.61 ± 0.04
	Decoder[U]	8.8 ± 0.2	8.54 ± 0.10
	Softmax MSE	8.62 ± 0.10	8.50 ± 0.03
	Ordinal Entropy	8.57 ± 0.09	8.54 ± 0.05
	ROT-REG	8.5 ± 0.1	8.43 ± 0.02
	GAR [MAE]	8.34 ± 0.04	8.32 ± 0.02
	GAR [MSE]	8.55 ± 0.09	8.50 ± 0.09
	MDN	8.5 ± 0.1	8.41 ± 0.08
	NLL	8.6 ± 0.1	8.51 ± 0.08
	beta-NLL	8.8 ± 0.2	8.67 ± 0.03
	AdaQR	8.36 ± 0.04	8.35 ± 0.03
	Quantile Loss	8.41 ± 0.05	8.395 ± 0.006
TabPack	MSE	8.48 ± 0.06	–
	HL-Gaussian	8.48 ± 0.09	–
	Quantile Loss	8.46 ± 0.07	–
	AdaQR	8.46 ± 0.07	–
RealMLP	MSE	8.31 ± 0.02	8.303 ± 0.007
	HL-Gaussian	8.35 ± 0.04	8.339 ± 0.008
	Quantile Loss	8.22 ± 0.01	8.217 ± 0.004
	AdaQR†	8.25 ± 0.03	8.24 ± 0.01
TabM	MSE	8.34 ± 0.05	8.32 ± 0.03
	HL-Gaussian	8.33 ± 0.04	8.32 ± 0.01
	Quantile Loss	8.24 ± 0.02	8.23 ± 0.02
	AdaQR	8.28 ± 0.03	8.27 ± 0.01

Table 22: Extended Beyond results with the Muon optimizer (continued).

miami_housing

Model	Objective	Single	E5
MLP	MSE	0.1446 ± 0.0008	0.1427 ± 0.0002
	MAE	0.148 ± 0.001	0.1454 ± 0.0004
	Huber Loss	0.1450 ± 0.0005	0.1438 ± 0.0002
	Soft Binning	0.150 ± 0.002	0.1465 ± 0.0004
	Hard Binning	0.1486 ± 0.0005	0.1480 ± 0.0002
	Joint End-To-End	0.147 ± 0.001	0.1441 ± 0.0006
	Decoder[N]	0.155 ± 0.002	0.149 ± 0.001
	HL-Gaussian	0.144 ± 0.001	0.1423 ± 0.0004
	Decoder[U]	0.155 ± 0.002	0.1488 ± 0.0004
	Softmax MSE	0.1434 ± 0.0007	0.1424 ± 0.0000
	Ordinal Entropy	0.1428 ± 0.0007	0.1420 ± 0.0002
	ROT-REG	0.1432 ± 0.0007	0.1425 ± 0.0004
	GAR [MAE]	0.1441 ± 0.0007	0.1430 ± 0.0005
	GAR [MSE]	0.1435 ± 0.0008	0.1427 ± 0.0006
	MDN	0.1458 ± 0.0010	0.1450 ± 0.0003
	NLL	0.1466 ± 0.0008	0.1442 ± 0.0003
	beta-NLL	0.1441 ± 0.0007	0.1428 ± 0.0000
	AdaQR	0.1430 ± 0.0005	0.1422 ± 0.0004
	Quantile Loss	0.144 ± 0.001	0.1431 ± 0.0005
TabPack	MSE	0.1323 ± 0.0004	–
	HL-Gaussian	0.1323 ± 0.0003	–
	Quantile Loss	0.1320 ± 0.0004	–
	AdaQR	0.1321 ± 0.0004	–
RealMLP	MSE	0.1348 ± 0.0004	0.1336 ± 0.0001
	HL-Gaussian	0.1331 ± 0.0007	0.1315 ± 0.0004
	Quantile Loss	0.1343 ± 0.0007	0.1325 ± 0.0001
	AdaQR†	0.1322 ± 0.0005	0.1316 ± 0.0002
TabM	MSE	0.1335 ± 0.0006	0.1329 ± 0.0001
	HL-Gaussian	0.1341 ± 0.0006	0.1336 ± 0.0001
	Quantile Loss	0.133 ± 0.001	0.1321 ± 0.0003
	AdaQR	0.1325 ± 0.0006	0.1319 ± 0.0003

mutual_funds_india

Model	Objective	Single	E5
MLP	MSE	5.9 ± 0.2	5.66 ± 0.03
	MAE	5.9 ± 0.1	5.65 ± 0.04
	Huber Loss	5.9 ± 0.1	5.77 ± 0.04
	Soft Binning	5.64 ± 0.10	5.59 ± 0.05
	Hard Binning	6.2 ± 0.1	6.09 ± 0.04
	Joint End-To-End	5.6 ± 0.1	5.56 ± 0.05
	Decoder[N]	6.5 ± 0.2	6.26 ± 0.07
	HL-Gaussian	5.56 ± 0.08	5.49 ± 0.05
	Decoder[U]	7.3 ± 0.4	6.39 ± 0.06
	Softmax MSE	6.0 ± 0.1	5.88 ± 0.04
	Ordinal Entropy	6.0 ± 0.2	5.72 ± 0.02
	ROT-REG	5.9 ± 0.1	5.71 ± 0.06
	GAR [MAE]	5.79 ± 0.09	5.68 ± 0.06
	GAR [MSE]	5.9 ± 0.1	5.79 ± 0.05
	MDN	5.8 ± 0.1	5.73 ± 0.04
	NLL	5.89 ± 0.09	5.776 ± 0.008
	beta-NLL	5.9 ± 0.1	5.76 ± 0.09
	AdaQR	5.84 ± 0.07	5.719 ± 0.008
	Quantile Loss	5.90 ± 0.09	5.76 ± 0.03
TabPack	MSE	5.07 ± 0.08	–
	HL-Gaussian	5.24 ± 0.08	–
	Quantile Loss	4.97 ± 0.07	–
	AdaQR	5.08 ± 0.08	–
RealMLP	MSE	6 ± 2	5.3 ± 0.2
	HL-Gaussian	5.17 ± 0.06	5.14 ± 0.04
	Quantile Loss	5.11 ± 0.05	5.09 ± 0.02
	AdaQR†	5.16 ± 0.07	5.100 ± 0.008
TabM	MSE	5.16 ± 0.07	5.12 ± 0.02
	HL-Gaussian	5.2 ± 0.1	5.19 ± 0.06
	Quantile Loss	5.0 ± 0.1	4.94 ± 0.06
	AdaQR	4.95 ± 0.06	4.90 ± 0.05

Table 23: Extended Beyond results with the Muon optimizer (continued).

obesity_estimation

Model	Objective	Single	E5
MLP	MSE	4.91 ± 0.03	4.904 ± 0.002
	MAE	4.85 ± 0.02	4.84 ± 0.01
	Huber Loss	4.94 ± 0.05	4.92 ± 0.03
	Soft Binning	4.91 ± 0.04	4.90 ± 0.02
	Hard Binning	4.96 ± 0.05	4.944 ± 0.006
	Joint End-To-End	4.96 ± 0.02	4.94 ± 0.02
	Decoder[N]	5.1 ± 0.2	4.85 ± 0.03
	HL-Gaussian	4.90 ± 0.03	4.895 ± 0.006
	Decoder[U]	5.3 ± 0.1	5.17 ± 0.08
	Softmax MSE	4.95 ± 0.04	4.93 ± 0.02
	Ordinal Entropy	4.89 ± 0.03	4.88 ± 0.02
	ROT-REG	4.94 ± 0.01	4.926 ± 0.005
	GAR [MAE]	4.86 ± 0.02	4.845 ± 0.010
	GAR [MSE]	4.92 ± 0.03	4.912 ± 0.010
	MDN	4.89 ± 0.02	4.877 ± 0.005
	NLL	4.88 ± 0.02	4.870 ± 0.010
	beta-NLL	4.91 ± 0.02	4.904 ± 0.005
	AdaQR	4.90 ± 0.03	4.89 ± 0.01
	Quantile Loss	4.86 ± 0.02	4.854 ± 0.005
TabPack	MSE	4.95 ± 0.05	–
	HL-Gaussian	4.93 ± 0.05	–
	Quantile Loss	4.88 ± 0.03	–
	AdaQR	4.89 ± 0.03	–
RealMLP	MSE	4.91 ± 0.06	4.84 ± 0.02
	HL-Gaussian	5.0 ± 0.1	4.95 ± 0.04
	Quantile Loss	4.89 ± 0.04	4.87 ± 0.02
	AdaQR†	4.825 ± 0.007	4.823 ± 0.002
TabM	MSE	4.90 ± 0.03	4.88 ± 0.02
	HL-Gaussian	4.88 ± 0.02	4.869 ± 0.006
	Quantile Loss	4.87 ± 0.02	4.87 ± 0.01
	AdaQR	4.89 ± 0.04	4.872 ± 0.007

physiochemical_protein

Model	Objective	Single	E5
MLP	MSE	3.38 ± 0.01	3.342 ± 0.003
	MAE	3.43 ± 0.02	3.36 ± 0.02
	Huber Loss	3.38 ± 0.01	3.359 ± 0.007
	Soft Binning	3.33 ± 0.02	3.297 ± 0.006
	Hard Binning	3.36 ± 0.01	3.313 ± 0.005
	Joint End-To-End	3.37 ± 0.02	3.317 ± 0.008
	Decoder[N]	3.9 ± 0.1	3.50 ± 0.02
	HL-Gaussian	3.35 ± 0.01	3.31 ± 0.01
	Decoder[U]	4.0 ± 0.3	3.50 ± 0.05
	Softmax MSE	3.37 ± 0.01	3.343 ± 0.004
	Ordinal Entropy	3.36 ± 0.01	3.334 ± 0.006
	ROT-REG	3.36 ± 0.01	3.342 ± 0.009
	GAR [MAE]	3.6 ± 0.7	3.34 ± 0.07
	GAR [MSE]	3.39 ± 0.02	3.338 ± 0.003
	MDN	3.366 ± 0.009	3.322 ± 0.004
	NLL	3.47 ± 0.02	3.35 ± 0.01
	beta-NLL	3.39 ± 0.02	3.337 ± 0.009
	AdaQR	3.37 ± 0.02	3.340 ± 0.005
	Quantile Loss	3.39 ± 0.01	3.369 ± 0.002
TabPack	MSE	3.24 ± 0.01	–
	HL-Gaussian	3.278 ± 0.010	–
	Quantile Loss	3.241 ± 0.008	–
	AdaQR	3.242 ± 0.010	–
RealMLP	MSE	3.168 ± 0.009	3.124 ± 0.003
	HL-Gaussian	3.17 ± 0.01	3.107 ± 0.001
	Quantile Loss	3.16 ± 0.01	3.126 ± 0.003
	AdaQR†	3.165 ± 0.005	3.1342 ± 0.0007
TabM	MSE	3.338 ± 0.010	3.312 ± 0.004
	HL-Gaussian	3.30 ± 0.04	–
	Quantile Loss	3.39 ± 0.06	3.33 ± 0.01
	AdaQR	3.35 ± 0.05	3.27 ± 0.01

Table 24: Extended Beyond results with the Muon optimizer (continued).

qsar_aquatic_toxicity

Model	Objective	Single	E5
MLP	MSE	1.17 ± 0.02	1.16 ± 0.01
	MAE	1.28 ± 0.03	1.256 ± 0.008
	Huber Loss	1.16 ± 0.02	1.156 ± 0.004
	Soft Binning	1.14 ± 0.02	1.140 ± 0.007
	Hard Binning	1.22 ± 0.04	1.211 ± 0.009
	Joint End-To-End	1.20 ± 0.03	1.179 ± 0.006
	Decoder[N]	1.41 ± 0.06	1.33 ± 0.05
	HL-Gaussian	1.17 ± 0.01	1.171 ± 0.005
	Decoder[U]	1.28 ± 0.06	1.15 ± 0.06
	Softmax MSE	1.19 ± 0.03	1.177 ± 0.009
	Ordinal Entropy	1.17 ± 0.02	1.16 ± 0.01
	ROT-REG	1.17 ± 0.02	1.16 ± 0.01
	GAR [MAE]	1.19 ± 0.04	1.18 ± 0.02
	GAR [MSE]	1.19 ± 0.01	1.182 ± 0.004
	MDN	1.23 ± 0.02	1.223 ± 0.004
	NLL	1.16 ± 0.02	1.144 ± 0.004
	beta-NLL	1.18 ± 0.01	1.179 ± 0.004
	AdaQR	1.18 ± 0.02	1.17 ± 0.01
	Quantile Loss	1.19 ± 0.01	1.19 ± 0.01
TabPack	MSE	1.18 ± 0.02	–
	HL-Gaussian	1.19 ± 0.03	–
	Quantile Loss	1.14 ± 0.02	–
	AdaQR	1.15 ± 0.03	–
RealMLP	MSE	1.15 ± 0.02	1.13 ± 0.01
	HL-Gaussian	1.10 ± 0.01	1.099 ± 0.001
	Quantile Loss	1.18 ± 0.02	1.178 ± 0.009
	AdaQR†	1.23 ± 0.03	1.206 ± 0.005
TabM	MSE	1.16 ± 0.05	1.12 ± 0.03
	HL-Gaussian	1.13 ± 0.03	1.13 ± 0.02
	Quantile Loss	1.16 ± 0.03	1.148 ± 0.007
	AdaQR	1.19 ± 0.03	1.172 ± 0.004

qsar_fish_toxicity

Model	Objective	Single	E5
MLP	MSE	0.93 ± 0.01	0.918 ± 0.007
	MAE	0.950 ± 0.006	0.949 ± 0.002
	Huber Loss	0.951 ± 0.008	0.950 ± 0.004
	Soft Binning	0.870 ± 0.005	0.864 ± 0.003
	Hard Binning	1.01 ± 0.04	0.92 ± 0.02
	Joint End-To-End	0.924 ± 0.007	0.921 ± 0.005
	Decoder[N]	0.97 ± 0.01	0.942 ± 0.002
	HL-Gaussian	0.889 ± 0.004	0.888 ± 0.002
	Decoder[U]	0.96 ± 0.02	0.905 ± 0.009
	Softmax MSE	0.91 ± 0.01	0.901 ± 0.006
	Ordinal Entropy	0.93 ± 0.01	0.924 ± 0.007
	ROT-REG	0.94 ± 0.02	0.934 ± 0.007
	GAR [MAE]	0.939 ± 0.009	0.937 ± 0.006
	GAR [MSE]	0.95 ± 0.02	0.941 ± 0.006
	MDN	0.99 ± 0.05	0.96 ± 0.01
	NLL	0.98 ± 0.03	0.95 ± 0.02
	beta-NLL	0.943 ± 0.007	0.941 ± 0.001
	AdaQR	0.97 ± 0.01	0.962 ± 0.005
	Quantile Loss	0.98 ± 0.01	0.965 ± 0.008
TabPack	MSE	0.93 ± 0.02	–
	HL-Gaussian	0.88 ± 0.02	–
	Quantile Loss	0.92 ± 0.01	–
	AdaQR	0.93 ± 0.02	–
RealMLP	MSE	0.95 ± 0.01	0.94 ± 0.01
	HL-Gaussian	0.91 ± 0.02	0.895 ± 0.007
	Quantile Loss	0.95 ± 0.02	0.930 ± 0.008
	AdaQR†	0.908 ± 0.009	0.897 ± 0.006
TabM	MSE	0.95 ± 0.01	0.937 ± 0.007
	HL-Gaussian	0.94 ± 0.01	0.938 ± 0.003
	Quantile Loss	0.889 ± 0.008	0.886 ± 0.006
	AdaQR	0.95 ± 0.02	0.933 ± 0.003

Table 25: Extended Beyond results with the Muon optimizer (continued).

qsar_tid_11

Model	Objective	Single	E5
MLP	MSE	0.784 ± 0.005	0.772 ± 0.001
	MAE	0.783 ± 0.005	0.7720 ± 0.0009
	Huber Loss	0.786 ± 0.006	0.775 ± 0.003
	Soft Binning	0.787 ± 0.004	0.782 ± 0.002
	Hard Binning	0.84 ± 0.01	0.819 ± 0.008
	Joint End-To-End	0.786 ± 0.003	0.7814 ± 0.0005
	Decoder[N]	0.83 ± 0.01	0.8022 ± 0.0008
	HL-Gaussian	0.784 ± 0.004	0.779 ± 0.003
	Decoder[U]	0.87 ± 0.01	0.810 ± 0.004
	Softmax MSE	0.796 ± 0.006	0.784 ± 0.003
	Ordinal Entropy	0.791 ± 0.008	0.772 ± 0.002
	ROT-REG	0.783 ± 0.006	0.772 ± 0.002
	GAR [MAE]	0.781 ± 0.005	0.770 ± 0.002
	GAR [MSE]	0.792 ± 0.004	0.7810 ± 0.0009
	MDN	0.783 ± 0.005	0.771 ± 0.002
	NLL	0.784 ± 0.005	0.775 ± 0.001
	beta-NLL	0.790 ± 0.006	0.7786 ± 0.0007
	AdaQR	0.790 ± 0.006	0.775 ± 0.004
	Quantile Loss	0.781 ± 0.005	0.771 ± 0.001
TabPack	MSE	0.749 ± 0.004	–
	HL-Gaussian	0.747 ± 0.005	–
	Quantile Loss	0.741 ± 0.003	–
	AdaQR	0.739 ± 0.003	–
RealMLP	MSE	0.757 ± 0.002	0.751 ± 0.001
	HL-Gaussian	0.752 ± 0.003	0.7480 ± 0.0009
	Quantile Loss	0.759 ± 0.004	0.7520 ± 0.0007
	AdaQR†	0.750 ± 0.002	0.7468 ± 0.0005
TabM	MSE	0.772 ± 0.004	0.7645 ± 0.0004
	HL-Gaussian	0.782 ± 0.006	0.768 ± 0.003
	Quantile Loss	0.769 ± 0.006	0.759 ± 0.001
	AdaQR	0.775 ± 0.006	0.7604 ± 0.0008

rossmann_store_sales

Model	Objective	Single	E5
MLP	MSE	1340 ± 125	1296 ± 55
	MAE	1362 ± 85	1333 ± 42
	Huber Loss	1315 ± 67	1292 ± 41
	Soft Binning	1172 ± 95	1115 ± 19
	Hard Binning	1195 ± 65	1175 ± 45
	Joint End-To-End	1256 ± 31	1218 ± 25
	Decoder[N]	1289 ± 83	1164 ± 8
	HL-Gaussian	1218 ± 133	1142 ± 30
	Decoder[U]	1252 ± 84	1144 ± 41
	Softmax MSE	1270 ± 82	1245 ± 43
	Ordinal Entropy	1319 ± 67	1301 ± 34
	ROT-REG	1291 ± 68	1274 ± 63
	GAR [MAE]	1283 ± 44	1265 ± 16
	GAR [MSE]	1293 ± 52	1284 ± 27
	MDN	1586 ± 224	1496 ± 115
	NLL	1360 ± 87	1327 ± 40
	beta-NLL	1248 ± 19	1234 ± 4
	AdaQR	1457 ± 130	1228 ± 32
	Quantile Loss	1354 ± 56	1331 ± 41
TabPack	MSE	994 ± 23	–
	HL-Gaussian	1011 ± 19	–
	Quantile Loss	997 ± 39	–
	AdaQR	1009 ± 30	–
RealMLP	MSE	924 ± 13	913 ± 4
	HL-Gaussian	1040 ± 8	1023 ± 4
	Quantile Loss	961 ± 20	947 ± 2
	AdaQR†	984 ± 19	964 ± 9
TabM	MSE	1149 ± 164	1045 ± 34
	HL-Gaussian	1204 ± 52	1180 ± 32
	Quantile Loss	1084 ± 59	1009 ± 23
	AdaQR	1036 ± 55	969 ± 28

Table 26: Extended Beyond results with the Muon optimizer (continued).

santander_transaction_value

Model	Objective	Single	E5
MLP	MSE	1.409 ± 0.006	1.403 ± 0.002
	MAE	1.408 ± 0.007	1.396 ± 0.003
	Huber Loss	1.403 ± 0.009	1.394 ± 0.005
	Soft Binning	1.399 ± 0.009	1.389 ± 0.003
	Hard Binning	1.415 ± 0.006	1.409 ± 0.003
	Joint End-To-End	1.412 ± 0.010	1.397 ± 0.009
	Decoder[N]	1.47 ± 0.01	1.415 ± 0.008
	HL-Gaussian	1.406 ± 0.007	1.397 ± 0.005
	Decoder[U]	1.49 ± 0.03	1.412 ± 0.006
	Softmax MSE	1.415 ± 0.006	1.407 ± 0.002
	Ordinal Entropy	1.410 ± 0.006	1.404 ± 0.004
	ROT-REG	1.407 ± 0.006	1.401 ± 0.003
	GAR [MAE]	1.393 ± 0.003	1.3912 ± 0.0009
	GAR [MSE]	1.413 ± 0.007	1.404 ± 0.004
	MDN	1.408 ± 0.009	1.395 ± 0.006
	NLL	1.390 ± 0.006	1.3881 ± 0.0003
	beta-NLL	1.400 ± 0.007	1.392 ± 0.002
	AdaQR	1.395 ± 0.004	1.391 ± 0.003
	Quantile Loss	1.393 ± 0.005	1.388 ± 0.001
TabPack	MSE	1.361 ± 0.004	–
	HL-Gaussian	1.359 ± 0.007	–
	Quantile Loss	1.359 ± 0.003	–
	AdaQR	1.361 ± 0.005	–
RealMLP	MSE	1.377 ± 0.006	1.369 ± 0.003
	HL-Gaussian	1.367 ± 0.004	1.362 ± 0.001
	Quantile Loss	1.364 ± 0.003	1.3628 ± 0.0008
	AdaQR†	1.369 ± 0.005	1.367 ± 0.001
TabM	MSE	1.379 ± 0.007	1.3743 ± 0.0007
	HL-Gaussian	1.380 ± 0.007	1.373 ± 0.006
	Quantile Loss	1.38 ± 0.01	1.370 ± 0.002
	AdaQR	1.38 ± 0.01	1.373 ± 0.007

sat11_hand_algo_runtime

Model	Objective	Single	E5
MLP	MSE	2.02 ± 0.07	1.991 ± 0.007
	MAE	2.01 ± 0.04	1.98 ± 0.02
	Huber Loss	2.04 ± 0.04	2.009 ± 0.007
	Soft Binning	1.96 ± 0.03	1.946 ± 0.006
	Hard Binning	2.02 ± 0.07	1.91 ± 0.02
	Joint End-To-End	2.01 ± 0.03	1.977 ± 0.004
	Decoder[N]	2.2 ± 0.1	1.98 ± 0.05
	HL-Gaussian	1.93 ± 0.04	1.88 ± 0.01
	Decoder[U]	2.2 ± 0.2	1.97 ± 0.10
	Softmax MSE	2.10 ± 0.06	2.02 ± 0.03
	Ordinal Entropy	2.00 ± 0.06	1.96 ± 0.01
	ROT-REG	1.96 ± 0.03	1.94 ± 0.01
	GAR [MAE]	1.98 ± 0.03	1.95 ± 0.01
	GAR [MSE]	2.01 ± 0.04	1.982 ± 0.005
	MDN	2.01 ± 0.06	1.943 ± 0.008
	NLL	1.93 ± 0.03	1.909 ± 0.007
	beta-NLL	1.99 ± 0.05	1.971 ± 0.009
	AdaQR	1.96 ± 0.04	1.945 ± 0.005
	Quantile Loss	1.94 ± 0.03	1.92 ± 0.01
TabPack	MSE	1.83 ± 0.04	–
	HL-Gaussian	1.89 ± 0.04	–
	Quantile Loss	1.85 ± 0.04	–
	AdaQR	1.84 ± 0.02	–
RealMLP	MSE	1.96 ± 0.04	1.95 ± 0.02
	HL-Gaussian	1.97 ± 0.03	1.93 ± 0.02
	Quantile Loss	1.83 ± 0.02	1.820 ± 0.010
	AdaQR†	1.91 ± 0.02	1.898 ± 0.006
TabM	MSE	1.90 ± 0.02	1.887 ± 0.005
	HL-Gaussian	1.87 ± 0.03	1.858 ± 0.006
	Quantile Loss	1.93 ± 0.02	1.922 ± 0.007
	AdaQR	1.92 ± 0.03	1.89 ± 0.03

Table 27: Extended Beyond results with the Muon optimizer (continued).

sberbank_housing_market_forecasting

Model	Objective	Single	E5
MLP	MSE	0.27 ± 0.02	0.26 ± 0.02
	MAE	0.263 ± 0.010	0.256 ± 0.004
	Huber Loss	0.26 ± 0.01	0.254 ± 0.006
	Soft Binning	0.263 ± 0.009	0.2521 ± 0.0010
	Hard Binning	0.254 ± 0.003	0.2498 ± 0.0003
	Joint End-To-End	0.255 ± 0.005	0.248 ± 0.002
	Decoder[N]	0.264 ± 0.005	0.256 ± 0.003
	HL-Gaussian	0.27 ± 0.03	0.253 ± 0.002
	Decoder[U]	0.28 ± 0.02	0.252 ± 0.003
	Softmax MSE	0.261 ± 0.006	0.257 ± 0.005
	Ordinal Entropy	0.261 ± 0.005	0.2555 ± 0.0002
	ROT-REG	0.260 ± 0.007	0.255 ± 0.002
	GAR [MAE]	0.263 ± 0.009	0.257 ± 0.002
	GAR [MSE]	0.26 ± 0.01	0.257 ± 0.002
	MDN	0.29 ± 0.02	0.2617 ± 0.0009
	NLL	0.257 ± 0.004	0.2552 ± 0.0006
	beta-NLL	0.256 ± 0.003	0.251 ± 0.001
	AdaQR	0.254 ± 0.003	0.2489 ± 0.0004
	Quantile Loss	0.28 ± 0.03	0.27 ± 0.02
TabPack	MSE	0.238 ± 0.002	–
	HL-Gaussian	0.237 ± 0.001	–
	Quantile Loss	0.2341 ± 0.0009	–
	AdaQR	0.234 ± 0.001	–
RealMLP	MSE	0.2390 ± 0.0009	0.2380 ± 0.0005
	HL-Gaussian	0.240 ± 0.002	0.2382 ± 0.0003
	Quantile Loss	0.2324 ± 0.0008	0.2318 ± 0.0006
	AdaQR†	0.236 ± 0.002	0.235 ± 0.001
TabM	MSE	0.260 ± 0.003	0.259 ± 0.002
	HL-Gaussian	0.241 ± 0.002	0.2393 ± 0.0003
	Quantile Loss	0.236 ± 0.001	0.2350 ± 0.0005
	AdaQR	0.240 ± 0.004	0.238 ± 0.001

sf_permit_time

Model	Objective	Single	E5
MLP	MSE	1.46 ± 0.01	1.455 ± 0.003
	MAE	1.47 ± 0.01	1.468 ± 0.010
	Huber Loss	1.51 ± 0.02	1.492 ± 0.003
	Soft Binning	1.469 ± 0.007	1.467 ± 0.005
	Hard Binning	1.493 ± 0.010	1.489 ± 0.002
	Joint End-To-End	1.48 ± 0.01	1.475 ± 0.005
	Decoder[N]	1.57 ± 0.05	1.54 ± 0.05
	HL-Gaussian	1.46 ± 0.01	1.455 ± 0.003
	Decoder[U]	1.53 ± 0.05	1.50 ± 0.02
	Softmax MSE	1.49 ± 0.01	1.486 ± 0.008
	Ordinal Entropy	1.48 ± 0.02	1.465 ± 0.004
	ROT-REG	1.44 ± 0.01	1.434 ± 0.005
	GAR [MAE]	1.46 ± 0.01	1.461 ± 0.003
	GAR [MSE]	1.494 ± 0.009	1.492 ± 0.002
	MDN	1.46 ± 0.02	1.450 ± 0.007
	NLL	1.465 ± 0.005	1.464 ± 0.003
	beta-NLL	1.47 ± 0.01	1.471 ± 0.002
	AdaQR	1.45 ± 0.02	1.441 ± 0.006
	Quantile Loss	1.460 ± 0.008	1.459 ± 0.002
TabPack	MSE	1.448 ± 0.006	–
	HL-Gaussian	1.448 ± 0.004	–
	Quantile Loss	1.447 ± 0.005	–
	AdaQR	1.449 ± 0.009	–
RealMLP	MSE	1.47 ± 0.01	1.465 ± 0.006
	HL-Gaussian	1.47 ± 0.01	1.472 ± 0.004
	Quantile Loss	1.457 ± 0.010	1.454 ± 0.005
	AdaQR†	1.46 ± 0.01	1.454 ± 0.006
TabM	MSE	1.47 ± 0.02	1.46 ± 0.01
	HL-Gaussian	1.48 ± 0.01	1.480 ± 0.005
	Quantile Loss	1.47 ± 0.03	1.46 ± 0.02
	AdaQR	1.48 ± 0.02	1.474 ± 0.007

Table 28: Extended Beyond results with the Muon optimizer (continued).

student_portuguese_performance

Model	Objective	Single	E5
MLP	MSE	3.09 ± 0.09	3.00 ± 0.03
	MAE	3.03 ± 0.08	2.96 ± 0.03
	Huber Loss	3.04 ± 0.08	2.99 ± 0.03
	Soft Binning	2.99 ± 0.09	2.92 ± 0.03
	Hard Binning	3.00 ± 0.02	2.979 ± 0.002
	Joint End-To-End	2.97 ± 0.04	2.94 ± 0.02
	Decoder[N]	3.17 ± 0.09	2.997 ± 0.008
	HL-Gaussian	2.99 ± 0.04	2.96 ± 0.01
	Decoder[U]	3.2 ± 0.2	3.2 ± 0.3
	Softmax MSE	2.96 ± 0.04	2.92 ± 0.02
	Ordinal Entropy	3.11 ± 0.06	3.06 ± 0.01
	ROT-REG	3.08 ± 0.08	2.99 ± 0.02
	GAR [MAE]	3.03 ± 0.07	2.94 ± 0.02
	GAR [MSE]	3.01 ± 0.06	2.92 ± 0.03
	MDN	2.99 ± 0.07	2.90 ± 0.02
	NLL	3.03 ± 0.08	2.96 ± 0.06
	beta-NLL	3.03 ± 0.06	2.98 ± 0.03
	AdaQR	3.02 ± 0.08	2.96 ± 0.03
	Quantile Loss	2.94 ± 0.04	2.90 ± 0.01
TabPack	MSE	2.82 ± 0.03	–
	HL-Gaussian	2.81 ± 0.05	–
	Quantile Loss	2.84 ± 0.03	–
	AdaQR	2.81 ± 0.04	–
RealMLP	MSE	2.92 ± 0.07	2.821 ± 0.008
	HL-Gaussian	2.87 ± 0.04	2.849 ± 0.005
	Quantile Loss	2.84 ± 0.02	2.83 ± 0.01
	AdaQR†	2.79 ± 0.03	2.77 ± 0.01
TabM	MSE	2.92 ± 0.03	2.91 ± 0.01
	HL-Gaussian	2.93 ± 0.02	2.92 ± 0.01
	Quantile Loss	2.91 ± 0.01	2.889 ± 0.002
	AdaQR	2.92 ± 0.02	2.90 ± 0.01

superconductivity

Model	Objective	Single	E5
MLP	MSE	9.76 ± 0.04	9.643 ± 0.006
	MAE	9.88 ± 0.08	9.74 ± 0.05
	Huber Loss	9.88 ± 0.06	9.73 ± 0.02
	Soft Binning	9.77 ± 0.04	9.67 ± 0.01
	Hard Binning	10.06 ± 0.09	9.86 ± 0.01
	Joint End-To-End	9.73 ± 0.03	9.64 ± 0.01
	Decoder[N]	10.8 ± 0.3	10.2 ± 0.1
	HL-Gaussian	9.79 ± 0.05	9.705 ± 0.008
	Decoder[U]	12 ± 1	10.3 ± 0.1
	Softmax MSE	9.77 ± 0.03	9.70 ± 0.02
	Ordinal Entropy	9.73 ± 0.04	9.63 ± 0.02
	ROT-REG	9.79 ± 0.04	9.67 ± 0.02
	GAR [MAE]	9.92 ± 0.07	9.77 ± 0.04
	GAR [MSE]	9.78 ± 0.03	9.66 ± 0.01
	MDN	9.89 ± 0.08	9.64 ± 0.01
	NLL	9.9 ± 0.1	9.67 ± 0.04
	beta-NLL	9.75 ± 0.04	9.69 ± 0.03
	AdaQR	9.75 ± 0.05	9.67 ± 0.02
	Quantile Loss	9.74 ± 0.04	9.594 ± 0.005
TabPack	MSE	9.47 ± 0.02	–
	HL-Gaussian	9.50 ± 0.03	–
	Quantile Loss	9.44 ± 0.03	–
	AdaQR	9.42 ± 0.04	–
RealMLP	MSE	9.64 ± 0.02	9.607 ± 0.006
	HL-Gaussian	9.54 ± 0.02	9.477 ± 0.003
	Quantile Loss	9.62 ± 0.03	9.59 ± 0.02
	AdaQR†	9.51 ± 0.03	9.462 ± 0.010
TabM	MSE	9.71 ± 0.07	9.642 ± 0.006
	HL-Gaussian	9.65 ± 0.03	9.588 ± 0.003
	Quantile Loss	9.62 ± 0.04	9.53 ± 0.01
	AdaQR	9.62 ± 0.04	9.58 ± 0.02

Table 29: Extended Beyond results with the Muon optimizer (continued).

telemonitoring_parkinsons_biomedical_voice_measurements

Model	Objective	Single	E5
MLP	MSE	12 ± 2	11.6 ± 0.3
	MAE	11.3 ± 0.5	11.0 ± 0.3
	Huber Loss	12 ± 1	11.40 ± 0.08
	Soft Binning	10.9 ± 0.3	10.6 ± 0.2
	Hard Binning	11.2 ± 0.7	10.27 ± 0.05
	Joint End-To-End	10.2 ± 0.2	10.1 ± 0.2
	Decoder[N]	13 ± 1	11.4 ± 0.6
	HL-Gaussian	10.7 ± 0.3	10.5 ± 0.1
	Decoder[U]	11.3 ± 1.0	10.2 ± 0.2
	Softmax MSE	11.7 ± 0.5	11.0 ± 0.1
	Ordinal Entropy	–	–
	ROT-REG	11.7 ± 0.7	11.2 ± 0.3
	GAR [MAE]	–	–
	GAR [MSE]	–	–
	MDN	12 ± 2	10.7 ± 0.6
	NLL	11.1 ± 0.7	10.6 ± 0.3
	beta-NLL	11.4 ± 0.6	10.9 ± 0.3
	AdaQR	11.1 ± 0.6	10.8 ± 0.2
	Quantile Loss	12 ± 1	11.2 ± 0.5
TabPack	MSE	11.3 ± 0.5	–
	HL-Gaussian	10.5 ± 0.8	–
	Quantile Loss	10.9 ± 0.7	–
	AdaQR	11.2 ± 0.6	–
RealMLP	MSE	10.6 ± 0.4	10.4 ± 0.1
	HL-Gaussian	10.9 ± 0.4	10.7 ± 0.3
	Quantile Loss	11.0 ± 0.3	10.88 ± 0.08
	AdaQR†	10.8 ± 0.4	10.6 ± 0.3
TabM	MSE	11.1 ± 0.2	11.09 ± 0.05
	HL-Gaussian	11.1 ± 0.4	10.9 ± 0.2
	Quantile Loss	10.9 ± 0.1	10.83 ± 0.03
	AdaQR	10.8 ± 0.2	10.7 ± 0.1

video_game_fps_prediction

Model	Objective	Single	E5
MLP	MSE	0.008 ± 0.002	0.0052 ± 0.0003
	MAE	0.008 ± 0.002	0.0064 ± 0.0005
	Huber Loss	0.010 ± 0.002	0.0077 ± 0.0002
	Soft Binning	0.0040 ± 0.0006	0.0034 ± 0.0002
	Hard Binning	0.046 ± 0.004	0.0444 ± 0.0005
	Joint End-To-End	0.0100 ± 0.0009	0.0084 ± 0.0005
	Decoder[N]	0.026 ± 0.003	0.022 ± 0.002
	HL-Gaussian	0.006 ± 0.001	0.0044 ± 0.0007
	Decoder[U]	0.039 ± 0.004	0.034 ± 0.004
	Softmax MSE	0.010 ± 0.002	0.0081 ± 0.0001
	Ordinal Entropy	0.007 ± 0.001	0.0063 ± 0.0002
	ROT-REG	0.008 ± 0.002	0.0074 ± 0.0010
	GAR [MAE]	0.0053 ± 0.0009	0.0041 ± 0.0005
	GAR [MSE]	0.0074 ± 0.0008	0.0067 ± 0.0002
	MDN	0.014 ± 0.004	0.010 ± 0.001
	NLL	0.010 ± 0.002	0.008 ± 0.001
	beta-NLL	0.009 ± 0.001	0.0065 ± 0.0002
	AdaQR	0.0025 ± 0.0008	0.0019 ± 0.0001
	Quantile Loss	0.0032 ± 0.0008	0.0025 ± 0.0005
TabPack	MSE	0.0041 ± 0.0005	–
	HL-Gaussian	0.0037 ± 0.0005	–
	Quantile Loss	0.0035 ± 0.0003	–
	AdaQR	0.0032 ± 0.0004	–
RealMLP	MSE	0.0011 ± 0.0000	0.0010 ± 0.0000
	HL-Gaussian	0.0011 ± 0.0000	0.0011 ± 0.0000
	Quantile Loss	0.0012 ± 0.0000	0.0011 ± 0.0000
	AdaQR†	0.0010 ± 0.0000	0.0010 ± 0.0000
TabM	MSE	0.0029 ± 0.0003	0.0027 ± 0.0000
	HL-Gaussian	0.0023 ± 0.0002	0.0022 ± 0.0000
	Quantile Loss	0.0019 ± 0.0004	0.0017 ± 0.0001
	AdaQR	0.0015 ± 0.0002	0.0014 ± 0.0000

Table 30: Extended Beyond results with the Muon optimizer (continued).

video_transcoding_time_prediction

Model	Objective	Single	E5
MLP	MSE	0.269 ± 0.007	0.2594 ± 0.0010
	MAE	0.261 ± 0.008	0.255 ± 0.004
	Huber Loss	0.259 ± 0.007	0.255 ± 0.005
	Soft Binning	0.252 ± 0.003	0.250 ± 0.001
	Hard Binning	0.264 ± 0.003	0.2628 ± 0.0008
	Joint End-To-End	0.244 ± 0.005	0.2402 ± 0.0006
	Decoder[N]	0.249 ± 0.004	0.244 ± 0.002
	HL-Gaussian	0.264 ± 0.009	0.2585 ± 0.0008
	Decoder[U]	0.26 ± 0.02	0.245 ± 0.003
	Softmax MSE	0.254 ± 0.007	0.247 ± 0.002
	Ordinal Entropy	0.253 ± 0.005	0.251 ± 0.001
	ROT-REG	0.265 ± 0.006	0.259 ± 0.001
	GAR [MAE]	0.26 ± 0.01	0.251 ± 0.003
	GAR [MSE]	0.257 ± 0.006	0.253 ± 0.002
	MDN	0.255 ± 0.004	0.252 ± 0.002
	NLL	0.255 ± 0.007	0.251 ± 0.003
	beta-NLL	0.253 ± 0.006	0.250 ± 0.004
	AdaQR	0.250 ± 0.004	0.247 ± 0.002
	Quantile Loss	0.257 ± 0.004	0.254 ± 0.002
TabPack	MSE	0.241 ± 0.008	–
	HL-Gaussian	0.259 ± 0.005	–
	Quantile Loss	0.240 ± 0.007	–
	AdaQR	0.240 ± 0.007	–
RealMLP	MSE	0.270 ± 0.006	0.269 ± 0.004
	HL-Gaussian	0.270 ± 0.005	0.269 ± 0.003
	Quantile Loss	0.251 ± 0.004	0.249 ± 0.001
	AdaQR†	0.246 ± 0.004	0.245 ± 0.002
TabM	MSE	0.243 ± 0.007	0.238 ± 0.002
	HL-Gaussian	0.250 ± 0.004	0.249 ± 0.002
	Quantile Loss	0.238 ± 0.005	0.236 ± 0.003
	AdaQR	0.250 ± 0.006	0.246 ± 0.001

wine_quality

Model	Objective	Single	E5
MLP	MSE	0.650 ± 0.003	0.647 ± 0.002
	MAE	0.653 ± 0.003	0.650 ± 0.002
	Huber Loss	0.648 ± 0.003	0.6451 ± 0.0004
	Soft Binning	0.640 ± 0.001	0.6387 ± 0.0004
	Hard Binning	0.646 ± 0.002	0.6431 ± 0.0003
	Joint End-To-End	0.650 ± 0.003	0.6466 ± 0.0008
	Decoder[N]	0.723 ± 0.006	0.69 ± 0.02
	HL-Gaussian	0.645 ± 0.002	0.6435 ± 0.0002
	Decoder[U]	0.77 ± 0.06	0.718 ± 0.010
	Softmax MSE	0.6457 ± 0.0008	0.6444 ± 0.0001
	Ordinal Entropy	0.649 ± 0.002	0.6477 ± 0.0005
	ROT-REG	0.649 ± 0.003	0.6452 ± 0.0006
	GAR [MAE]	0.650 ± 0.003	0.649 ± 0.001
	GAR [MSE]	0.646 ± 0.002	0.6436 ± 0.0004
	MDN	0.662 ± 0.004	0.650 ± 0.002
	NLL	0.647 ± 0.001	0.6448 ± 0.0002
	beta-NLL	0.650 ± 0.002	0.6470 ± 0.0009
	AdaQR	0.646 ± 0.002	0.6453 ± 0.0009
	Quantile Loss	0.646 ± 0.002	0.6435 ± 0.0004
TabPack	MSE	0.623 ± 0.004	–
	HL-Gaussian	0.621 ± 0.003	–
	Quantile Loss	0.624 ± 0.004	–
	AdaQR	0.622 ± 0.003	–
RealMLP	MSE	0.625 ± 0.003	0.618 ± 0.001
	HL-Gaussian	0.627 ± 0.004	0.6188 ± 0.0002
	Quantile Loss	0.625 ± 0.003	0.620 ± 0.001
	AdaQR†	0.622 ± 0.002	0.618 ± 0.002
TabM	MSE	0.631 ± 0.007	0.625 ± 0.001
	HL-Gaussian	0.625 ± 0.003	0.6191 ± 0.0006
	Quantile Loss	0.631 ± 0.005	0.625 ± 0.003
	AdaQR	0.630 ± 0.003	0.6256 ± 0.0005

Table 31: Extended Beyond results with the Muon optimizer (continued).

wine_world_cost

Model	Objective	Single	E5
MLP	MSE	0.43 ± 0.01	0.4116 ± 0.0008
	MAE	0.415 ± 0.005	0.4119 ± 0.0009
	Huber Loss	0.431 ± 0.010	0.414 ± 0.007
	Soft Binning	0.440 ± 0.007	0.421 ± 0.002
	Hard Binning	0.450 ± 0.006	0.445 ± 0.002
	Joint End-To-End	0.43 ± 0.01	0.419 ± 0.006
	Decoder[N]	0.476 ± 0.009	0.447 ± 0.006
	HL-Gaussian	0.426 ± 0.008	0.414 ± 0.005
	Decoder[U]	0.49 ± 0.01	0.45 ± 0.01
	Softmax MSE	0.427 ± 0.007	0.416 ± 0.002
	Ordinal Entropy	0.422 ± 0.008	0.411 ± 0.005
	ROT-REG	0.431 ± 0.009	0.419 ± 0.003
	GAR [MAE]	0.415 ± 0.008	0.4084 ± 0.0009
	GAR [MSE]	0.417 ± 0.004	0.4093 ± 0.0003
	MDN	0.44 ± 0.01	0.420 ± 0.008
	NLL	0.417 ± 0.004	0.4127 ± 0.0003
	beta-NLL	0.419 ± 0.007	0.414 ± 0.003
	AdaQR	0.423 ± 0.006	0.4163 ± 0.0009
	Quantile Loss	0.425 ± 0.007	0.417 ± 0.003
TabPack	MSE	0.386 ± 0.003	–
	HL-Gaussian	0.40 ± 0.01	–
	Quantile Loss	0.387 ± 0.003	–
	AdaQR	0.392 ± 0.003	–
RealMLP	MSE	0.383 ± 0.003	0.381 ± 0.002
	HL-Gaussian	0.389 ± 0.004	0.387 ± 0.003
	Quantile Loss	0.394 ± 0.003	0.393 ± 0.001
	AdaQR†	0.390 ± 0.009	0.386 ± 0.004
TabM	MSE	0.402 ± 0.003	0.400 ± 0.002
	HL-Gaussian	0.395 ± 0.005	0.392 ± 0.001
	Quantile Loss	0.398 ± 0.006	0.395 ± 0.003
	AdaQR	0.404 ± 0.002	0.4029 ± 0.0006