A CONTROLLED STUDY OF SCALING TRAINING DATA IN TABULAR MACHINE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite increased attention to the field of tabular machine learning, the niche of large-scale (1M+) data remains under-explored. In this work we study training-data scaling on eight datasets with up to ten million examples, evaluating tabular ML methods under realistic distribution shifts with in-distribution (ID) controls. First, we find that GBDTs and MLP-based models generally improve with additional data and exhibit broadly similar scaling trends. Tabular foundation models vary substantially across pretrained versions, with recent models benefiting more consistently from larger contexts, in some cases up to one million examples under distribution shift. Our analysis of MLP-based models shows that, as training data scales, ID and out-of-distribution (OOD) performance can diverge: for MLP-based models selected using ID validation, additional data can improve ID performance while degrading OOD performance. We show that target-aligned validation mitigates this effect through tuning and early stopping. In contrast, tabular foundation models lack a commonly adopted mechanism for using validation as a separate feedback signal, and we demonstrate that in the presence of strong distribution shift a common strategy of adding validation examples to the context does not prevent OOD degradation as training data grows.

1 INTRODUCTION

Machine Learning (ML) on tabular data has recently grown more popular and mature. The field had it's foundation model moment (Hollmann et al., 2025; Qu et al., 2026; Zhang et al., 2026; Jäger et al., 2026), produced improvements to MLP-based DL models (Gorishniy et al., 2025; 2026a; Holzmüller et al., 2024), and started forming sound benchmarking practices (Gorishniy et al., 2021; Erickson et al., 2025; Liu et al., 2025).

Recent work has highlighted the importance of focusing on more diverse and realistic evaluation settings in tabular ML. One such setting concerns the presence of distribution shift between training and evaluation, which is common in many practical tabular prediction tasks (Gardner et al., 2024; Rubachev et al., 2024; Purucker et al., 2026). Another practically important aspect is the availability of large amounts of data in some applications where tabular ML methods are used (Anil et al., 2022; Zou et al., 2022; Ploshkin et al., 2025). This practical relevance, together with the role of data scaling in the success of DL in other modalities (Hestness et al., 2017; Ardalani et al., 2022; Weng, 2026), motivates a deliberate study of data scaling in tabular ML.

Studying these questions requires large tabular datasets with realistic evaluation splits, which remain scarce in existing benchmarks. To enable this study, we assemble a collection of eight datasets from prior work, open ML competitions and adjacent domains, with training sizes reaching around ten million examples. Modern tabular benchmarks typically compare methods across vast collections of datasets (Erickson et al., 2025; Liu et al., 2025) of different properties. Across these datasets, sample size varies together with the prediction task, feature space, and evaluation conditions. Comparing performance across tasks of different sizes therefore does not isolate the effect of dataset size. We instead study scaling within individual tasks, varying training size from 10 000 to around ten million examples while keeping features and evaluation sets fixed. We use natural shifted evaluation alongside in-distribution controls and examine whether gains from additional data persist under distribution shift, and how this depends on model selection.

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.

First, we evaluate GBDTs, MLP-based models and tabular foundation models across training sizes. Non-foundation DL models and GBDTs show similar scaling trends, with task-dependent saturation, and improvements to architectures and training procedures remain beneficial across scales. Foundation models (TabICL v2, TabPFN v3 and v3.5) differ substantially, with recent versions benefiting more consistently from larger contexts.

Then we show that model selection considerations become increasingly important with scale. We perform wide sweeps of MLP parameter sizes and hyperparameter configurations, varying the validation distribution used for tuning and early stopping. As training size grows, in-distribution (ID) validation favors larger models, and increasing the parameter count brings greater gains in in-distribution performance. Under out-of-distribution (OOD) validation and evaluation, we see no gains from increase in model size. We also observe the increasing role validation distribution plays when training data is scaled in the recently introduced TabPack model (Gorishniy et al., 2026a), which uses validation for both early stopping and greedy ensemble selection. We observe that beyond 1M example scale the ID–OOD performance become inversely correlated on many datasets.

This leads us to investigate how validation, better aligned with the held-out test distribution, is used in different tabular modeling paradigms. Under stronger distribution shifts, we observe that standard MLP procedures use validation feedback in model selection, keeping OOD performance stable or improving as training size grows. In ICL-based foundation models, using validation as a separate signal is not natively supported; a common default is to add it to the context. Our experiments show that this improves performance, however, it does not prevent OOD performance decrease as we scale the training dataset size, hinting that ICL-based foundation models may also have a hard time balancing both ID and OOD perfomance.

This motivates further investigation of how best to use target-aligned data in both MLP-based models and PFNs. Combining its roles as feedback for model selection and additional data for fitting, while mitigating the negative ID–OOD relationship that can arise as in-domain data scales.

We summarize our contributions as follows:

- We study how training data size and natural distribution shift jointly affect tabular models, with in-distribution evaluation as a control. To this end, we assemble eight tabular datasets with up to around ten million training examples, and vary training size within each task while keeping features and evaluation sets fixed.
- We find broadly similar scaling trends across parametric models (GBDTs and MLP-based models), with improvements to architectures and training procedures remaining beneficial across scales we consider. Tabular foundation models show substantially different scaling behavior across pretrained versions, with recent models benefiting more consistently from larger contexts, even in challenging out-of-distribution conditions.
- We find that MLP-based tabular DL models can become more sensitive to distribution shift as training dataset size grows. At smaller scales in-distribution and out-of-distribution performance improve together, but this relationship can reverse as training size grows, making validation aligned with the target distribution increasingly important, exhibiting an "*accuracy on the wrong line*"-like pattern (Sanyal et al., 2025) on *real* tabular tasks with *natural* shifts.
- We explore the above effect further and find that validation aligned with the evaluation distribution helps MLP-based models improve as in-distribution training set size grows as it constrains model selection (including tuning and early stopping). For ICL-based foundation models, a common strategy of using the whole available data for in-context "training" improves the overall performance but does not prevent OOD performance decrease at larger training set scales.

2 Related Work

Tabular ML Evaluation. Modern tabular ML benchmarks typically compare models across a set of diverse tabular prediction tasks (Erickson et al., 2025; Liu et al., 2025; Gorishniy et al., 2026a). Recent work has extended these evaluations to tasks with temporal and grouped components with natural distribution shifts (Rubachev et al., 2024; Purucker et al., 2026). Evaluations at multi-million-sample scales have sometimes appeared in prior work (Liu et al., 2025; Gorishniy et al.,

2025; 2026a; Prior Labs, 2025; Richman, 2026), though controlled within-task data scaling on diverse realistic tabular datasets has received relatively little attention. We complement the existing efforts with a systematic study of within-task data scaling up to around ten million training examples under natural distribution shifts, with in-distribution evaluation as a control.

Data Scaling. Our focus on scaling the available task-specific training data is closely related to the study of Hestness et al. (2017), they observed predictable gains from additional data across language, vision, and speech domains, with model improvements primarily shifting the scaling curves rather than changing their power-law exponents. We observe similar trends for parametric tabular models. In deep learning recommendation models, Ardalani et al. (2022) likewise found continued gains from additional data but limited returns to parameter scaling in the architectures studied. We find a related contrast in tabular prediction, where the benefits of additional capacity strongly depend on the evaluation distribution.

For tabular foundation models, multiple existing works study the scaling of the in-context learning pretraining stage (Zhang et al., 2026; Google Research, 2026; Ma et al., 2025), in contrast we study the downstream task scaling, increasing the in-context examples, which is closer in spirit to many-shot in-context learning (Agarwal et al., 2024). We observe substantial differences in scaling behavior across different pretrained model versions, with recent models benefiting more consistently from larger contexts and, in some cases, continuing to improve with up to one million context examples.

Domain Generalization. Understanding how models generalize beyond their training distribution is a long-standing question in deep learning (Gulrajani & Lopez-Paz, 2020; Koh et al., 2021; Malinin et al., 2022). One prominent observation, put forward by Miller et al. (2021) as "*accuracy on the line*," is that models with better in-distribution (ID) performance tend to perform better out of distribution (OOD). In tabular ML, Gardner et al. (2024); Nastl & Hardt (2024) report similar positive correlations. However, Teney et al. (2023) show that ID and OOD performance can also be negatively correlated on real-world datasets. Of particular relevance to data scaling, Sanyal et al. (2025) show how label noise and nuisance features can produce negative correlations, with additional training data improving ID performance while worsening OOD performance. We investigate how this relationship depends on training-set size in real-world tabular tasks, and what it implies for model selection across scales.

3 Setup

3.1 Datasets

We use eight tasks, summarized in Table 2. For Cooking Time, Delivery ETA, Maps Routing, and Weather from TabReD (Rubachev et al., 2024), we retain the existing features and temporal splits and use the larger training sets. We additionally prepare the following four tasks, contributing feature-construction and split recipes for future work.

Jane Street. We use the public training data from the Jane Street forecasting competition,[1] which contains anonymized financial market observations derived from production trading systems. The task is to predict the anonymized market response `responder_6` from market features and historical responses. Following the competition protocol and public solutions, we construct lag features, including responses from earlier validation and test observations. We use the final two periods of 100 date timestamps each for validation and test, respectively, and the preceding data for training.

Aeolus Flight Delay. We use the tabular component of the Aeolus dataset (Xu et al., 2025). The task is to predict whether a flight will depart more than 15 minutes late. We construct calendar and route features, airport-level weather climatology, and smoothed historical delay rates for airports, carriers, and carrier–airport pairs. Historical delay rates use only training-period flights scheduled on earlier calendar days. We use 2016–2022 for training, 2023 for validation, and 2024 for test.

Bank Fraud. We use the Guardian task from Data Fusion Contest 2026.[2] The task is to identify fraudulent transactions using transaction information and customer history. Following public solu-

[1] `https://www.kaggle.com/competitions/jane-street-real-time-market-data-forecasting/data`
[2] `https://ods.ai/competitions/data-fusion2026-guardian`

tions, we construct transaction features and unlabeled customer-history features from the released logs. We use the final two consecutive two-week periods for validation and test, respectively, and the preceding data for training. Following the post-competition analysis, for each customer in an evaluation period, we retain all days with fraud if any are present, and sample one random day otherwise. For training, we sample negatives at a 15% rate. We draw ID splits from the training period with the same class balance as the temporal splits. We use larger evaluation sets than for most other tasks to reduce noise in AP estimates, since positives constitute only about 0.5% of evaluation examples.

Istella. We use the Istella22 search relevance dataset (Dato et al., 2022). The task is to estimate the relevance of a document to a search query. We treat the graded relevance labels as regression targets and retain the 200 non-constant query-document features. We combine the official training and validation partitions for training, and construct validation and test sets from disjoint query groups in the official test partition, whose distribution differs from those of the training and validation partitions. Each holdout contains approximately 600 000 query-document pairs, about 40% of the official test partition, sampled by query group. We use these larger evaluation sets to reduce noise in performance estimates, since they contain only 895 validation and 860 test query groups.

3.2 Evaluation Protocol

In this section we describe the details of our evaluation protocol, which we reuse in the next sections.

Dataset splits. Each task has validation and test sets under its temporal or query-based split for out-of-distribution (OOD) evaluation, together with historical row holdouts as in-distribution (ID) controls. ID rows are reserved before subsampling and match the OOD holdout sizes. For each dataset, validation and test sets remain fixed across all training sizes and models. All models use the same training subsets at each size. Training subsets are nested and the sizes are doubling from 10 000 to 5 120 000, followed by the largest subset which is capped at 10M for datasets significantly larger than this size, and kept for others.

Non-foundation tabular ML models. We evaluate XGBoost, LightGBM, MLP, MLP with piecewise-linear embeddings (denoted by MLP‡) (Gorishniy et al., 2022), MLP trained with Muon optimizer (Gorishniy et al., 2026b), TabM (Gorishniy et al., 2025), TabPack and TabPack with its variant of periodic embeddings (Gorishniy et al., 2026a) (denoted by TabPack†). We tune each method separately at each dataset and training size with 100 trials, except TabPack, which selects a greedy ensemble from 64 sampled configurations.

Tabular foundation models. We evaluate TabICLv2 (Qu et al., 2026), TabPFN v3, v3.5 and its fast variant (Jäger et al., 2026). For all models we use the provided default configurations.

Result reporting. We evaluate ten seeds for each model. We report means and standard deviations of test scores. For regression, we use root mean squared error (RMSE). For binary classification, we use average precision (AP), since positive classes are substantially less frequent than negatives.

4 Results

We start with experiments on how test performance changes as training sets grow under the benchmark's default splits with natural distribution shifts. We first report scaling trends for parametric tabular models in subsection 4.1, and then study tabular foundation models in subsection 4.2.

4.1 Parametric Tabular Models

Results for a representative set of tabular ML models trained with increasing dataset sizes are summarized in Figure 1, for more models see tables in the Appendix F. Our findings are summarized below.

Scaling data provides significant improvements. Overall we see that increasing the training-set size improves test performance across a broad range of parametric models. On many tasks improvements continue through the largest training sizes, approaching ten million examples. The extent of these gains depends on the task: Jane Street largely saturates around one million examples, while improvements on Weather become smaller at the largest sizes.

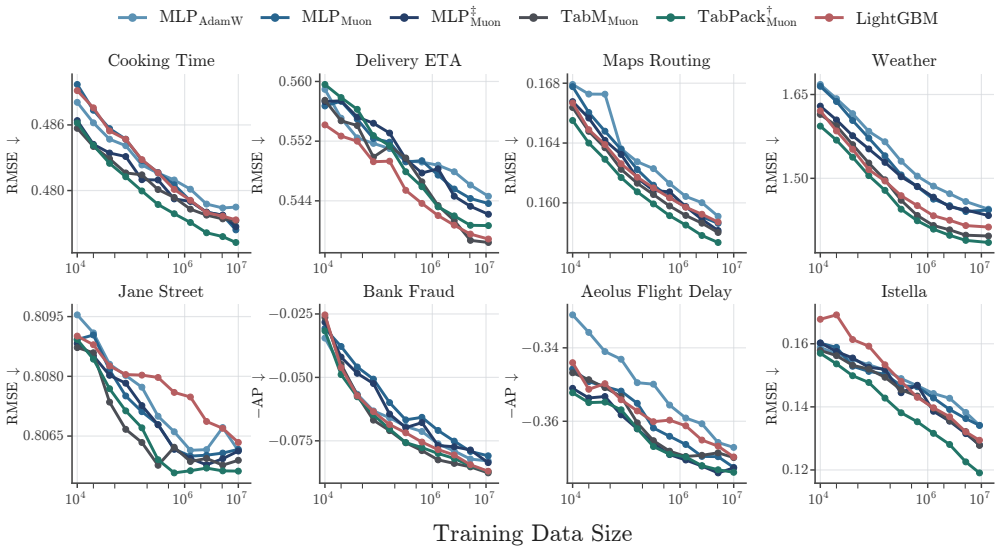

Figure 1: Performance of parametric models across training-set sizes. Horizontal axis shows the number of training examples on a logarithmic scale. We report RMSE for regression and negative average precision ($-$AP) for classification, so lower is better on every plot. On most tasks, additional training data improves performance across methods, improvements to architectures and training procedures also remain beneficial across scales.

Method improvements remain relevant as data scales. Muon often improves vanilla MLP performance across training sizes, numerical feature embeddings provide further gains on several tasks. TabPack† ranks among the top methods across most sizes considered. Although the benefits of individual changes vary across tasks and sizes and are quite noisy, additional data does not eliminate performance gaps. Improvements to models and training procedures thus complement improvements seen from data scaling at this scale.

Overall gains from additional data follow broadly similar trends across different methods within each task. This is consistent with Hestness et al. (2017), who found that model improvements in other deep learning domains mainly shift the data-scaling curves while leaving their power-law exponents largely unchanged. To assess whether the similar scaling trends could be observed across methods on datasets from our evaluation we compare power-law fits with individual and shared exponents in Appendix A and find that individual exponents provides small improvements in fit quality (with the exception of Jane Street dataset).

4.2 Tabular Foundation Models

Results for tabular foundation model in-context training set scaling are summarized in Figure 2. We outline our findings below.

Additional context improves performance under distribution shift. Increasing the in-context training-set size improves test performance on many tasks, with gains continuing up to around one million examples in some cases. The extent of these gains depends on the task: curves on Jane Street and Aeolus Flight Delay can flatten or reverse at larger context sizes.

Scaling behavior differs across pretrained models. Unlike the broadly similar scaling trends of parametric models, foundation models differ substantially in their gains from additional context. TabPFN v3.5 and its fast variant show more consistent improvements with larger contexts than TabPFN v3 and TabICL v2. How these differences relate to model architectures and the alignment between pretraining synthetic prior and real data distributions remains an open question.

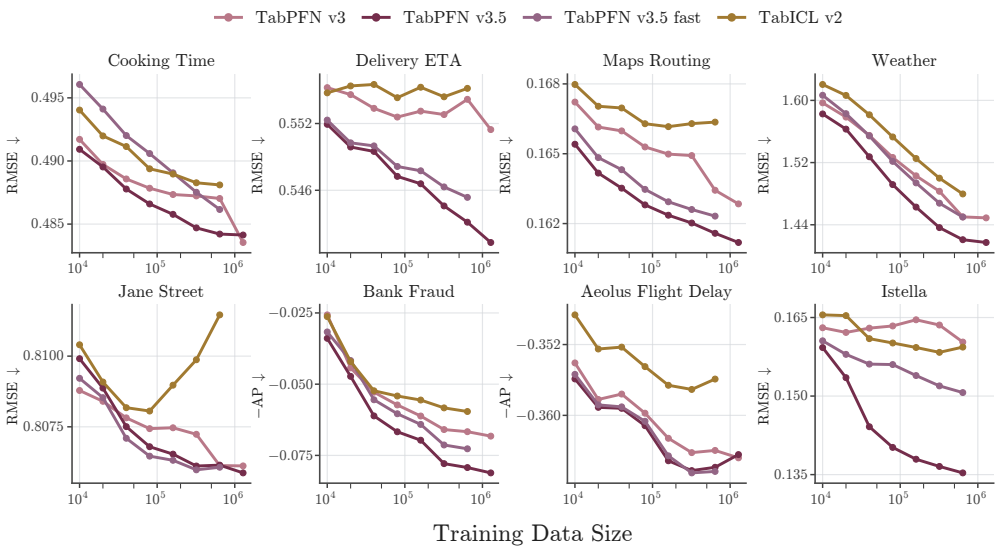

Figure 2: Performance of tabular foundation models across training-set sizes. The figure follows Figure 1 conventions. Tabular foundation models also improve with in-context training set scale, but the scaling trends vary significantly across different models.

5 Analysis

Sections 4.1 and 4.2 showed that additional training data improves out-of-distribution test performance for most models and tasks. In this section, we show that increasing the training data scale non-trivially affects questions concerning model selection and out-of-dirstribution generalization. We use in-distribution validation and test splits as controls for the analysis.

In subsection 5.1, we sweep MLP widths and hyperparameters across training sizes. We find that in-distribution and out-of-distribution evaluation increasingly "disagree" on the benefits of model capacity. In subsection 5.2, we show that this disagreement matters for model selection, starting at 1M examples scale. In subsection 5.3, we compare how MLPs and tabular foundation models use validation data aligned with the test distribution and how this design decision affects scaling.

5.1 Model Capacity Across Training Sizes

To study how training size and model capacity jointly affect performance, we run a large sweep of MLPs over widths, hyperparameters, and training sizes. Batched packed hyperparameter ensembles introduced in(Gorishniy et al., 2026a) make such a sweep inexpensive and allows to efficiently explore a large space of MLP models. The setup is as follows:

- **Models.** MLPs with three hidden blocks and 11 widths from 64 to 1936, spanning approximately 10^4 to 10^7 parameters.
- **Hyperparameters.** For each width, one pack of 100 MLPs covering a 10×10 grid of learning rates and weight decays.
- **Training.** AdamW for 100 000 steps with batch size 1 024, evaluating every 1 000 steps.
- **Selection.** For each width and training size, we select the learning rate, weight decay, and checkpoint on either the OOD or the ID validation set. Both protocols share the same training runs.

In total, we train 95 700 MLPs across eight datasets. Beyond the selected models, the stored trajectories give an overview of learning dynamics across training sizes and capacities, and allow for experimenting with different model and hyperparameter selection strategies post-hock. Results of

the sweeps are summarized in Figure 3. Results on all datasets are in Appendix D. Below we provide our key takeaways.

ID and OOD evaluation "disagree" on model capacity at scale. On OOD test sets, more training data improves performance. But, larger models bring little or no consistent gain (one can achieve Best performance with around 100k parameters, on most datasets). In contrast, on in-distribution controls, larger models improve performance on almost all datasets, and these gains grow with training size. Ardalani et al. (2022) similarly report stronger gains from data than from parameters in the MLP components of DLRMs. In our setting, the benefit of capacity depends on the evaluation distribution. As training size grows, ID and OOD validation therefore increasingly favor different models, the phenomenon we examine further in the next section.

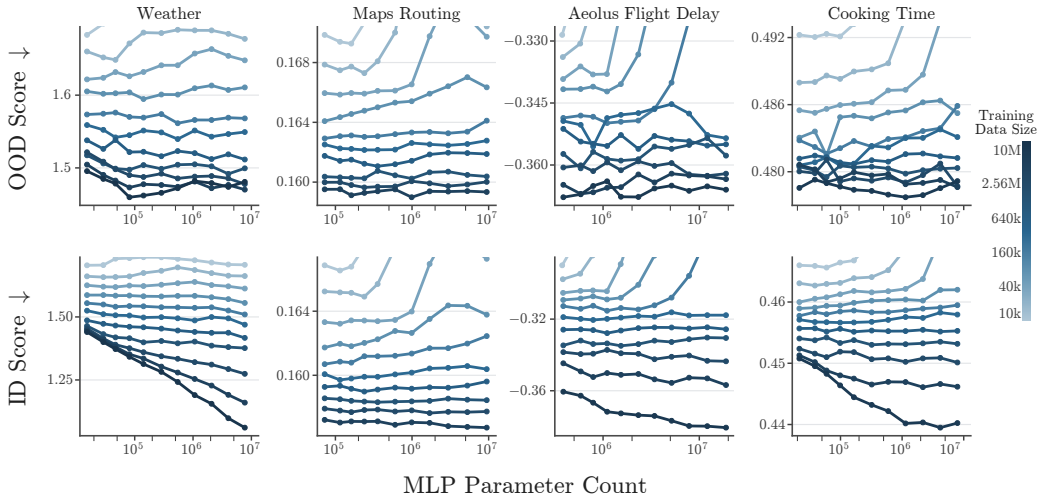

Figure 3: MLP width sweeps across training sizes Top row: OOD test performance; bottom row: ID controls. The horizontal axis shows parameters per MLP on a logarithmic scale; darker curves indicate larger training sets. Learning rate, weight decay, and checkpoint are selected on the corresponding validation split. We report RMSE for all datasets except Aeolus Flight Delay, where we report negative average precision ($-$AP). Lower is better. Gains from increasing model size are limited under distribution shift, but become more pronounced with larger training sets under ID evaluation

5.2 Validation Distribution at Scale

If ID and OOD validation favor different models, the choice of validation distribution should matter more at larger training sizes. We test this hypothesis with a recently introduced TabPack model (Gorishniy et al., 2026a), which uses validation for early stopping and for greedy ensemble selection, intuitively this should make it increasingly dependend on validation set changes.

We run TabPack with either the default OOD validation set or the ID validation set, keeping training subsets and all other settings fixed. We evaluate both variants on ID and OOD test sets. Results are summarized in Figure 4. Below we outline our key observation.

Accuracy on the wrong line. At smaller training sizes, ID and OOD performance generally improve together, with little difference between the two validation protocols. As training size grows, however, this relationship reverses: with ID-based selection, additional data continues to improve ID performance, and the rate of improvement increases, while OOD performance deteriorates. With OOD-based selection, OOD performance remains stable or improves, although gains on the ID test sets are smaller. These results resemble the "accuracy on the wrong line" pattern (Sanyal et al., 2025) on real tabular tasks with natural distribution shifts. Moreover, the observed pattern would have been barely noticable on a sub 1M train set scale. This also aligns with the observations in Sanyal et al. (2025), where on noisy datasets increasing data size exacerbates the divergence betwen ID and OOD performance.

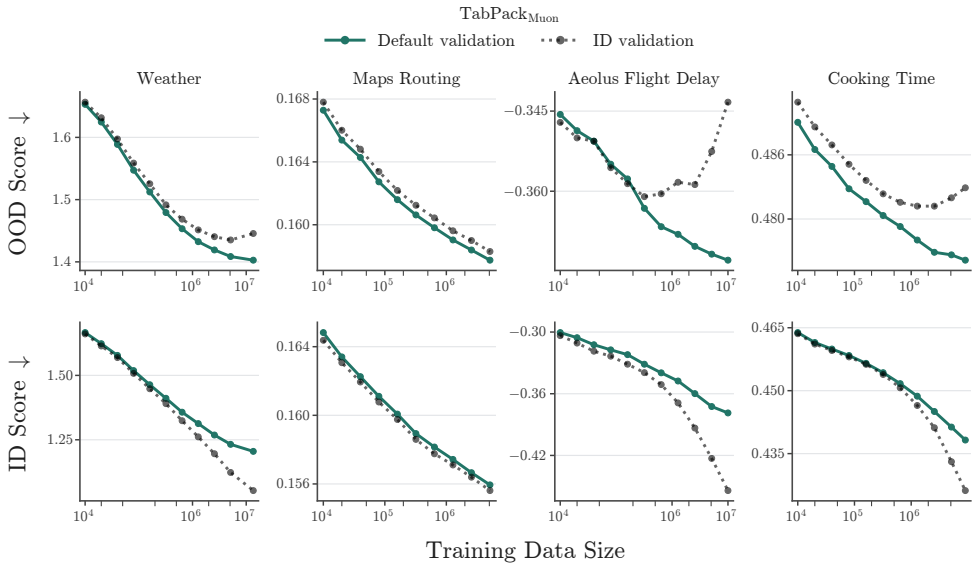

Figure 4: Validation distribution affects ID and OOD scaling in TabPack. Solid curves use OOD validation ("Default validation"); dotted curves use ID validation. We report RMSE for all datasets except Aeolus Flight Delay, where we report negative average precision ($-\text{AP}$). Lower is better in all panels. At larger training sizes, ID validation can improve ID performance while OOD performance deteriorates; conversely, OOD validation can improve OOD performance at the cost of ID performance.

5.3 Using Validation in MLPs and Foundation Models

Previous section results showed that, at scale, validation aligned with the test distribution matters for parametric models. The two main modeling paradigms use such validation data differently. MLP-based models often utilize validation as a held-out signal for model selection (through tuning and early-stopping). Tabular foundation models do not have a separate validation concept and the default strategy is to fit on all the available data.

We try to compare the two strategies and modeling paradigms in this section. To imitate the effect we observed in the previous sections at sub-1M scale we manually increase the level of distribution shift between training and evaluation, intuitively this should move the ID–OOD tradeoff to smaller scale datasets. We achieve this differently by taking different subsets of the training data on Weather, Aeolus Flight Delay and Cooking Time datasets that are more OOD compared to the validation and test sets (interpretable strategies are described in Appendix B). We then repeat the protocol of data scaling, but additionally we also vary the representative validation set size and run two models MLP and TabPFNv3.5.

Results are summarized in Figure 5. Our key takeaways are as follows:

Validation as additional context. Using representative validation examples in-context yields large gains at small training sizes, and larger validation sets provide increasingly larger gains. This is expected as using more relevant data for training should improve performance.

Validation as a model-selection signal. However, we see on all three datasets that using all the test-aligned validation examples in ICL does not save tabular foundation models from the "accuracy on the wrong line" like behaviour and, with the amount of in-distribution data increasing, out-of-distribution performance starts to degrade. MLP, which uses validation for model selection, is generally more stable, and even improves as training size grows at this scale, and this does not really depend on the validation set size.

Takeaway. These two observations together illustrate that the optimal strategy of target aligned validation is not yet settled in both paradigms. MLP-like non-foundation models could benefit from

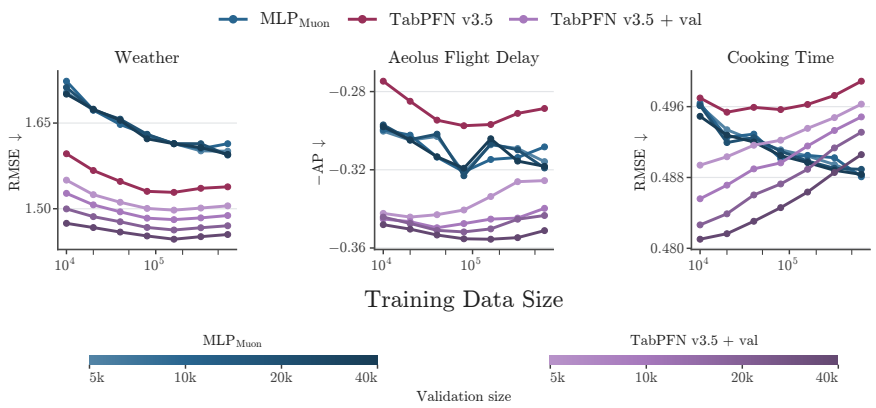

Figure 5: Validation as context vs. as a selection signal on restricted training pools of Weather, Aeolus Flight Delay, and Cooking Time. Shades of blue (MLP) and purple (TabPFN v3.5 with appended validation) indicate validation sizes from 5 000 to 40 000. Conventions follow Figure 1. TabPFN benefits from additional validation examples, but adding more training data can hurt performance. The MLP benefits more consistently from additional training data, but gains little from larger validation sets.

procedures like refitting, and foundation models could benefit from having a more explicit way to control model selection, or specify the "shift direction" with some hold-out validation.

6 Limitations

Our work is a controlled study rather than a benchmark. In all the experiments we keep validation and test sets fixed across training sizes for comparability, which may be less representative of real-world practice at small training sizes, but provides cleaner signal for studying the effects of scaling training data. We do not directly compare parametric models and tabular foundation models. The two paradigms currently operate at different scales (10M is easy for non-foundation models, but 1M is the current limit for foundation models), running foundation models on ten million examples requires additional techniques such as context subsampling or retrieval, which could potentially introduce confounding effects. Furthermore, as section 5 shows, a proper comparison also depends on how each paradigm uses validation data, and there is no settled answer to this yet. We also use eight datasets, the representative minimum we considered sufficient for studying within-task scaling.

7 Conclusion

Large training sets are common in practical and important tabular applications but they were rarely studied in tabular ML. We assembled a representative set of datasets and studied this regime. We find that progress that tabular ML community has made holds up to scaling data. Non-foundation model architectural improvements complement improvements from scaling training data, foundation models continue to improve, and also show promising scaling trends, even in challenging realistic conditions. However, our analysis showed that increasing training data scale, in the presence of distribution shifts introduces new challenges. At small scales, ID and OOD performance mostly improve together. At larger scales, they can diverge, making OOD generalization a more important concern. This raises questions we find worth pursuing in future work. How should target-aligned data be split between fitting and model selection as in-domain data grows? Can foundation models be given an effective mechanism to treat validation examples differently from training examples? How does model-level generalization interventions interplay with hyperparameter tuning at scale, are there new generalization inducing methods to be discovered that are particularly interesting at scale.

AI USE STATEMENT

In this work, we used generative AI tools to implement parts of the experimental code, to help with preparing new datasets (with humans ni the loop), to provide feedback on research methodology and experiments, and to discuss the interpretation of results. We have not used generative AI tools to propose hypotheses. Other uses requiring disclosure are not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We supply the minimal code needed for dataset preprocessing and running the experiments in supplementary materials. Dataset sources, feature construction, and splits are described in section 3 Hyperparameter spaces, tuning protocols, and training details are described in Appendix C.

REFERENCES

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-Shot In-Context Learning, October 2024.

Rohan Anil, Sandra Gadanho, Da Huang, Nijith Jacob, Zhuoshu Li, Dong Lin, Todd Phillips, Cristina Pop, Kevin Regan, Gil I. Shamir, Rakesh Shivanna, and Qiqi Yan. On the Factory Floor: ML Engineering for Industrial-Scale Ads Recommendation Models, September 2022.

Newsha Ardalani, Carole-Jean Wu, Zeliang Chen, Bhargav Bhushanam, and Adnan Aziz. Understanding Scaling Laws for Recommendation Models, August 2022.

Domenico Dato, Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, and Nicola Tonellotto. The Istella22 Dataset: Bridging Traditional and Neural Learning to Rank Evaluation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pp. 3099–3107, New York, NY, USA, July 2022. Association for Computing Machinery. ISBN 978-1-4503-8732-3. doi: 10.1145/3477495.3531740.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A Living Benchmark for Machine Learning on Tabular Data, November 2025.

Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking Distribution Shift in Tabular Data with TableShift, February 2024.

Google Research. TabFM: Tabular Foundation Models. Google˜Research, September 2026.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting Deep Learning Models for Tabular Data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943. Curran Associates, Inc., 2021.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On Embeddings for Numerical Features in Tabular Deep Learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24991–25004. Curran Associates, Inc., 2022. doi: 10.52202/068431-1812.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing Tabular Deep Learning with Parameter-Efficient Ensembling, February 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*, June 2026a.

Yury Gorishniy, Ivan Rubachev, Dmitrii Feoktistov, and Artem Babenko. Benchmarking Optimizers for MLPs in Tabular Deep Learning, April 2026b.

Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization, July 2020.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep Learning Scaling is Predictable, Empirically, December 2017.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6.

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*, volume 37, pp. 26577–26658. Curran Associates, Inc., 2024. doi: 10.52202/079017-0837.

Benjamin Jäger, Nick Erickson, Léo Grinsztajn, Felix Birkel, Klemens Flöge, Oscar Key, Kürşat Kaya, Jonas Kübler, Adèle Frankel, Tobias Schröder, Anurag Garg, Jan Hendrik Metzen, David Salinas, Simon Bing, Kristina Collins, Tuana Çelik, Vahid Balazadeh, Lydia Sidhoum, Tomás Pereda, Brendan Roof, Andrej Tschalzev, Siyuan Guo, Philipp Singer, Lennart Purucker, Jake Robertson, Marie Salmon, Philipp Jund, Jerry Chen, Diana Kriuchkova, Arthur Cahu, Eliott Kalfon, Adrian Hayler, Georg Grab, Vitor Monteiro, Lilly Wehrhahn, Dominik Safaric, Clara Cornu, Alan Arazi, Rylee Grace, Simone Alessi, Mihir Manium, Bernhard Schölkopf, Yann LeCun, Madelon Hulsebos, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-3.5: Technical Report, September 2026.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A Benchmark of in-the-Wild Distribution Shifts, July 2021.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Huai-Hong Yin, Tao Zhou, Jun-Peng Jiang, and Han-Jia Ye. Talent: A tabular analytics and learning toolbox. *Journal of Machine Learning Research*, 26 (226):1–16, 2025.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling Tabular Foundation Models on Real Data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025.

Andrey Malinin, Neil Band, Ganshin, Alexander, German Chesnokov, Yarin Gal, Mark J. F. Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Roginskiy, Denis, Mariya Shmatova, Panos Tigas, and Boris Yangel. Shifts: A Dataset of Real Distributional Shift Across Multiple Large-Scale Tasks, February 2022.

John P. Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7721–7735. PMLR, July 2021.

Vivian Y. Nastl and Moritz Hardt. Do causal predictors generalize better to new domains?, October 2024.

Alexander Ploshkin, Vladislav Tytskiy, Alexey Pismenny, Vladimir Baikalov, Evgeny Taychinov, Artem Permiakov, Daniil Burlakov, and Eugene Krofto. Yambda-5B — A Large-Scale Multi-Modal Dataset for Ranking and Retrieval. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, pp. 894–901, New York, NY, USA, September 2025. Association for Computing Machinery. ISBN 979-8-4007-1364-4. doi: 10.1145/3705328.3748163.

Prior Labs. TabPFN scaling mode: Built for production. Scaled to millions. https://priorlabs.ai/tabpfn/large-data, 2025.

Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David Holzmüller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, Gaël Varoquaux, and Frank Hutter. Beyond IID: How General Are Tabular Foundation Models, Really?, June 2026.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model, September 2026.

Ronald Richman. Scaling Laws, Tabular Data and Actuarial Ratemaking Models, January 2026.

Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, and Artem Babenko. TabReD: Analyzing Pitfalls and Filling the Gaps in Tabular Deep Learning Benchmarks. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Amartya Sanyal, Yaxi Hu, Yaodong Yu, Yian Ma, Yixin Wang, and Bernhard Schölkopf. Accuracy on the wrong line: On the pitfalls of noisy data for out-of-distribution generalisation. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, pp. 2170–2178. PMLR, April 2025.

Damien Teney, L. I. N. Yong, Seong Joon Oh, and Ehsan Abbasnejad. ID and OOD Performance Are Sometimes Inversely Correlated on Real-world Datasets. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Lilian Weng. Scaling Laws, Carefully. https://lilianweng.github.io/posts/2026-06-24-scaling-laws/, June 2026.

Lin Xu, Xinyun Yuan, Yuxuan Liang, Suwan Yin, and Yuankai Wu. Aeolus: A Multi-structural Flight Delay Dataset. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, October 2025.

Xingxuan Zhang, Gang Ren, Hao Yuan, Hao Zou, Hongze Tan, Hui Wang, Jianhao Song, Jiansheng Li, Jiayao Zhang, Jinghan Zhang, Kaifang Li, Lang Mo, Li Mao, Mingchao Hao, Nuo Xu, Rui Ding, Ruiji Zhang, Shuyang Li, Siyu Mei, Tianyang Zhang, Weiyang Mu, Yancheng Dong, Yongxian Wei, Yuan Xue, Yuanrui Wang, Yue He, Zijia Yang, Ziyun Li, Dongzhe Li, Fuqiang Wang, Jiandong Liu, Jiawei Chen, Jiaxin Du, Kaijie Cheng, Kehan Li, Lei Sun, Linjun Zhou, Ningbo Dai, Qi Wang, Renzhe Xu, Shaoxing Du, Shumeng Yang, Wang Lu, Wenjing Chu, Xiannan Huang, Xiaoyu Lin, Xing Ai, Xinyan Han, Xuanyue Li, Xuanyue Su, Xukun Zhang, Yan Lu, Yaxin Zhang, Yi Qin, Yifei Huang, Yihan Xu, Yongle Lv, Yuanyuan Jiang, Yushan Han, and Peng Cui. LimiX-2: A Contextual Mechanism Network Towards General Structured-Data Intelligence, September 2026.

Lixin Zou, Haitao Mao, Xiaokai Chu, Jiliang Tang, Wenwen Ye, Shuaiqiang Wang, and Dawei Yin. A Large Scale Search Dataset for Unbiased Learning to Rank. *Advances in Neural Information Processing Systems*, 35:1127–1139, December 2022. doi: 10.52202/068431-0083.

A SHARED SCALING EXPONENTS ACROSS METHODS

We compare method-specific and dataset-shared power-law exponents using 71 primary-test learning curves from nine parametric methods on eight datasets. Each point averages ten seeds; curves require at least five completed sizes $N \geq 10^4$. We fit

$$E_{d,m}(N) = a_{d,m}(N/10^4)^{-b} + c_{d,m}, \qquad b = b_{d,m} \text{ or } b_d, \tag{1}$$

where E is RMSE or $1 - \mathrm{AP}$. Amplitudes and asymptotes remain method-specific; a shared exponent is estimated separately for each dataset.

We minimize the mean squared residual for each curve. For the shared fit, we normalize each curve by its error at the smallest training size, then sum the mean squared residuals across methods. The bounds are $a, c \geq 0$, $10^{-4} \leq b \leq 3$, and $c \leq 1$ for classification.

Sharing the exponent reduces mean in-sample R^2 by less than 0.01 on 7 of eight datasets (Table 1). Jane Street is the main exception. This supports similar scaling across methods within most datasets over the observed size range, while allowing different error levels and asymptotes.

Dataset	b_d	R^2_{ind}	R^2_{shared}	ΔR^2
Cooking Time	0.1625	0.9938	0.9931	0.0006
Delivery ETA	0.0055	0.9727	0.9673	0.0054
Maps Routing	0.1811	0.9935	0.9929	0.0006
Weather	0.1843	0.9921	0.9904	0.0017
Jane Street	0.3403	0.9312	0.9024	0.0288
Bank Fraud	0.3373	0.9930	0.9849	0.0082
Aeolus Flight Delay	0.0985	0.9663	0.9632	0.0031
Istella	0.0401	0.9773	0.9752	0.0021

Table 1: Dataset-shared exponents and mean in-sample R^2 over methods. $\Delta R^2 = R^2_{\text{ind}} - R^2_{\text{shared}}$, computed before rounding.

Table 2: Dataset statistics (rounded). Training counts are in millions; validation and test counts are in thousands, per evaluation condition. The first four datasets are from TabReD. OOD splits are temporal except for Istella (query-based). R: regression; B: binary classification.

Dataset	Train (M)	Val. (K)	Test (K)	Features	Task
Maps Routing	5.12	60	60	986	R
Cooking Time	9.00	51	42	192	R
Delivery ETA	11.08	34	37	223	R
Weather	13.54	42	41	103	R
Istella	9.21	600	600	200	R
Jane Street	10.00	93	93	94	R
Aeolus Flight Delay	10.00	100	100	91	B
Bank Fraud	10.97	591	590	119	B

B Dataset construction for the validation-data experiment

To study adaptation to temporal out-of-distribution (OOD) data, we restrict which historical periods contribute training examples while keeping the original temporal validation and test periods:

- **Weather: the latest 10% of training rows.** We keep the latest 10% of the original training rows in chronological order, covering the late part of spring 2023, with validation in June and testing in July. Restricting seasonal coverage makes the model transfer from a narrow recent window to a later part of the seasonal cycle. Additional training examples from this window do not restore the missing seasonal coverage.
- **Aeolus Flight Delay: March–December 2020.** From the benchmark's existing training pool, we retain only flights with scheduled departures from March 1 through December 31, 2020. Validation uses 2023 and testing uses 2024. This isolates the pandemic-era period and excludes the more recent 2021–2022 training data, increasing temporal separation and testing transfer across periods with potentially different traffic and delay patterns. The restriction applies to training rows; the original historical-context features are retained.
- **Cooking Time: the earliest 25% of training rows.** We keep the earliest 25% of the original training rows in chronological order. Validation remains December 21–27, 2023, and testing starts on December 28. Removing the later training rows widens the gap to evaluation, exposing the model to changes in the order population and cooking-time relationships that may occur over time. Increasing the training size still draws examples only from the earlier period.

C PREPROCESSING AND HYPERPARAMETER TUNING

C.1 DATASET METADATA

C.2 EXPERIMENTAL SETUP

Data preprocessing. Our default numerical preprocessing for MLP, MLP-PLE, TabM and TabPack is quantile normalization to a normal distribution (Gorishniy et al., 2021; 2025). Missing numerical values are filled with zero, constant columns are removed, and binary features are treated as categorical. MLP, MLP-PLE, TabM and TabPack use one-hot categorical encoding; tree models use native categorical handling without additional numerical normalization. TabPFN and TabICL receive ordinal-encoded categories and retain their internal preprocessing. Regression targets are standardized using training statistics; predictions are transformed back before evaluation.

Training neural networks. We minimize mean squared error for regression and binary cross-entropy for classification, using constant learning rates. MLP/MLP-PLE and TabM use gradient clipping at norm 1 and patience parameter 16. Batch size depends on the training set size N as follows.

Training set size N	Batch size
$N \leq 20\,000$	256
$20\,000 < N < 160\,000$	512
$160\,000 \leq N < 1\,280\,000$	1024
$1\,280\,000 \leq N < 5\,120\,000$	2048
$N \geq 5\,120\,000$	4096

Hyperparameter tuning and evaluation. We tune each model separately for each dataset and training size, selecting minimum validation RMSE or maximum AP. MLP, MLP-PLE, TabM, XGBoost and LightGBM use 100 TPE trials; TabPack uses the random search described below. Selected configurations are retrained for evaluation. Hyperparameters and checkpoints are selected using the validation split. The tables below give the default search spaces.

C.3 MLP AND MLP-PLE

Both variants use the same backbone space. MLP-PLE adds quantile-based piecewise-linear embeddings (Gorishniy et al., 2022; 2025), with training-set bin boundaries, a learned projection and no embedding activation. Either AdamW or Muon is used.

Parameter	Distribution or value
Number of blocks	$\mathrm{UniformInt}[1, 6]$
Hidden width	$\mathrm{UniformInt}[64, 1024]$, step 16
Dropout	$\{0, \mathrm{Uniform}[0, 0.5]\}$
Embedding width (PLE only)	$\mathrm{UniformInt}[8, 32]$, step 4
Number of bins (PLE only)	$\mathrm{UniformInt}[2, 128]$
Tuning trials	100

C.4 TABM

TabM uses $k = 16$ ensemble members with independent training batches and Muon optimization.

Parameter	Distribution or value
Ensemble size k	16
Number of blocks	$\mathrm{UniformInt}[1, 5]$
Hidden width	$\mathrm{UniformInt}[64, 1024]$, step 16
Dropout	$\{0, \mathrm{Uniform}[0, 0.5]\}$
Tuning trials	100

C.5 TABPACK

TabPack uses 64 independently parameterized ReLU networks with Muon. Configurations are sampled by random search. The cosine variant TabPack† concatenates each numerical input with $d-1$ learned cosine features and applies an elementwise affine transform, where d is the embedding width.

Parameter	Distribution or value
Number of blocks	$\mathrm{UniformInt}[1,4]$; cosine: $\mathrm{UniformInt}[1,3]$
Hidden width	384
Dropout	$\{0, \mathrm{Uniform}[0,0.5]\}$
Embedding width	$\mathrm{UniformInt}[8,32]$, step 4
Initialization scale	$\mathrm{LogUniform}[0.01,10]$
Sampled configurations	64
Maximum ensemble size	32

C.6 OPTIMIZERS

Optimizer and architecture parameters are tuned jointly. AdamW is used alone for the AdamW variants and as the auxiliary optimizer for Muon. Below, (A) denotes MLP/MLP-PLE and (B) denotes TabM/TabPack, including cosine. The dropout choice $\{0, \mathrm{Uniform}[0,0.5]\}$ is categorical, with equal probabilities under random search.

AdamW and auxiliary AdamW

Parameter	Distribution
Learning rate	(A) $\mathrm{LogUniform}[3\cdot10^{-5}, 10^{-3}]$ (B) $\mathrm{LogUniform}[10^{-4}, 0.005]$
Weight decay	(A) $\mathrm{LogUniform}[0.005, 5]$ (B) $\mathrm{LogUniform}[0.001, 1]$

Muon

Parameter	Distribution
Muon learning rate	(A) $\mathrm{LogUniform}[10^{-4}, 0.03]$ (B) $\mathrm{LogUniform}[0.001, 0.1]$

Muon uses the auxiliary AdamW rate above; its weight decay follows the same ranges.

C.7 GRADIENT-BOOSTED TREES

XGBoost and LightGBM use at most 10,000 trees and early stopping after 300 rounds without validation improvement.

XGBoost

Parameter	Distribution or value
Maximum depth	$\mathrm{UniformInt}[3,10]$
Minimum child weight	$\mathrm{LogUniform}[10^{-4}, 100]$
Row subsampling	$\mathrm{Uniform}[0.5,1]$
Feature subsampling	$\mathrm{Uniform}[0.5,1]$
Learning rate	$\mathrm{LogUniform}[0.001,1]$
Split penalty γ	$\{0, \mathrm{LogUniform}[0.001, 100]\}$
ℓ_1 and ℓ_2 regularization	$\{0, \mathrm{LogUniform}[10^{-4}, 5]\}$, separately
Growth policy	{depthwise, lossguide}
Tuning trials	100

LightGBM

Parameter	Distribution or value
Number of leaves	$\mathrm{UniformInt}[2, 256]$
Minimum child samples	$\mathrm{UniformInt}[5, 256]$
Minimum leaf Hessian sum	0.001
Row subsampling	$\mathrm{Uniform}[0.5, 1]$, every round
Feature subsampling	$\mathrm{Uniform}[0.5, 1]$
Learning rate	$\mathrm{LogUniform}[0.005, 0.1]$
ℓ_2 regularization	$\{0, \mathrm{LogUniform}[0.1, 10]\}$
Extra trees	{false, true}
Tuning trials	100

For both tree models, zero-or-log-uniform parameters use a categorical switch.

D Full model-size sweeps

Figures 6 and 7 show the model-size sweeps on all eight datasets. For each training-set size and MLP width, we select the learning rate, weight decay, and checkpoint using the corresponding validation split. Darker curves indicate larger training sets. The metrics are RMSE and negative average precision; lower is better. Only completed runs are shown.

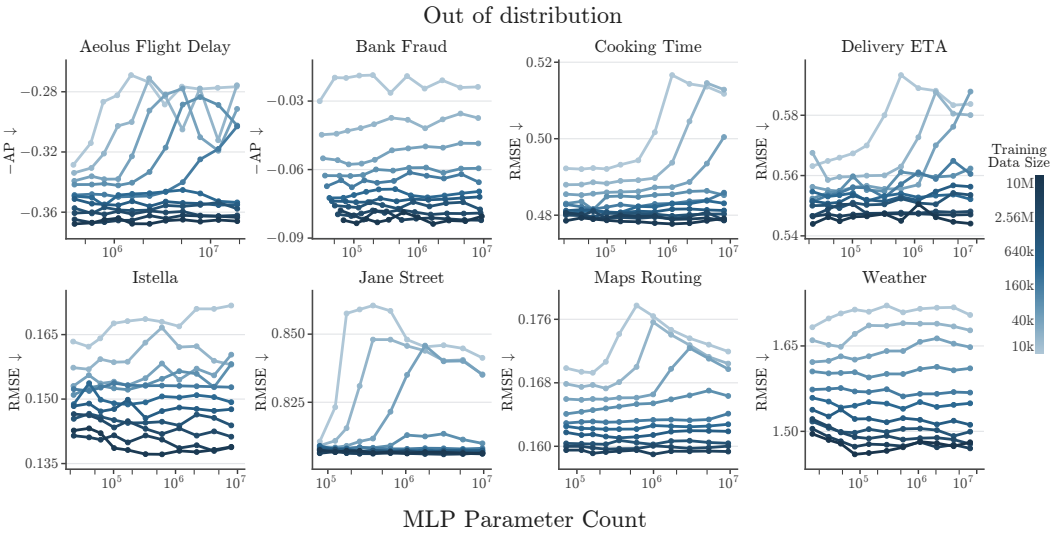

Figure 6: Default protocol: temporal validation and test splits, or query for Istella. Points show seed means; shaded bands show one standard deviation where multiple seeds are available.

E TabPack with different validation splits

Figures 8 and 9 compare TabPack with Muon using default and ID validation on all eight datasets. Each figure evaluates both validation protocols on the same test split. Points show means over available saved seeds. The metrics are RMSE and negative average precision; lower is better.

F Full results for parametric methods

Primary-test means and sample standard deviations. —: no result. **Bold** marks first place and *italics* second place among multi-seed runs at each size (sequential Welch tests, two-sided unadjusted $p < 0.05$). ‡: piecewise linear embeddings; †: cosine feature embeddings. "full" is the full training split.

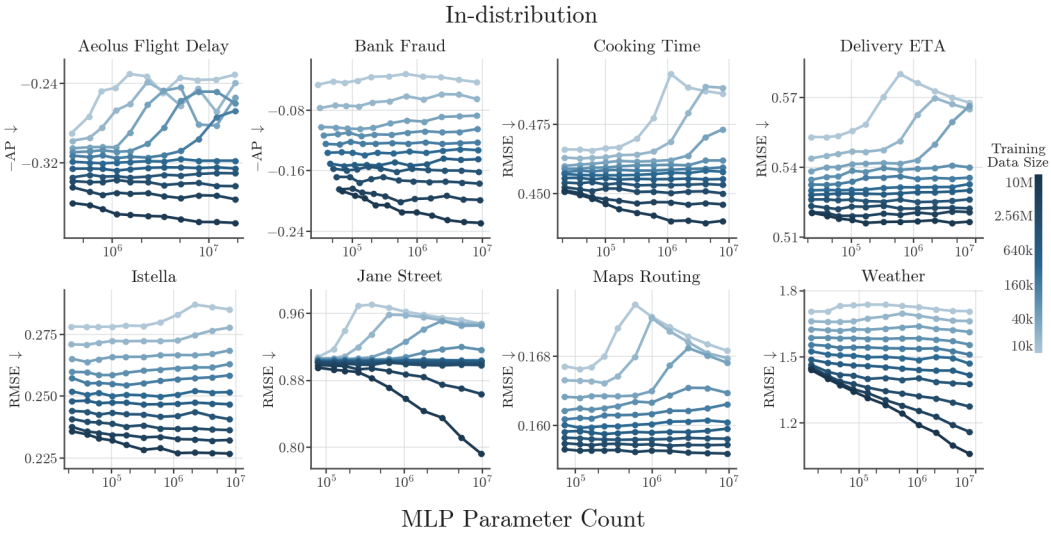

Figure 7: IID protocol: checkpoint and hyperparameter selection on IID validation, evaluation on IID test. Points show seed means; error bars show one standard deviation where multiple seeds are available.

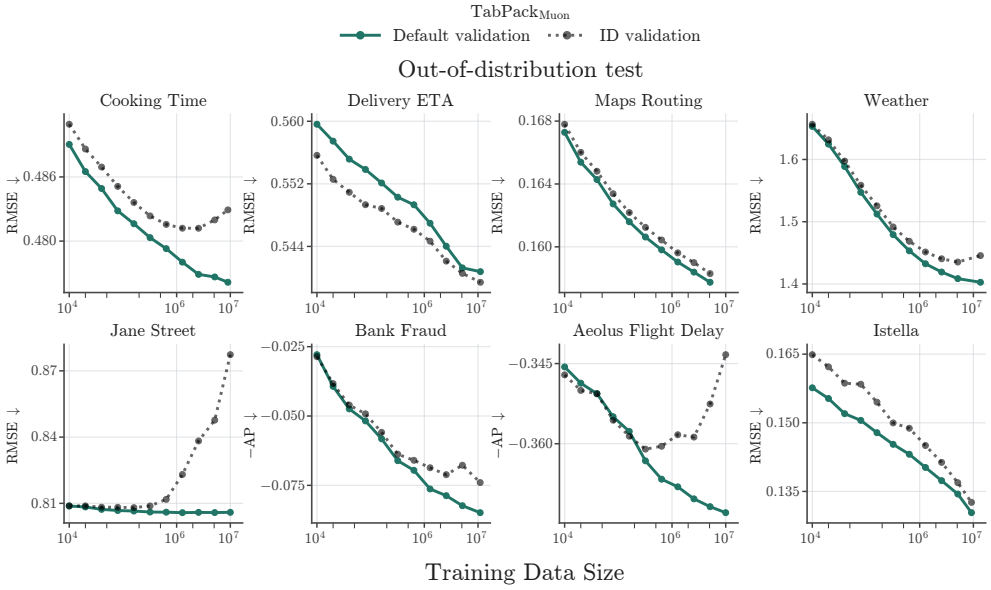

Figure 8: OOD test results: temporal test splits, or query for Istella. Green solid curves use default validation (temporal, or core for Istella); black dotted curves use ID validation.

Cooking Time / RMSE ↓

Method	Training set size										
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M	full
XGBoost	0.4893 ±0.0001	0.4873 ±0.0001	0.4853 ±0.0000	0.4844 ±0.0001	0.4827 ±0.0000	0.4816 ±0.0000	0.4803 ±0.0000	0.4790 ±0.0001	0.4781 ±0.0000	0.4778 ±0.0001	0.4769 ±0.0001
LightGBM	0.4891 ±0.0001	0.4876 ±0.0001	0.4854 ±0.0001	0.4847 ±0.0001	0.4828 ±0.0001	0.4816 ±0.0001	0.4801 ±0.0001	0.4791 ±0.0001	0.4780 ±0.0001	0.4777 ±0.0001	0.4773 ±0.0001
$\text{MLP}_{\text{AdamW}}$	0.4881 ±0.0004	0.4862 ±0.0004	0.4847 ±0.0004	0.4841 ±0.0004	0.4823 ±0.0003	0.4816 ±0.0002	0.4810 ±0.0004	0.4801 ±0.0004	0.4788 ±0.0004	0.4784 ±0.0007	0.4785 ±0.0006
MLP_{Muon}	0.4897 ±0.0005	0.4873 ±0.0002	0.4857 ±0.0002	0.4847 ±0.0003	0.4828 ±0.0003	0.4817 ±0.0003	0.4805 ±0.0004	0.4791 ±0.0003	0.4780 ±0.0005	0.4777 ±0.0006	0.4764 ±0.0002
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	0.4871 ±0.0004	*0.4853* ±0.0003	*0.4836* ±0.0003	0.4824 ±0.0004	0.4818 ±0.0004	0.4810 ±0.0006	0.4798 ±0.0003	0.4794 ±0.0008	0.4778 ±0.0004	0.4778 ±0.0005	0.4770 ±0.0004
$\text{MLP}^{\ddagger}_{\text{Muon}}$	*0.4864* ±0.0010	**0.4842** ±0.0003	*0.4834* ±0.0001	0.4831 ±0.0010	*0.4810* ±0.0004	0.4810 ±0.0004	*0.4792* ±0.0005	0.4790 ±0.0004	0.4780 ±0.0005	0.4778 ±0.0003	0.4767 ±0.0004
$\text{TabM}_{\text{Muon}}$	**0.4857** ±0.0004	**0.4840** ±0.0004	**0.4829** ±0.0004	*0.4816* ±0.0003	0.4814 ±0.0004	*0.4801* ±0.0003	*0.4794* ±0.0003	0.4783 ±0.0003	0.4778 ±0.0003	0.4774 ±0.0003	0.4772 ±0.0002

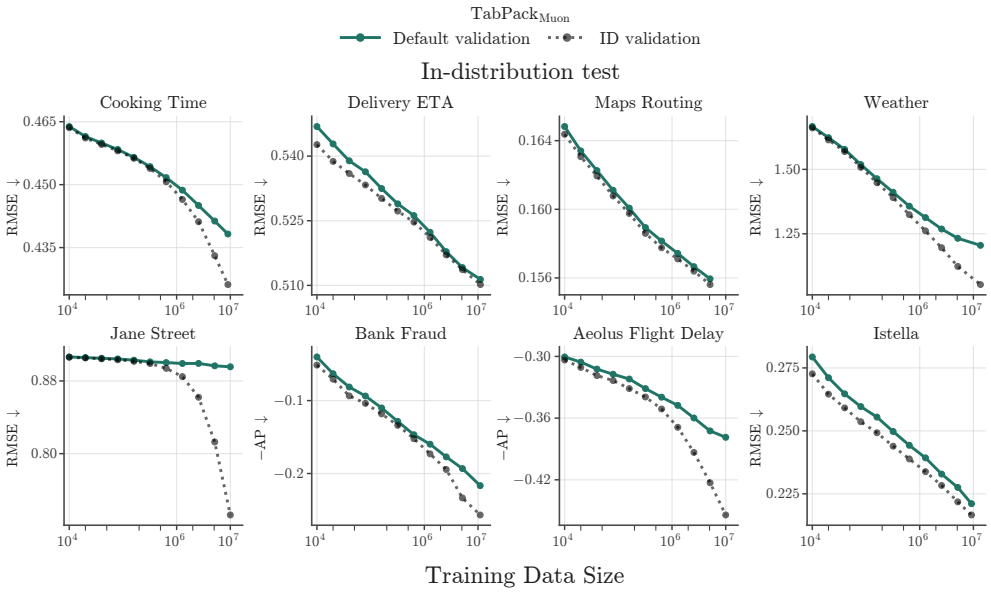

Figure 9: In-distribution test results: both validation protocols are evaluated on IID test. Colors and line styles follow Figure 8.

Delivery ETA / RMSE ↓

Method	Training set size										
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M	full
XGBoost	*0.5549* ±0.0001	*0.5529* ±0.0001	*0.5528* ±0.0001	0.5503 ±0.0001	**0.5491** ±0.0001	*0.5458* ±0.0001	*0.5442* ±0.0002	*0.5422* ±0.0001	**0.5406** ±0.0001	*0.5394* ±0.0001	**0.5380** ±0.0001
LightGBM	**0.5542** ±0.0001	**0.5527** ±0.0001	**0.5520** ±0.0001	**0.5492** ±0.0001	*0.5493* ±0.0001	**0.5454** ±0.0001	**0.5437** ±0.0001	**0.5420** ±0.0002	*0.5407* ±0.0001	0.5395 ±0.0001	0.5389 ±0.0001
$\text{MLP}_{\text{AdamW}}$	0.5589 ±0.0017	0.5551 ±0.0006	**0.5524** ±0.0009	0.5517 ±0.0009	0.5510 ±0.0008	0.5493 ±0.0006	0.5492 ±0.0005	0.5488 ±0.0012	0.5479 ±0.0013	0.5461 ±0.0011	0.5446 ±0.0010
MLP_{Muon}	0.5567 ±0.0005	0.5573 ±0.0016	0.5549 ±0.0008	0.5524 ±0.0011	0.5519 ±0.0008	0.5492 ±0.0005	0.5494 ±0.0009	0.5474 ±0.0006	0.5456 ±0.0009	0.5443 ±0.0016	0.5436 ±0.0011
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	0.5570 ±0.0018	0.5583 ±0.0025	0.5564 ±0.0019	0.5566 ±0.0017	0.5549 ±0.0010	0.5529 ±0.0015	0.5503 ±0.0016	0.5469 ±0.0015	0.5476 ±0.0029	0.5463 ±0.0015	0.5434 ±0.0020
$\text{MLP}^{\ddagger}_{\text{Muon}}$	0.5573 ±0.0028	0.5574 ±0.0009	0.5552 ±0.0011	0.5544 ±0.0009	0.5531 ±0.0016	0.5496 ±0.0014	0.5477 ±0.0012	0.5483 ±0.0027	0.5446 ±0.0011	0.5433 ±0.0013	0.5422 ±0.0010
$\text{TabM}_{\text{Muon}}$	0.5575 ±0.0021	0.5547 ±0.0010	0.5541 ±0.0015	*0.5499* ±0.0004	0.5513 ±0.0010	0.5497 ±0.0008	0.5466 ±0.0005	0.5434 ±0.0009	0.5413 ±0.0005	**0.5387** ±0.0003	*0.5384* ±0.0002
$\text{TabPack}_{\text{Muon}}$	0.5596 ±0.0009	0.5575 ±0.0007	0.5552 ±0.0007	0.5538 ±0.0006	0.5521 ±0.0005	0.5503 ±0.0003	0.5493 ±0.0004	0.5470 ±0.0004	0.5440 ±0.0006	0.5413 ±0.0005	0.5408 ±0.0003
$\text{TabPack}^{\dagger}_{\text{Muon}}$	0.5596 ±0.0004	0.5578 ±0.0007	0.5563 ±0.0007	0.5527 ±0.0007	0.5514 ±0.0004	0.5479 ±0.0004	0.5459 ±0.0004	0.5432 ±0.0004	0.5420 ±0.0003	0.5407 ±0.0004	0.5407 ±0.0004

Maps Routing / RMSE ↓

Method	Training set size									
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M
XGBoost	0.1665 ±0.0000	0.1649 ±0.0000	0.1640 ±0.0000	0.1627 ±0.0000	0.1619 ±0.0000	0.1611 ±0.0000	0.1604 ±0.0000	0.1598 ±0.0000	0.1591 ±0.0000	0.1585 ±0.0001
LightGBM	0.1667 ±0.0001	*0.1649* ±0.0000	0.1639 ±0.0000	0.1626 ±0.0000	0.1617 ±0.0000	0.1610 ±0.0000	0.1603 ±0.0001	0.1597 ±0.0000	0.1592 ±0.0000	0.1587 ±0.0000
$\text{MLP}_{\text{AdamW}}$	0.1679 ±0.0002	0.1673 ±0.0004	0.1673 ±0.0011	0.1636 ±0.0001	0.1627 ±0.0003	0.1623 ±0.0003	0.1613 ±0.0001	0.1604 ±0.0001	0.1600 ±0.0003	0.1591 ±0.0001
MLP_{Muon}	0.1678 ±0.0002	0.1660 ±0.0001	0.1648 ±0.0001	0.1636 ±0.0002	0.1622 ±0.0001	0.1612 ±0.0001	0.1604 ±0.0000	0.1597 ±0.0001	0.1591 ±0.0001	0.1586 ±0.0002
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	0.1675 ±0.0002	0.1662 ±0.0003	0.1646 ±0.0004	0.1633 ±0.0001	0.1621 ±0.0002	0.1612 ±0.0001	0.1604 ±0.0001	0.1595 ±0.0002	0.1591 ±0.0001	0.1584 ±0.0001
$\text{MLP}^{\ddagger}_{\text{Muon}}$	0.1668 ±0.0001	0.1657 ±0.0002	0.1642 ±0.0001	0.1632 ±0.0003	0.1617 ±0.0001	0.1608 ±0.0001	0.1607 ±0.0002	0.1598 ±0.0002	0.1589 ±0.0001	0.1582 ±0.0001
$\text{TabM}_{\text{Muon}}$	*0.1664* ±0.0001	*0.1648* ±0.0002	*0.1637* ±0.0001	*0.1622* ±0.0001	*0.1613* ±0.0001	*0.1606* ±0.0001	*0.1598* ±0.0001	0.1592 ±0.0001	0.1586 ±0.0001	*0.1580* ±0.0001
$\text{TabPack}_{\text{Muon}}$	0.1673 ±0.0001	0.1654 ±0.0001	0.1643 ±0.0001	0.1627 ±0.0001	0.1616 ±0.0000	*0.1606* ±0.0001	*0.1598* ±0.0001	*0.1590* ±0.0001	*0.1584* ±0.0001	**0.1577** ±0.0000
$\text{TabPack}^{\dagger}_{\text{Muon}}$	**0.1655** ±0.0001	**0.1640** ±0.0001	**0.1629** ±0.0001	**0.1617** ±0.0001	**0.1607** ±0.0001	**0.1599** ±0.0001	**0.1591** ±0.0001	**0.1585** ±0.0000	**0.1578** ±0.0001	0.1574 ±0.0000

Weather / RMSE ↓

Method	Training set size										
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M	full
XGBoost	*1.6161* ±0.0006	1.5875 ±0.0007	1.5582 ±0.0006	1.5232 ±0.0008	1.4984 ±0.0009	1.4716 ±0.0006	1.4466 ±0.0005	1.4318 ±0.0004	1.4221 ±0.0004	1.4136 ±0.0014	1.4073 ±0.0007
LightGBM	1.6208 ±0.0007	*1.5848* ±0.0011	*1.5520* ±0.0011	*1.5156* ±0.0013	*1.4952* ±0.0008	1.4690 ±0.0007	1.4512 ±0.0017	1.4330 ±0.0008	1.4250 ±0.0007	1.4154 ±0.0011	1.4128 ±0.0000
$\text{MLP}_{\text{AdamW}}$	1.6680 ±0.0051	1.6425 ±0.0029	1.6157 ±0.0035	1.5841 ±0.0040	1.5654 ±0.0044	1.5306 ±0.0067	1.5040 ±0.0042	1.4861 ±0.0050	1.4725 ±0.0052	1.4585 ±0.0057	1.4448 ±0.0030
MLP_{Muon}	1.6647 ±0.0026	1.6375 ±0.0017	1.6035 ±0.0030	1.5720 ±0.0033	1.5403 ±0.0036	1.5061 ±0.0025	1.4851 ±0.0041	1.4627 ±0.0033	1.4511 ±0.0027	1.4406 ±0.0018	1.4433 ±0.0031
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	1.6374 ±0.0043	1.6133 ±0.0030	1.5881 ±0.0042	1.5670 ±0.0032	1.5412 ±0.0029	1.5227 ±0.0032	1.5045 ±0.0051	1.4909 ±0.0042	1.4692 ±0.0023	1.4495 ±0.0043	1.4389 ±0.0034
$\text{MLP}^{\ddagger}_{\text{Muon}}$	1.6292 ±0.0057	1.6045 ±0.0042	1.5762 ±0.0013	1.5526 ±0.0039	1.5286 ±0.0023	1.5028 ±0.0032	1.4855 ±0.0043	1.4663 ±0.0037	1.4491 ±0.0018	1.4433 ±0.0056	1.4334 ±0.0041
$\text{TabM}_{\text{Muon}}$	*1.6143* ±0.0031	1.5939 ±0.0033	1.5603 ±0.0042	1.5273 ±0.0038	*1.4974* ±0.0037	*1.4607* ±0.0030	*1.4335* ±0.0034	*1.4156* ±0.0016	*1.4078* ±0.0039	*1.3982* ±0.0048	*1.3968* ±0.0047
$\text{TabPack}_{\text{Muon}}$	1.6531 ±0.0012	1.6246 ±0.0016	1.5887 ±0.0015	1.5473 ±0.0024	1.5123 ±0.0006	1.4793 ±0.0013	1.4532 ±0.0012	1.4324 ±0.0017	1.4192 ±0.0017	1.4086 ±0.0010	1.4027 ±0.0012
$\text{TabPack}^{\dagger}_{\text{Muon}}$	**1.5934** ±0.0024	**1.5683** ±0.0009	**1.5379** ±0.0013	**1.5049** ±0.0013	**1.4785** ±0.0017	**1.4448** ±0.0017	**1.4241** ±0.0011	**1.4090** ±0.0013	**1.3980** ±0.0011	**1.3890** ±0.0011	**1.3852** ±0.0016

Jane Street / RMSE ↓

Method	Training set size										
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M	full
XGBoost	0.8090 ±0.0001	*0.8087* ±0.0001	0.8081 ±0.0001	0.8075 ±0.0001	0.8084 ±0.0001	0.8080 ±0.0001	0.8078 ±0.0001	0.8073 ±0.0001	0.8068 ±0.0001	0.8072 ±0.0000	0.8062 ±0.0002
LightGBM	0.8090 ±0.0001	0.8088 ±0.0000	0.8083 ±0.0001	0.8080 ±0.0000	0.8080 ±0.0001	0.8080 ±0.0001	0.8076 ±0.0001	0.8075 ±0.0001	0.8069 ±0.0001	0.8067 ±0.0001	0.8063 ±0.0000
$\text{MLP}_{\text{AdamW}}$	0.8095 ±0.0006	0.8091 ±0.0004	0.8083 ±0.0003	0.8080 ±0.0003	0.8077 ±0.0005	0.8070 ±0.0005	0.8066 ±0.0003	0.8061 ±0.0002	0.8062 ±0.0002	0.8067 ±0.0003	0.8061 ±0.0002
MLP_{Muon}	0.8089 ±0.0001	0.8090 ±0.0004	0.8082 ±0.0001	0.8075 ±0.0001	0.8071 ±0.0003	0.8068 ±0.0002	*0.8062* ±0.0003	0.8060 ±0.0002	*0.8060* ±0.0002	0.8061 ±0.0002	0.8062 ±0.0003
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	0.8091 ±0.0002	0.8089 ±0.0002	0.8081 ±0.0004	**0.8070** ±0.0004	*0.8066* ±0.0002	0.8063 ±0.0002	0.8069 ±0.0006	0.8063 ±0.0003	0.8062 ±0.0003	0.8065 ±0.0003	0.8067 ±0.0005
$\text{MLP}^{\ddagger}_{\text{Muon}}$	*0.8088* ±0.0001	**0.8085** ±0.0002	0.8080 ±0.0002	0.8078 ±0.0002	0.8073 ±0.0003	0.8068 ±0.0006	*0.8062* ±0.0003	*0.8059* ±0.0003	**0.8058** ±0.0002	*0.8059* ±0.0002	0.8061 ±0.0003
$\text{TabM}_{\text{Muon}}$	**0.8087** ±0.0000	*0.8086* ±0.0001	**0.8074** ±0.0003	**0.8067** ±0.0002	**0.8063** ±0.0001	**0.8058** ±0.0002	0.8062 ±0.0002	*0.8059* ±0.0001	*0.8059* ±0.0001	**0.8058** ±0.0001	*0.8059* ±0.0001
$\text{TabPack}_{\text{Muon}}$	*0.8088* ±0.0001	**0.8084** ±0.0001	**0.8073** ±0.0002	**0.8068** ±0.0002	*0.8065* ±0.0001	0.8061 ±0.0001	*0.8060* ±0.0001	*0.8058* ±0.0001	*0.8059* ±0.0001	*0.8058* ±0.0001	*0.8059* ±0.0002
$\text{TabPack}^{\dagger}_{\text{Muon}}$	0.8089 ±0.0001	**0.8084** ±0.0001	*0.8077* ±0.0001	*0.8071* ±0.0001	*0.8067* ±0.0003	*0.8059* ±0.0001	**0.8056** ±0.0001	**0.8056** ±0.0001	**0.8057** ±0.0002	**0.8056** ±0.0002	**0.8056** ±0.0001

Bank Fraud Datafusion / AP ↑

Method	Training set size										
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M	full
XGBoost	0.0293 ±0.0015	0.0423 ±0.0012	0.0542 ±0.0007	0.0622 ±0.0005	0.0676 ±0.0004	0.0697 ±0.0003	0.0738 ±0.0004	0.0772 ±0.0001	0.0795 ±0.0003	0.0829 ±0.0004	*0.0849* ±0.0017
LightGBM	0.0254 ±0.0013	*0.0461* ±0.0003	**0.0572** ±0.0003	0.0635 ±0.0003	*0.0686* ±0.0003	*0.0718* ±0.0004	0.0755 ±0.0005	0.0785 ±0.0002	0.0806 ±0.0004	*0.0844* ±0.0004	**0.0870** ±0.0007
$\text{MLP}_{\text{AdamW}}$	**0.0345** ±0.0017	*0.0458* ±0.0016	*0.0566* ±0.0008	0.0632 ±0.0011	0.0659 ±0.0011	0.0693 ±0.0012	0.0713 ±0.0018	0.0762 ±0.0011	0.0790 ±0.0011	0.0822 ±0.0009	0.0829 ±0.0006
MLP_{Muon}	*0.0307* ±0.0017	0.0378 ±0.0011	0.0459 ±0.0011	0.0507 ±0.0009	0.0599 ±0.0013	0.0667 ±0.0015	0.0657 ±0.0008	0.0709 ±0.0012	0.0751 ±0.0008	0.0792 ±0.0012	0.0809 ±0.0012
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	*0.0299* ±0.0032	*0.0465* ±0.0018	*0.0555* ±0.0015	0.0625 ±0.0015	0.0670 ±0.0014	*0.0722* ±0.0014	0.0755 ±0.0013	0.0782 ±0.0011	0.0784 ±0.0018	0.0814 ±0.0028	0.0825 ±0.0023
$\text{MLP}^{\ddagger}_{\text{Muon}}$	0.0283 ±0.0016	0.0420 ±0.0008	0.0484 ±0.0007	0.0524 ±0.0013	0.0644 ±0.0008	0.0698 ±0.0008	0.0678 ±0.0021	0.0769 ±0.0012	0.0773 ±0.0008	0.0786 ±0.0022	*0.0836* ±0.0012
$\text{TabM}_{\text{Muon}}$	0.0264 ±0.0051	*0.0449* ±0.0023	**0.0570** ±0.0009	**0.0669** ±0.0009	**0.0710** ±0.0008	**0.0758** ±0.0006	**0.0789** ±0.0005	**0.0825** ±0.0009	**0.0839** ±0.0005	**0.0853** ±0.0010	**0.0876** ±0.0009
$\text{TabPack}_{\text{Muon}}$	0.0279 ±0.0021	0.0393 ±0.0013	0.0474 ±0.0014	0.0518 ±0.0009	0.0582 ±0.0013	0.0661 ±0.0013	0.0696 ±0.0010	0.0763 ±0.0007	0.0787 ±0.0008	0.0823 ±0.0007	*0.0848* ±0.0007
$\text{TabPack}^{\dagger}_{\text{Muon}}$	*0.0317* ±0.0017	**0.0489** ±0.0015	**0.0576** ±0.0006	*0.0647* ±0.0008	**0.0704** ±0.0008	**0.0757** ±0.0006	*0.0777* ±0.0007	*0.0800* ±0.0007	*0.0820* ±0.0007	**0.0848** ±0.0004	**0.0871** ±0.0005

Aeolus Flight Delay / AP ↑

Method	Training set size										
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M	full
XGBoost	*0.3503* ±0.0005	**0.3551** ±0.0004	**0.3562** ±0.0003	**0.3581** ±0.0003	**0.3628** ±0.0005	0.3630 ±0.0003	0.3656 ±0.0002	0.3655 ±0.0002	0.3648 ±0.0002	0.3689 ±0.0002	0.3712 ±0.0005
LightGBM	0.3440 ±0.0020	0.3513 ±0.0011	0.3498 ±0.0008	0.3541 ±0.0006	0.3572 ±0.0007	0.3601 ±0.0006	0.3597 ±0.0003	0.3612 ±0.0004	0.3651 ±0.0002	0.3668 ±0.0002	0.3697 ±0.0000
$\text{MLP}_{\text{AdamW}}$	0.3310 ±0.0016	0.3358 ±0.0023	0.3410 ±0.0014	0.3431 ±0.0013	0.3495 ±0.0017	0.3499 ±0.0010	0.3556 ±0.0019	0.3591 ±0.0011	0.3607 ±0.0022	0.3658 ±0.0009	0.3671 ±0.0018
MLP_{Muon}	0.3458 ±0.0009	0.3492 ±0.0009	0.3506 ±0.0008	0.3518 ±0.0013	0.3551 ±0.0020	0.3618 ±0.0012	0.3641 ±0.0008	0.3663 ±0.0011	*0.3696* ±0.0008	0.3696 ±0.0013	*0.3726* ±0.0009
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	0.3441 ±0.0015	0.3451 ±0.0025	0.3447 ±0.0034	0.3520 ±0.0005	0.3548 ±0.0018	0.3592 ±0.0016	0.3639 ±0.0023	*0.3683* ±0.0021	0.3684 ±0.0012	*0.3698* ±0.0028	*0.3728* ±0.0018
$\text{MLP}^{\ddagger}_{\text{Muon}}$	**0.3510** ±0.0022	*0.3537* ±0.0017	*0.3533* ±0.0031	**0.3582** ±0.0025	**0.3621** ±0.0028	**0.3659** ±0.0014	**0.3691** ±0.0019	**0.3706** ±0.0010	**0.3722** ±0.0010	**0.3741** ±0.0012	*0.3727* ±0.0010
$\text{TabM}_{\text{Muon}}$	0.3468 ±0.0009	0.3487 ±0.0016	0.3508 ±0.0014	0.3532 ±0.0011	0.3604 ±0.0015	*0.3653* ±0.0011	**0.3681** ±0.0010	*0.3696* ±0.0008	0.3693 ±0.0008	0.3686 ±0.0007	0.3699 ±0.0010
$\text{TabPack}_{\text{Muon}}$	0.3456 ±0.0012	0.3487 ±0.0014	0.3506 ±0.0010	*0.3549* ±0.0010	0.3577 ±0.0009	0.3632 ±0.0009	*0.3666* ±0.0010	0.3681 ±0.0009	*0.3703* ±0.0011	*0.3718* ±0.0011	*0.3729* ±0.0007
$\text{TabPack}^{\dagger}_{\text{Muon}}$	**0.3522** ±0.0012	**0.3549** ±0.0009	*0.3548* ±0.0009	**0.3569** ±0.0016	*0.3620* ±0.0009	**0.3669** ±0.0007	**0.3690** ±0.0012	**0.3697** ±0.0010	**0.3721** ±0.0009	**0.3732** ±0.0008	**0.3739** ±0.0012

Istella / RMSE ↓

Method	Training set size										
	10k	20k	40k	80k	160k	320k	640k	1.28M	2.56M	5.12M	full
XGBoost	0.1661 ±0.0004	0.1639 ±0.0003	0.1619 ±0.0002	0.1594 ±0.0002	0.1542 ±0.0002	0.1477 ±0.0002	*0.1427* ±0.0004	0.1397 ±0.0005	0.1381 ±0.0003	0.1332 ±0.0005	0.1319 ±0.0004
LightGBM	0.1678 ±0.0003	0.1692 ±0.0010	0.1613 ±0.0003	0.1592 ±0.0005	0.1533 ±0.0002	0.1481 ±0.0005	0.1430 ±0.0002	0.1397 ±0.0003	0.1369 ±0.0006	0.1321 ±0.0002	0.1294 ±0.0003
$\text{MLP}_{\text{AdamW}}$	0.1589 ±0.0015	0.1568 ±0.0010	0.1550 ±0.0012	0.1533 ±0.0009	0.1517 ±0.0010	0.1490 ±0.0012	0.1467 ±0.0010	0.1442 ±0.0010	0.1427 ±0.0017	0.1382 ±0.0010	0.1342 ±0.0011
MLP_{Muon}	0.1602 ±0.0015	0.1589 ±0.0064	0.1529 ±0.0009	0.1512 ±0.0005	0.1503 ±0.0008	0.1467 ±0.0005	0.1464 ±0.0005	0.1428 ±0.0007	0.1392 ±0.0003	0.1363 ±0.0003	0.1341 ±0.0006
$\text{MLP}^{\ddagger}_{\text{AdamW}}$	*0.1582* ±0.0008	*0.1553* ±0.0011	*0.1517* ±0.0008	*0.1500* ±0.0006	0.1497 ±0.0005	0.1477 ±0.0011	0.1446 ±0.0014	0.1417 ±0.0011	0.1373 ±0.0008	0.1351 ±0.0010	0.1302 ±0.0011
$\text{MLP}^{\ddagger}_{\text{Muon}}$	0.1603 ±0.0027	0.1575 ±0.0021	0.1555 ±0.0017	0.1525 ±0.0007	0.1519 ±0.0021	*0.1445* ±0.0013	0.1468 ±0.0017	*0.1386* ±0.0007	*0.1355* ±0.0010	*0.1315* ±0.0004	*0.1280* ±0.0008
$\text{TabM}_{\text{Muon}}$	*0.1579* ±0.0010	0.1562 ±0.0006	0.1532 ±0.0004	0.1520 ±0.0002	0.1494 ±0.0006	*0.1454* ±0.0006	0.1434 ±0.0003	0.1395 ±0.0004	*0.1357* ±0.0004	0.1322 ±0.0010	*0.1278* ±0.0007
$\text{TabPack}_{\text{Muon}}$	*0.1576* ±0.0004	*0.1553* ±0.0006	*0.1520* ±0.0003	*0.1505* ±0.0005	*0.1478* ±0.0005	*0.1453* ±0.0003	0.1431 ±0.0003	0.1402 ±0.0003	0.1374 ±0.0003	0.1344 ±0.0004	0.1304 ±0.0002
$\text{TabPack}^{\dagger}_{\text{Muon}}$	**0.1570** ±0.0007	**0.1536** ±0.0008	**0.1499** ±0.0006	**0.1477** ±0.0008	**0.1428** ±0.0006	**0.1381** ±0.0003	**0.1353** ±0.0004	**0.1316** ±0.0003	**0.1281** ±0.0005	**0.1226** ±0.0005	**0.1191** ±0.0002

G FULL RESULTS FOR PFN METHODS

Primary-test means and sample standard deviations. —: no result. **Bold** marks first place and *italics* second place among multi-seed runs at each size (sequential Welch tests, two-sided unadjusted $p < 0.05$). Sizes without any PFN result are omitted.

Cooking Time / RMSE ↓

Method	Training set size							
	10k	20k	40k	80k	160k	320k	640k	1.28M
TabICL v2	*0.4940* ±0.0006	*0.4920* ±0.0006	*0.4911* ±0.0006	0.4894 ±0.0006	0.4890 ±0.0006	0.4883 ±0.0006	0.4881 ±0.0006	—
TabPFN v3	**0.4917** ±0.0005	**0.4897** ±0.0002	**0.4886** ±0.0002	*0.4878* ±0.0003	*0.4873* ±0.0002	*0.4872* ±0.0003	0.4870 ±0.0004	**0.4835** ±0.0003
TabPFN v3.5 fast	0.4961 ±0.0015	0.4941 ±0.0017	*0.4920* ±0.0019	0.4906 ±0.0017	0.4891 ±0.0014	*0.4875* ±0.0012	*0.4862* ±0.0011	—
TabPFN v3.5	**0.4909** ±0.0012	**0.4895** ±0.0013	**0.4878** ±0.0012	**0.4866** ±0.0012	**0.4858** ±0.0011	**0.4847** ±0.0009	**0.4842** ±0.0009	**0.4841** ±0.0005

Delivery ETA / RMSE ↓

Method	Training set size							
	10k	20k	40k	80k	160k	320k	640k	1.28M
TabICL v2	*0.5548* ±0.0005	0.5554 ±0.0003	0.5555 ±0.0004	0.5543 ±0.0004	0.5553 ±0.0006	0.5544 ±0.0006	0.5552 ±0.0008	—
TabPFN v3	*0.5552* ±0.0005	*0.5546* ±0.0004	0.5534 ±0.0004	0.5526 ±0.0002	0.5531 ±0.0004	0.5528 ±0.0004	0.5542 ±0.0008	*0.5515* ±0.0006
TabPFN v3.5 fast	**0.5523** ±0.0004	**0.5503** ±0.0006	*0.5500* ±0.0006	*0.5481* ±0.0006	*0.5478* ±0.0006	*0.5463* ±0.0007	*0.5454* ±0.0007	—
TabPFN v3.5	**0.5519** ±0.0005	**0.5499** ±0.0004	**0.5495** ±0.0005	**0.5472** ±0.0006	**0.5466** ±0.0007	**0.5446** ±0.0005	**0.5431** ±0.0005	**0.5413** ±0.0004

Maps Routing / RMSE ↓

Method	Training set size							
	10k	20k	40k	80k	160k	320k	640k	1.28M
TabICL v2	0.1680 ±0.0003	0.1670 ±0.0003	0.1670 ±0.0003	0.1663 ±0.0003	0.1662 ±0.0003	0.1663 ±0.0003	0.1664 ±0.0003	—
TabPFN v3	0.1672 ±0.0002	0.1661 ±0.0001	0.1660 ±0.0001	0.1653 ±0.0001	0.1650 ±0.0001	0.1649 ±0.0001	0.1634 ±0.0008	*0.1628* ±0.0001
TabPFN v3.5 fast	*0.1661* ±0.0002	*0.1648* ±0.0001	*0.1643* ±0.0001	*0.1635* ±0.0001	*0.1629* ±0.0001	*0.1626* ±0.0001	*0.1623* ±0.0001	—
TabPFN v3.5	**0.1654** ±0.0001	**0.1642** ±0.0001	**0.1635** ±0.0001	**0.1628** ±0.0001	**0.1624** ±0.0000	**0.1620** ±0.0000	**0.1616** ±0.0001	**0.1612** ±0.0001

Weather / RMSE ↓

Method	Training set size							
	10k	20k	40k	80k	160k	320k	640k	1.28M
TabICL v2	1.6205 ±0.0042	1.6064 ±0.0039	1.5815 ±0.0039	1.5530 ±0.0036	1.5252 ±0.0035	1.5000 ±0.0032	1.4797 ±0.0033	—
TabPFN v3	*1.5968* ±0.0021	*1.5785* ±0.0023	*1.5550* ±0.0020	1.5266 ±0.0024	1.5030 ±0.0016	1.4831 ±0.0017	*1.4501* ±0.0022	*1.4489* ±0.0024
TabPFN v3.5 fast	1.6066 ±0.0041	1.5829 ±0.0041	*1.5543* ±0.0043	*1.5216* ±0.0047	*1.4939* ±0.0052	*1.4678* ±0.0049	*1.4500* ±0.0045	—
TabPFN v3.5	**1.5826** ±0.0037	**1.5630** ±0.0041	**1.5275** ±0.0052	**1.4917** ±0.0056	**1.4628** ±0.0057	**1.4364** ±0.0062	**1.4208** ±0.0057	**1.4172** ±0.0064

Jane Street / RMSE ↓

Method	Training set size							
	10k	20k	40k	80k	160k	320k	640k	1.28M
TabICL v2	0.8104 ±0.0002	0.8091 ±0.0001	0.8082 ±0.0001	0.8081 ±0.0002	0.8090 ±0.0003	0.8099 ±0.0004	*0.8115* ±0.0004	—
TabPFN v3	**0.8088** ±0.0000	**0.8084** ±0.0000	0.8078 ±0.0001	0.8074 ±0.0001	0.8075 ±0.0001	0.8072 ±0.0001	**0.8061** ±0.0001	**0.8061** ±0.0001
TabPFN v3.5 fast	*0.8092* ±0.0001	*0.8085* ±0.0001	**0.8071** ±0.0001	**0.8065** ±0.0001	**0.8063** ±0.0001	**0.8060** ±0.0001	**0.8061** ±0.0001	—
TabPFN v3.5	0.8099 ±0.0001	0.8089 ±0.0001	*0.8075* ±0.0001	*0.8068* ±0.0001	*0.8065* ±0.0001	*0.8061* ±0.0001	**0.8061** ±0.0001	**0.8059** ±0.0002

Bank Fraud Datafusion / AP ↑

Method	Training set size							
	10k	20k	40k	80k	160k	320k	640k	1.28M
TabICL v2	0.0263 ±0.0014	0.0421 ±0.0006	0.0523 ±0.0007	0.0542 ±0.0005	0.0556 ±0.0004	0.0583 ±0.0006	0.0596 ±0.0006	—
TabPFN v3	0.0257 ±0.0017	*0.0443* ±0.0013	0.0528 ±0.0012	0.0573 ±0.0005	0.0612 ±0.0019	0.0659 ±0.0016	0.0667 ±0.0017	*0.0682* ±0.0017
TabPFN v3.5 fast	*0.0317* ±0.0007	0.0417 ±0.0009	*0.0555* ±0.0007	*0.0604* ±0.0014	*0.0641* ±0.0014	*0.0713* ±0.0014	*0.0727* ±0.0013	—
TabPFN v3.5	**0.0339** ±0.0009	**0.0473** ±0.0012	**0.0611** ±0.0007	**0.0667** ±0.0007	**0.0697** ±0.0007	**0.0779** ±0.0005	**0.0793** ±0.0006	**0.0811** ±0.0008

Aeolus Flight Delay / AP ↑

Method	Training set size							
	10k	20k	40k	80k	160k	320k	640k	1.28M
TabICL v2	0.3486 ±0.0014	0.3525 ±0.0010	0.3523 ±0.0015	0.3545 ±0.0018	0.3566 ±0.0022	0.3571 ±0.0026	0.3559 ±0.0035	—
TabPFN v3	*0.3541* ±0.0003	*0.3582* ±0.0004	*0.3576* ±0.0005	*0.3598* ±0.0006	*0.3626* ±0.0008	*0.3642* ±0.0008	*0.3640* ±0.0008	**0.3648** ±0.0010
TabPFN v3.5 fast	**0.3554** ±0.0006	**0.3588** ±0.0005	**0.3590** ±0.0008	**0.3607** ±0.0010	**0.3646** ±0.0009	**0.3665** ±0.0009	**0.3663** ±0.0010	—
TabPFN v3.5	**0.3559** ±0.0006	**0.3591** ±0.0003	**0.3592** ±0.0004	**0.3612** ±0.0006	**0.3651** ±0.0004	**0.3662** ±0.0008	**0.3659** ±0.0011	**0.3644** ±0.0009

Istella / RMSE ↓

Method	Training set size						
	10k	20k	40k	80k	160k	320k	640k
TabICL v2	0.1655 ±0.0007	0.1654 ±0.0009	0.1610 ±0.0006	0.1601 ±0.0006	0.1593 ±0.0009	0.1584 ±0.0011	0.1594 ±0.0013
TabPFN v3	0.1630 ±0.0005	0.1622 ±0.0005	0.1630 ±0.0005	0.1634 ±0.0004	0.1646 ±0.0002	0.1636 ±0.0028	0.1603 ±0.0003
TabPFN v3.5 fast	*0.1606* ±0.0006	*0.1580* ±0.0009	*0.1561* ±0.0006	*0.1560* ±0.0007	*0.1539* ±0.0005	*0.1519* ±0.0005	*0.1507* ±0.0005
TabPFN v3.5	**0.1593** ±0.0008	**0.1535** ±0.0017	**0.1441** ±0.0010	**0.1402** ±0.0009	**0.1379** ±0.0007	**0.1366** ±0.0004	**0.1353** ±0.0005