Transformers Also Struggle with Extrapolating Ill-Behaved Learning Curves

Adelina-Andreea Cazacu[1], Cheng Yan[1], Sayak Mukherjee[1], and Tom Viering[1]

Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

Abstract. Sample-wise learning curves capture how model performance scales with data, helping practitioners predict performance at larger data scales and allocate resources more effectively. While both parametric models and neural networks are used for learning curve extrapolation, their comparative advantages remain underexplored. In this work, we present a systematic comparison of Real Data Learning Curve Prior-Fitted Networks (Real Data LC-PFN) against three parametric models (POW4, MMF4, and WBL4) using the Learning Curves Database 1.1 (LCDB 1.1). Our analysis reveals that Real Data LC-PFN consistently achieves stronger extrapolation accuracy across diverse generalization scenarios, with notable advantages when only limited observations are available. However, while it handles the commonly observed, well-behaved monotone and convex curve shapes well, performance on ill-behaved learning curves, such as dipping, remains less competitive than parametric models. Our findings highlight the importance of context-aware model selection rather than universal approaches. All code used in our study is publicly available [1].

Keywords: Machine Learning · Learning Curve · LCDB · Extrapolation · Prior-Fitted Networks · Transformer · In-context Learning.

1 Introduction

Machine learning (ML) practitioners in both academic and industry mediums often face a critical question when designing an ML application: "*How much data is enough?*". Learning curves, which plot model performance as a function of training set size, are a fundamental tool for answering this question by revealing how performance scales with available data. Accurately modeling this relationship enables informed decision making on whether the desired accuracy targets are achievable within particular practical constraints, while also helping to reduce computational cost and environmental impact [5]. Unlike the more widely studied epoch-based learning curves that track performance over training iterations, sample-size learning curves enable practitioners to devise data collection strategies before substantial investments are made, even for models like K-Nearest-Neighbors that do not involve iterative training [13,10].

Recent advancements in this area have explored both parametric and neural network approaches to learning curve extrapolation. Parametric models, such as MMF4

[1] Code: https://github.com/adelinacazacu/Extrapolating-Learning-Curves-When-Do-Neural-Networks-Outperform-Parametric-Models-

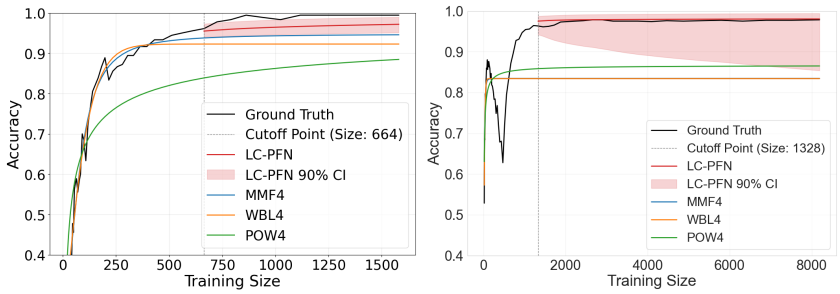


Fig. 1: Examples of learning curve extrapolation using parametric models (MMF4, WBL4 and POW4) and neural network (Real Data LC-PFN). Data points on the left-hand side of the cutoff (represented by the vertical dotted line) are observed by the model and then used to extrapolate the data points on the right-hand side.

and WBL4, have demonstrated strong performance across a wide range of learning scenarios [11], while neural network approaches such as Real Data Learning Curve Prior-Fitted Networks (Real Data LC-PFNs) [14] have shown promise in epoch-based learning curve extrapolation [1]. Examples of neural network versus parametric models are shown in Figure 1. However, Viering et al. [14] observed that these approaches have different strengths and weaknesses, particularly when generalizing to unseen datasets and learners. Yet, the specific conditions under which neural networks outperform parametric models remain poorly understood. This work challenges common assumptions about the automatic superiority of neural networks for complex, irregular patterns.

Moreover, the aforementioned comparative study has been conducted on LCDB 1.0 [11]. In a recent study, an extensive and higher-quality LCDB 1.1 [15] is proposed, which reveals significantly higher rates of ill-behaved learning curves characterized by non-monotonic patterns, plateaus, or sudden performance drops [13,9]. Specifically, [15] reports four typical shapes: (i) *flat* curves (showing minimal change with increase in training data size), (ii) *monotone & convex* curves (displaying smooth and asymptotic improvement; considered well-behaved, abbreviated as MonoConv), (iii) *dipping* curves (achieving best performance at intermediate training set sizes), and (iv) *peaking* curves (experiencing temporary performance decline before recovering). Thus, it is essential to re-evaluate the comparative performance of neural network methods against parametric models, systematically analyzing the conditions under which one takes precedence over the other. To address this gap, we ask:

> When do neural networks outperform parametric models in learning curve extrapolation?

In our study, we specifically focus on the transformer-based Real Data LC-PFN, which extended LC-PFN [12]. Unlike LC-PFN, which is trained on synthetic learning curves, Real Data LC-PFN leverages a data-driven prior by training on a real learning curve database. Thus, it is assumed to be better equipped to handle ill-behaved learning curves. We structure our analysis around the following three research questions while summarizing our contributions as answers to them:

RQ1: *How do parametric models and neural networks compare in learning curve extrapolation across diverse scenarios for generalization?*

We evaluate four generalization scenarios, where the models are tested on known datasets and known learners (KDKL), unseen datasets (UD), unseen learners (UL), and simultaneously unseen datasets and learners (UDUL). LC-PFN consistently outperforms parametric models in all scenarios, capturing complex curve dynamics beyond traditional functional forms. Its balanced performance across KDKL and out-of-distribution scenarios highlights robustness without overfitting.

RQ2: *How does the amount of the observed learning curve (the region before the cutoff) affect the extrapolation performance of parametric models and neural networks?*

We evaluate by varying the proportion of observed points prior to extrapolation. Real Data LC-PFN maintains strong predictive accuracy across all cutoff percentages, demonstrating reliable extrapolation even from very limited observed data.

RQ3: *How does the shape of the learning curve influence the relative performance of parametric models and neural networks in learning curve extrapolation?*

Real Data LC-PFN performs well on commonly occurring monotone convex learning curves but struggles with less frequent ill-behaved cases. Specifically, it offers a limited advantage on flat and peaking curves and fails on irregular dipping patterns. This outcome challenges the assumption that training on real learning curves equips Real Data LC-PFN with the ability to handle ill-behaved learning curves.

2 Related Work

In this section, we review the literature relevant to our comparative study.

Learning Curve Database. The Learning Curve Database (LCDB) 1.1 [15] is a large-scale collection of simple-wise learning curves spanning 265 OpenML classification tasks and 24 machine learning algorithms. It improves upon its predecessor, LCDB 1.0 [11], by mainly increasing the estimation points (also known as *anchor*) four times, observing more ill-behavior in learning curves. Thus, it can serve as a more challenging benchmark for evaluating different modeling methods.

Parametric Models. Many parametric approaches have been proposed in the literature for extrapolating learning curves from partial observations [13] by leveraging inductive biases in machine learning performance trends. While several empirical studies have attempted to identify effective models [3,2,6], each has notable limitations. To address this, [11] conducted a comprehensive comparison of parametric models. Three nonlinear functional forms, namely MMF4, WBL4, and POW4, are found to be significantly effective for extrapolating learning curves [8,4].

Neural Networks. Learning Curve Prior-Data Fitted Network (LC-PFN) [1] extends Prior-Data Fitted Networks (PFNs) [12] to the task of learning curve extrapolation. LC-PFN is pre-trained on synthetic data generated from parametric priors over epoch-wise learning curves. Unlike parametric methods that provide point estimates, LC-PFN can provide uncertainty estimates while achieving remarkable prediction efficiency compared with classical Markov Chain Monte Carlo (MCMC) methods. However, LC-PFN relies on pre-defined parametric priors over learning curves, which require substantial manual design effort and may need to be re-specified when extrapolating to different curve families, for example, NSL-PFN for neural scaling laws [7].

To reduce the effort of manually specifying priors and illustrate how to systematically design new priors, Viering et al. [14] proposed a framework that trains LC-PFN directly on real data, called Real Data LC-PFN. The results are evaluated on the sample-wise learning curve database LCDB 1.0 [11], and it outperformed either the original LC-PFN or LC-PFN trained with priors derived from a single parametric model.

Therefore, [14] is the most closely related work to ours. Nonetheless, LCDB 1.0, as discussed above, is limited by substantial missing values within the curves and a restricted number of anchors. Furthermore, prior work did not include a direct comparison of Real Data LC-PFN with traditional parametric fitting methods.

3 Methodology

This section outlines our learning curve extrapolation approaches and evaluation setup.

Real Data LC-PFN. For Real Data LC-PFN [14], the architecture strictly mirrors that of the original LC-PFN [1], but it is trained directly on real data. A 12-layer LC-PFN model is used with 512 embedding dimension. During training, the model is optimized with the negative log-likelihood loss on a discretized probability distribution, which enables the network to output full predictive distributions over learning curve values rather than point estimates. To support fine-grained uncertainty modeling 1000 bins are used. The model is trained over 1000 epochs using a batch size of 100 and a learning rate of 1×10^{-4} following the setup in [1]. The data augmentation strategy proposed in [14] is adopted. For evaluation, the median value of the predicted distribution is reported as the final extrapolated curve.

Parametric Models. The parametric model is fitted to the observed portion of the curve via Levenberg-Marquadt [3]. Based on the comprehensive empirical evaluation by Mohr et al. [11], we choose three most competitive candidates: MMF4 ($f(n) = (ab + cn^d)/(b + n^d)$), WBL4 ($f(n) = c - b \cdot \exp(-a \cdot n^d)$), and POW4 ($f(n) = a - b \cdot (d + n)^{-c}$), where $f(\cdot)$ is a function of the training set size n. Additionally, a, b, c, d are tunable parameters.

The initial parameter values and bounds are fixed a priori (see Appendix A), informed by empirical stability. Fitting is performed using analytical Jacobians to improve convergence speed, prioritizing bounded optimization. In cases where this fails (e.g., due to local minima or ill-conditioning), an unbounded optimization is attempted as a fallback. If both strategies fail, the model reverts to using the initial guess to ensure

the procedure remains complete for all curves.

Evaluation Protocol. For each learning curve evaluation, a cutoff point is established to simulate real-world scenarios where only partial learning curves are available. The curves are divided into observed (training) and unobserved (testing) portions, with models fitted on the observed portion and evaluated on their ability to extrapolate to the unobserved region. The cutoff point is determined by finding the anchor size closest to the target percentage of the range between minimum and maximum training sizes in each curve. Figure 1 shows a visual example of extrapolation performed by all four models.

Model performance is evaluated using three complementary metrics: Symmetric Mean Absolute Percentage Error (SMAPE), Mean Squared Error (MSE), and Mean Absolute Error (MAE). SMAPE (expressed as $(100/m)\sum_{i=1}^{m}(|y_i - \hat{y}_i|/|y_i| + |\hat{y}_i|)$, where $\hat{y}_i \in \mathbb{R}$ is the predicted and $y_i \in \mathbb{R}$ is the actual value and m is the number of test instances) is chosen as the primary metric because learning curve extrapolation involves comparing prediction quality across algorithm-dataset combinations with vastly different performance scales. SMAPE's scale-independence enables fair comparison of extrapolation accuracy regardless of performance regime, focusing on relative prediction errors rather than absolute deviations. MSE, (expressed as $(1/m)\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$), and MAE, (expressed as $(i/m)\sum_{i=1}^{m}|y_i - \hat{y}_i|$), are included as widely used regression metrics to facilitate comparison with other studies, with MSE providing sensitivity to outliers and MAE offering robust, easily interpretable average prediction errors.

4 Experimental Setup

This section describes the two complementary experiments to evaluate performance across different generalization scenarios, extrapolation cutoffs, and learning curve shapes. In each experiment, evaluation is conducted 5 times with different sampling seeds.

4.1 Experiment 1: Comparison Across Generalization Scenarios and Cutoffs

This experiment aims to simultaneously investigate performance across diverse generalization scenarios (**RQ1**) and extrapolation cutoff points (**RQ2**). Datasets are randomly split into training (80%) and testing (20%) subsets. The 24 machine learning algorithms are similarly partitioned into training (80%) and testing (20%) subsets. Each learning curve belongs to one of four evaluation scenarios based on whether its dataset and learner are assigned to training or testing: KDKL (both training), UD (dataset testing, learner training), UL (dataset training, learner testing), and UDUL (both testing). This dual partitioning enables evaluation of different aspects of generalization. Five cutoff percentages $(10\%, 30\%, 50\%, 70\%, \text{and } 90\%)$ are applied using 1000 randomly sampled learning curves per scenario and cutoff combination.

4.2 Experiment 2: Comparison Across Learning Curve Shapes

The experiment aims to identify the influence of the shape of learning curves on extrapolation performance (**RQ3**). Following [15], we group the learning curves in the test

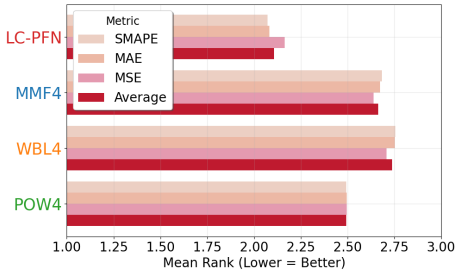


Fig. 2: Mean model rankings across three evaluation metrics (SMAPE, MAE, MSE).

set into the four shapes. The learning curves are randomly split into training (80%) and testing (20%) subsets. Performance comparisons are conducted across five cutoff percentages ($10\%, 30\%, 50\%, 70\%,$ and 90%). Sampling equally across the fixed cutoffs, as in Experiment 1, is employed to reduce the variability of results. For each shape and cut-off combination, 1000 randomly sampled learning curves are evaluated.

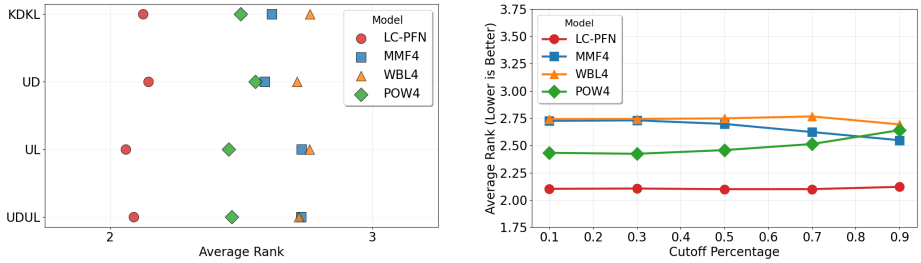
5 Results

This section presents the findings from two experiments defined in Section 4.

5.1 Results 1: Comparison Across Generalization Scenarios and Cutoffs

The comprehensive evaluation of the four models (Real Data LC-PFN, MMF4, WBL4, and POW4) across different generalization scenarios and cutoff percentages reveals several key findings regarding their relative performance and extrapolation capabilities. Figure 2 shows the relative mean rankings of the models across the three evaluation metrics. All three evaluation metrics produced similar rankings across the four models, strengthening the reliability of the chosen metrics in the context of this experimental setup (See Appendix B for the absolute performances). Considering the average ranking across all metrics, Real Data LC-PFN achieved the best overall performance, with reasonable margins ahead of the other models, followed by POW4, MMF4, and WBL4.

Performance Across Transfer Scenarios. Analysis of model performance across generalization scenarios reveals distinct patterns for each model, as shown in Figure 3a. Real Data LC-PFN consistently demonstrated the strongest performance among all four scenarios (KDKL, UD, UDUL, and UL), maintaining an average ranking of approximately $2.0 - 2.3$ across all. This consistent performance demonstrates Real Data LC-PFN's robust generalization capability across different types of learning curve data and learner configurations. POW4 ranked second across all transfer scenarios, showing stable but inferior performance compared to LC-PFN. The remaining parametric models (MMF4 and WBL4) exhibited some variability across transfer scenarios, with both models performing similarly and ranking third and fourth, respectively.



(a) Average model rankings across the four scenarios.

(b) Ranking across different learning curve cutoffs.

Fig. 3: Model performance analysis across transfer scenarios and cutoff percentages

Performance Across Cutoff Percentages. The analysis of performance across different cutoff percentages shown in Figure 3b reveals that Real Data LC-PFN maintains the best performance consistently across all cutoff points, demonstrating remarkable stability with rankings around $2.0 - 2.1$ regardless of how much learning curve data was available for fitting. POW4 showed the second-best performance across most cutoff percentages, while both MMF4 and WBL4 showed some variation in performance as more of the learning curve became available for fitting, with both models generally improving slightly as more data became available (see Figures 8, 9, 10 in Appendix B).

Statistical Significance Analysis. We applied independent samples t-tests for pairwise model comparisons within each shape-cutoff combination, testing the null hypothesis that mean MAE/SMAPE values are equal between models. The analysis reveals that Real Data LC-PFN differs significantly from all parametric models, with very low p-values (ranging from $3.6e-7$ to $9.4e-05$). These low p-values demonstrate that Real Data LC-PFN's superior performance is statistically significant and not due to random variation, though the large sample size ($n = 1000$) used in the analysis contributes to the statistical power. In contrast, comparisons between the parametric models (MMF4, POW4, WBL4) yielded higher p-values (ranging from 0.27 to 0.97), indicating that the performance differences between these models are not statistically significant. For boxplots capturing performance distributions and stability across seeds, as well as more analysis on ranking, see Appendix B.

5.2 Results 2: Comparison Across Learning Curve Shapes

This shape-specific evaluation of model performance provides some insights into the specialized strengths and weaknesses of both parametric and neural network approaches. Figure 4 shows win rates of Real Data LC-PFN against three baseline models (MMF4, POW4, and WBL4) across different learning curve shapes and cutoff percentages. Here, the win rates are computed based on MSE (additional metrics using MAE and SMAPE see Appendix C). In general, the win-rates are pretty close to 50%, indicating many ties. Real Data LC-PFN emerges as the strongest performer on monotone convex curves by a small margin, obtaining a maximum win-rate of 60%. It aligns with expectations since

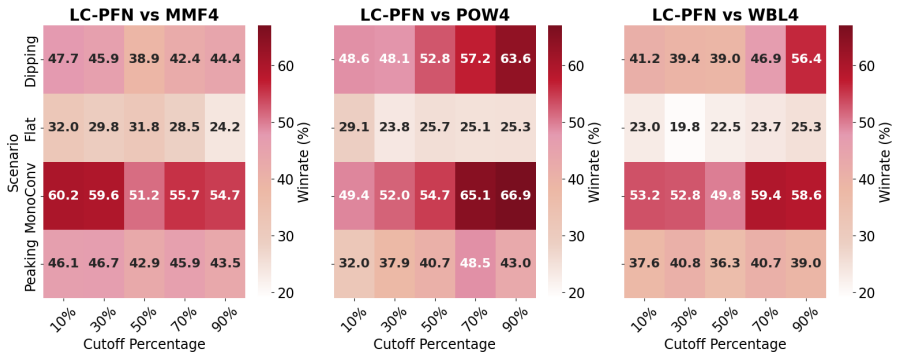


Fig. 4: MSE win rates of Real Data LC-PFN against three baseline models (MMF4, POW4, and WBL4) across different learning curve shapes and cutoff percentages.

the training set was created by randomly sampling from LCDB 1.1, which contains a significant proportion of monotone convex curves (around 78%) [15]. Consequently, performance on monotone convex curves is strong. In contrast, the performance of Real Data LC-PFN on dipping and flat curves is generally less competitive than that of parametric models. For dipping curves, Real Data LC-PFN exhibits relatively stronger performance at higher cutoff levels, where it surpasses both POW4 and WBL4. However, its win rates remain consistently lower than those of MMF4. Still, the win rate is fairly close, at around 40%.

To further validate these observations, we provide an additional analysis in Figure 5, which presents the MSE differences between Real Data LC-PFN and parametric models with different curve types and cutoff percentages (for additional metrics using MAE and SMAPE see Appendix C). We can see that Real Data LC-PFN maintains highly competitive performance against parametric models across nearly all curve types, with the only exception being the dipping scenario, where it underperforms relative to MMF4, particularly in low cutoffs. The results align with our observations discussed above. More analysis on ranking is presented in Appendix C.

6 Discussion

Addressing **RQ1**, Real Data LC-PFN demonstrates superior performance across all generalization scenarios, excelling not only in familiar scenarios (KDKL) but also in generalization scenarios (UD, UL, UDUL). Its ability to maintain performance when encountering completely unseen datasets and learners simultaneously (UDUL) is particularly noteworthy, indicating robust feature extraction capabilities that generalize beyond the training distribution. Among parametric competitors, POW4 emerges as the strongest contender, likely due to its power-law formulation's as the typical shapes of learning curves [13].

A crucial finding for **RQ2** reveals that Real Data LC-PFN maintains the best performance consistently across multiple cutoff percentages, demonstrating remarkable

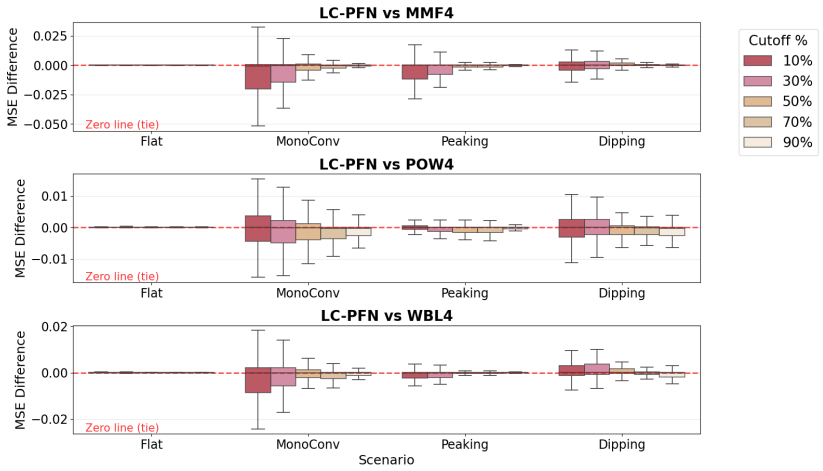


Fig. 5: The MSE differences between Real Data LC-PFN and parametric models across different curve types and cutoff percentages. Negative values indicate Real Data LC-PFN has lower MSE (better performance), while positive values indicate higher MSE (worse performance) as compared to the parametric models.

stability with rankings around $2.0 - 2.1$ regardless of observed data before cutoff. In terms of absolute performance, all models struggle with low cutoffs, but LC-PFN's marginally better resilience provides meaningful practical value. This shows that LC-PFN is remarkably robust, even when observing very few points, precisely the scenario most valuable to practitioners seeking early performance predictions without extensive computational investment.

Investigating **RQ3** reveals that Real Data LC-PFN's performance is highly contextual rather than uniformly superior. For monotone convex curves, Real Data LC-PFN achieves meaningful advantages, particularly against MMF4 and POW4, likely because these patterns dominate in the training dataset. This is noteworthy since POW4 was specifically designed to model such diminishing-returns scenarios, yet Real Data LC-PFN still outperforms it. The $50 - 68\%$ win rates represent minor improvements over random chance. For flat curves, while Real Data LC-PFN's low win rates appear concerning, the practical impact is negligible with MAE differences of $0.01 - 0.02$. Such minimal prediction errors rarely affect real-world decisions. This suggests that ranking-based evaluation may overstate the practical significance of small performance gaps when absolute errors are universally low. For peaking curves, the observation is similar to that in flat curves, but less pronounced. MSE differences remain small, indicating Real Data LC-PFN still performs decently despite falling behind in win rates. Dipping curves, however, reveal where Real Data LC-PFN truly struggles. Win rates hovering around $40 - 50\%$ indicate performance barely distinguishable from parametric models. At the same time, the high variance in MAE differences and the struggling absolute performances across models suggest that neither Real Data LC-PFN nor parametric

models can reliably capture these complex dynamics. This challenges the notion that neural networks automatically excel at irregular complex pattern recognition.

One limitation of our comparison is that the performance of parametric models could be further improved by employing multiple repeats, as was done in LCDB 1.0 [11]. Likewise, Real Data LC-PFN itself may benefit from more extensive hyperparameter tuning or larger scale of parameters, which could potentially narrow some of the observed performance gaps.

Overall, Real Data LC-PFN shows general advantages over parametric models. Although it performs well in monotone convex curves (and acceptably on flat curves) and offers uncertainty quantification capabilities, its benefits diminish significantly for ill-behaved learning patterns (particularly dipping), suggesting that the neural approach is constrained by its training data distribution. Practitioners should weigh these performance trade-offs alongside computational considerations: Real Data LC-PFN provides near-instantaneous inference after training, while parametric models require no pre-training but involve slower inference.

7 Conclusion

Our study demonstrates that Real Data LC-PFN shows promising but mixed performance compared to established parametric models (POW4, MMF4, WBL4) for learning curve extrapolation across LCDB 1.1. While the neural network approach maintains consistent performance regardless of the amount of observed learning curve data available, from early-stage extrapolation (10% cutoff) to near-complete curves (90% cutoff), its advantages are more nuanced than initially expected. The key findings across our three research questions reveal important caveats. Real Data LC-PFN maintains advantages across all generalization scenarios (**RQ1**), demonstrates consistent performance across all cutoff percentages (**RQ2**), but shows limited superiority across learning curve shapes, particularly struggling with ill-behaved patterns where parametric models perform only slightly better (**RQ3**). These results indicate that while neural network approaches show promise for learning curve extrapolation, they do not provide the universal superiority over traditional parametric methods as anticipated when learning irregular shape patterns from data.

Future Work. Our findings suggest that model selection for learning curve extrapolation should be context-dependent, considering both curve characteristics and computational constraints, rather than adopting a one-size-fits-all neural network approach. Consequently, future research could focus on developing hybrid approaches that route different curve types through specialized models, rather than seeking universal solutions. Additionally, investigating training strategies that better handle data imbalance and ill-behaved learning curves could address Real Data LC-PFN's current limitations.

References

1. Adriaensen, S., Rakotoarison, H., Müller, S., Hutter, F.: Efficient bayesian learning curve extrapolation using prior-data fitted networks. In: NeurIPS (2023)
2. Cohn, D., Tesauro, G.: Can neural networks do better than the vapnik-chervonenkis bounds? Advances in Neural Information Processing Systems **3** (1990)
3. Gu, B., Hu, F., Liu, H.: Modelling Classification Performance for Large Data Sets. In: Wang, X.S., Yu, G., Lu, H. (eds.) Advances in Web-Age Information Management. pp. 317–328. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
4. Kolachina, P., Cancedda, N., Dymetman, M., Venkatapathy, S.: Prediction of learning curves in machine translation. In: Li, H., Lin, C.Y., Osborne, M., Lee, G.G., Park, J.C. (eds.) Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 22–30. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012), https://aclanthology.org/P12-1003/
5. Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T.: Quantifying the carbon emissions of machine learning (2019), https://arxiv.org/abs/1910.09700
6. Last, M.: Predicting and optimizing classifier utility with the power law. In: Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007). pp. 219–224. IEEE (2007)
7. Lee, D., Lee, D.B., Adriaensen, S., Lee, J., Hwang, S.J., Hutter, F., Kim, S.J., Lee, H.B.: Bayesian neural scaling laws extrapolation with prior-fitted networks. arXiv preprint arXiv:2505.23032 (2025)
8. Lin, X., Zhou, X., Liu, C., Zhou, X.: Efficiently computing weighted proximity relationships in spatial databases. In: International Conference on Web-Age Information Management. pp. 279–290. Springer (2001)
9. Loog, M., Viering, T.: A survey of learning curves with bad behavior: or how more data need not lead to better performance (2022), https://arxiv.org/abs/2211.14061
10. Mohr, F., van Rijn, J.N.: Fast and informative model selection using learning curve cross-validation. IEEE TPAMI (2023)
11. Mohr, F., Viering, T.J., Loog, M., van Rijn, J.N.: Lcdb 1.0: An extensive learning curves database for classification tasks. In: ECML PKDD. pp. 3–19. Springer (2022)
12. Müller, S., Hollmann, N., Arango, S.P., Grabocka, J., Hutter, F.: Transformers can do bayesian inference. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=KSugKcbNf9
13. Viering, T., Loog, M.: The shape of learning curves: a review. IEEE TPAMI **45**(6), 7799–7819 (2022)
14. Viering, T.J., Adriaensen, S., Rakotoarison, H., Hutter, F.: From epoch to sample size: Developing new data-driven priors for learning curve prior-fitted networks. In: AutoML Conference 2024 (Workshop Track) (2024), https://openreview.net/forum?id=neEKHQDTHV
15. Yan, C., Mohr, F., Viering, T.: Lcdb 1.1: A database illustrating learning curves are more ill-behaved than previously thought (2025), https://arxiv.org/abs/2505.15657

A Parametric Models - Methodology Used

For MMF4, the initial guess $[a, b, c, d] = [0.9, 1000.0, 0.1, 1.0]$ encodes a strong asymptotic behavior with gradual curvature, while the bounds $[0.01, 1e\text{-}6, 0.0, 0.01]$, $[1.0, \infty, 1.0, 10.0]$ are selected to reflect plausible limits for accuracy and to prevent numerical instability during optimization.

In the case of WBL4, the initial parameters are set to $[a, b, c, d] = [0.001, 0.8, 0.9, 1.0]$, reflecting slow exponential decay toward high performance. The fitting bounds ensure smooth convergence and prevent divergence, especially in the presence of steep curves or small-scale datasets.

POW4 is initialised with $[a, b, c, d] = [0.9, 0.8, 1.0, 100.0]$, with bounds $[0.01, 0.01, 0.001, 1.0]$, $[1.0, 2.0, 5.0, 10000.0]$ to ensure the model remains flexible yet stable during fitting.

B Experiment 1: Additional Resulting Plots

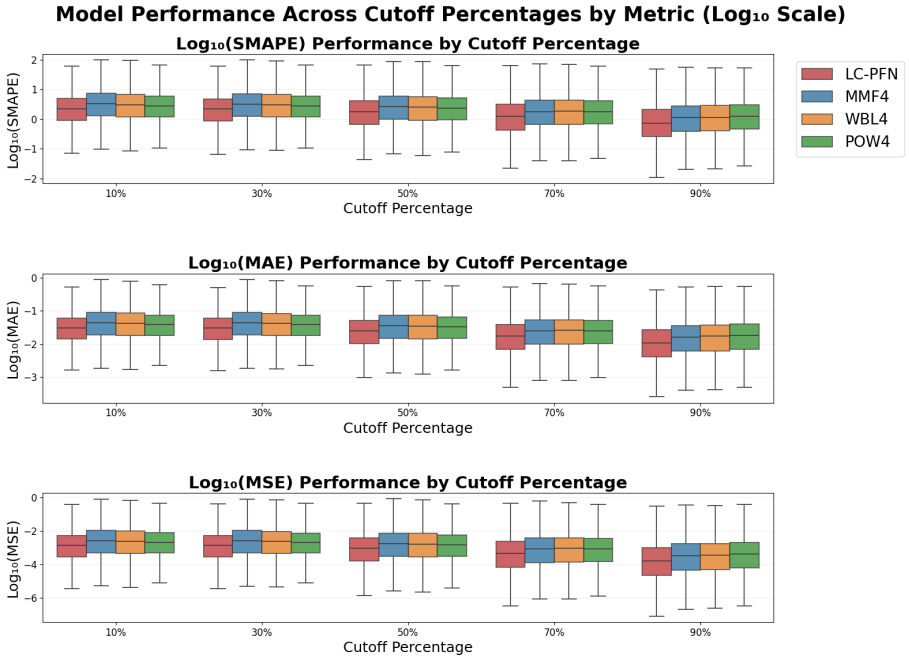


Fig. 6: Model performance across different cutoff percentages (10%–90%) for SMAPE, MAE, and MSE. Lower is better.

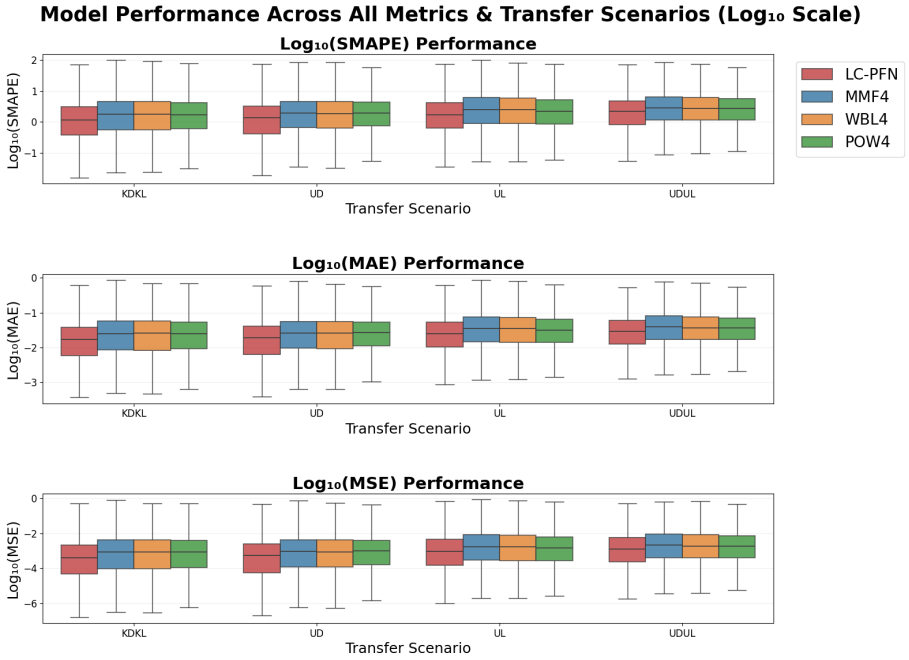


Fig. 7: Model performance across four transfer scenarios: Known Dataset Known Learner (KDKL), Unseen Dataset (UD), Unseen Learner (UL), and Unseen Dataset Unseen Learner (UDUL). Lower is better.

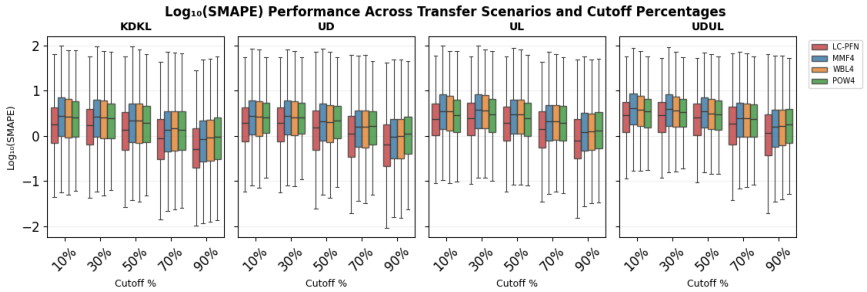


Fig. 8: Performance indicated by SMAPE across all transfer scenarios and cutoffs. 4 transfer scenarios x 5 cutoff points = 20 experimental conditions. Lower is better.

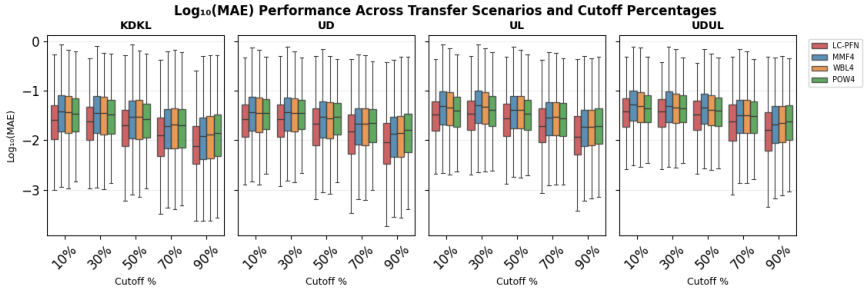


Fig. 9: Performance indicated by MAE across all transfer scenarios and cutoffs. 4 transfer scenarios x 5 cutoff points = 20 experimental conditions. Lower is better.

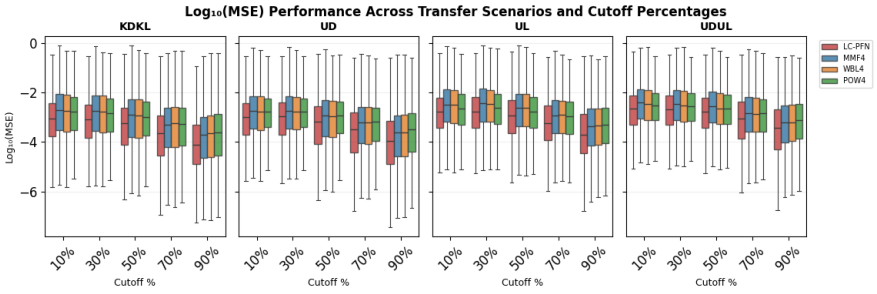


Fig. 10: Performance indicated by MSE across all transfer scenarios and cutoffs. 4 transfer scenarios x 5 cutoff points = 20 experimental conditions. Lower is better.

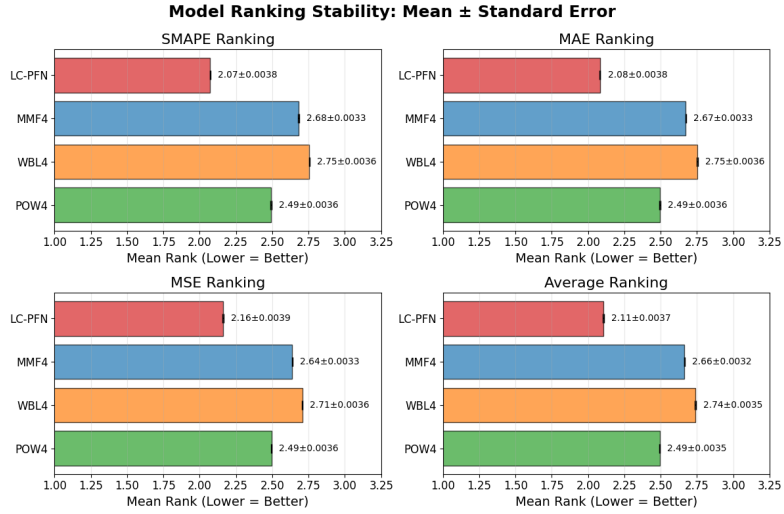


Fig. 11: Model ranking stability across all experimental conditions. Bar charts show mean rankings with standard error bars for each metric (SMAPE, MAE, MSE) and overall average ranking. Rankings are computed within each experimental condition (seed, scenario, cutoff percentage, learning curve) where 1 = best performance, 4 = worst performance. LC-PFN consistently achieves the best rankings across all metrics with exceptional stability (SE $\approx$ 0.004). Error bars represent standard errors across 5 sampling seeds.

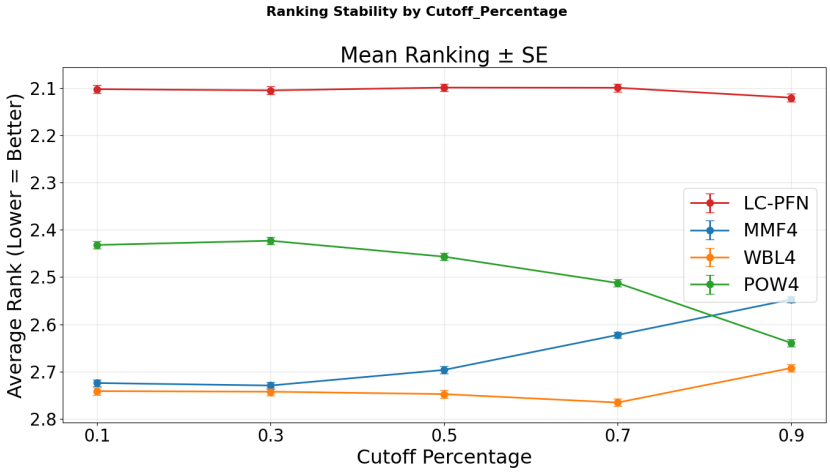


Fig. 12: Model ranking stability across different cutoff percentages. Plot shows mean rankings $\pm$ standard error for each model as a function of training data availability (10%-90% of learning curve used for training). All models demonstrate exceptional ranking stability with standard errors below 0.0084.

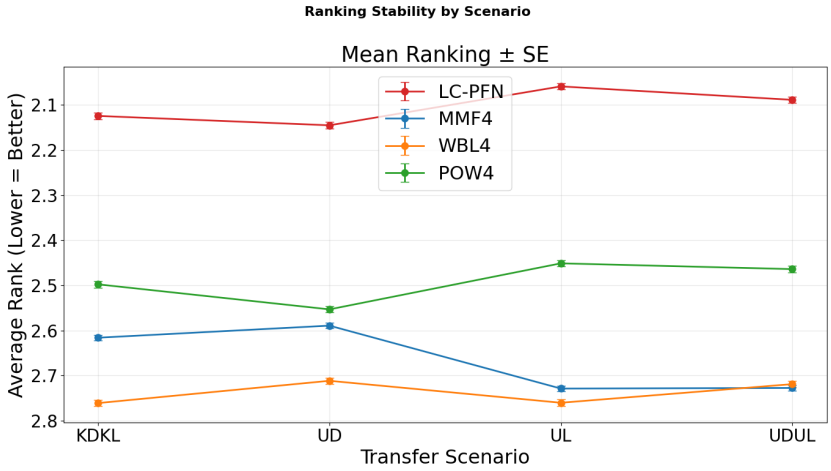
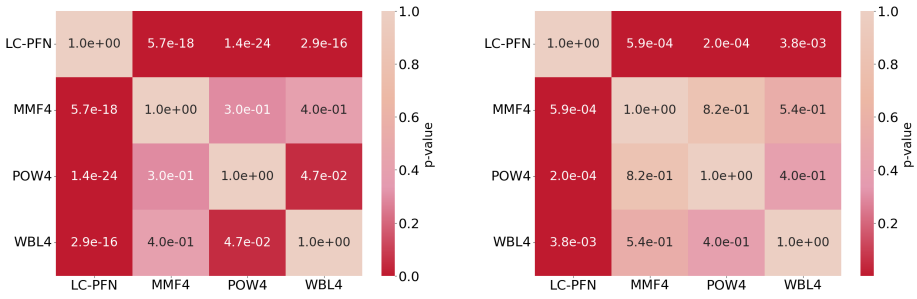


Fig. 13: Model ranking stability across transfer learning scenarios. Plot presents mean rankings ± standard error for each model across four transfer scenarios: Known Data-Known Learners (KDKL), Unseen Data-Known Learners (UD), Known Data-Unseen Learners (UL), and Unseen Data-Unseen Learners (UDUL). The minimal standard errors (SE < 0.008) across all models indicate high confidence in model performance rankings with different seeds for sampling.



(a) Statistical significance test using MAE

(b) Statistical significance test using SMAPE

Fig. 14: Statistical significance heatmaps showing p-values for pairwise model comparisons. LC-PFN shows statistically significant differences from all parametric models with very low p-values, while comparisons between parametric models show higher p-values indicating less significant differences.

C Experiment 2: Additional Resulting Plots

For flat curves, although LC-PFN appears to trail behind other models in terms of win rate, the boxplot analysis in Figures 21, 22, 23, reveals that prediction errors are extremely small across all models for this shape category, making the ranking differences

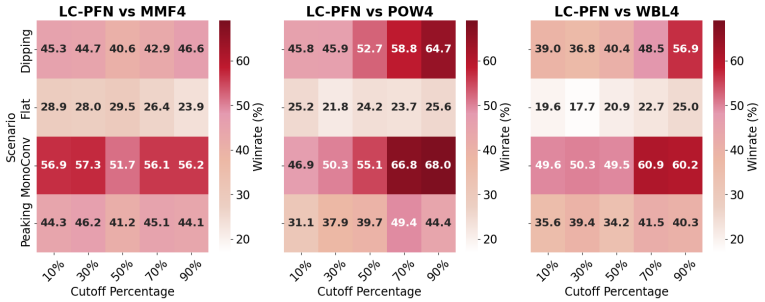


Fig. 15: MAE win rates of Real Data LC-PFN against three baseline models (MMF4, POW4, and WBL4) across different learning curve shapes and cutoff percentages.

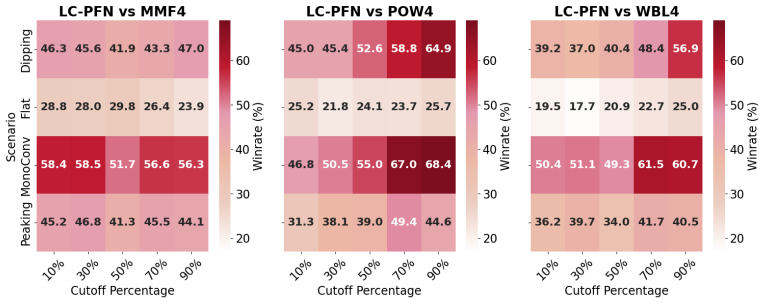
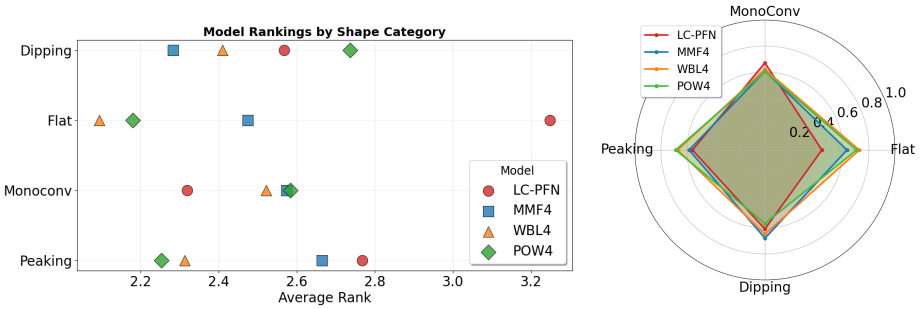


Fig. 16: SMAPE win rates of Real Data LC-PFN against three baseline models (MMF4, POW4, and WBL4) across different learning curve shapes and cutoff percentages.



(a) Average model rankings across four learning curve shape categories. LC-PFN demonstrates a slight advantage for monotone convex curves, while its ranks lag behind in the extrapolation of flat curves.

(b) Radar chart showing normalized performance scores (0-1 scale) for each model across learning curve shapes. LC-PFN shows strong performance for monotone convex shapes.

Fig. 17: Model performance analysis across curve shapes.

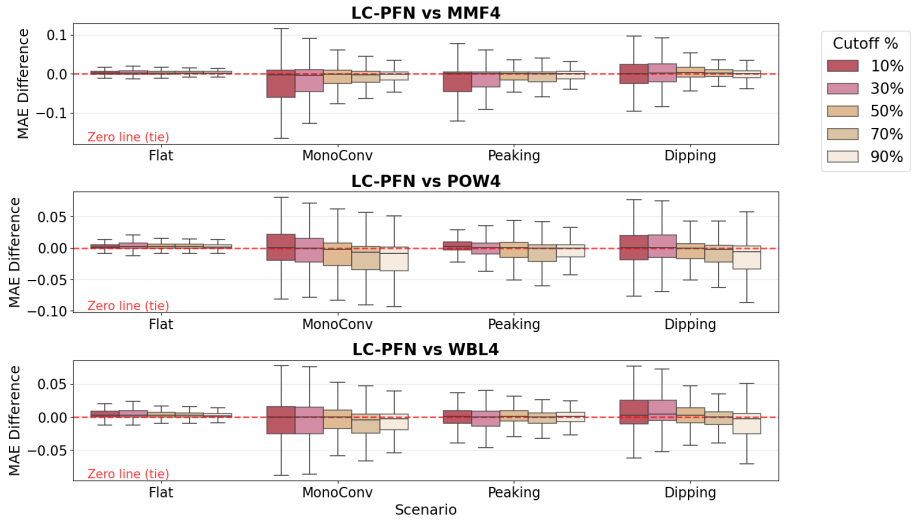


Fig. 18: The MAE differences between Real Data LC-PFN and parametric model with different curve types and cutoff percentages. Negative values indicate Real Data LC-PFN achieves lower prediction errors (better performance), while positive values indicate higher errors (worse performance).

negligible in practical terms and confirming that all models handle flat curves effectively despite LC-PFN's relative disadvantage. The more challenging peaking and dipping curve shapes prove difficult for all models to predict accurately, and the LC-PFN is not able to predict ill-behavior any better than the parametric models. Detailed performance distributions are captured through standard error (see Figure 24) and boxplot visualizations (see Figures 25, 26).

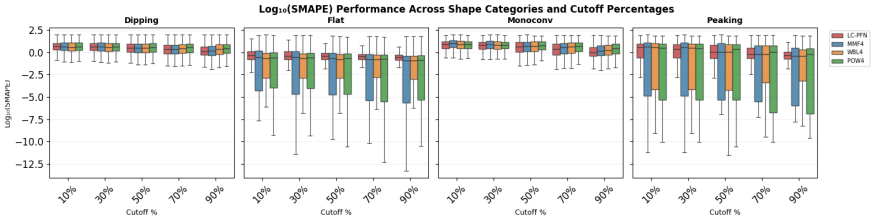


Fig. 21: Performance indicated by SMAPE across all shapes and cutoffs. 4 shapes x 5 cutoff points = 20 experimental conditions. Lower is better.

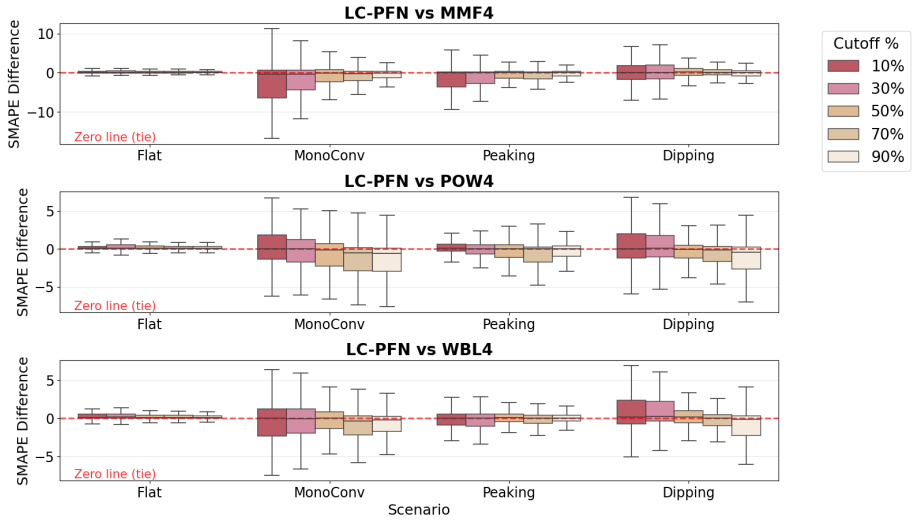


Fig. 19: The SMAPE differences between Real Data LC-PFN and parametric model with different curve types and cutoff percentages. Negative values indicate Real Data LC-PFN achieves lower prediction errors (better performance), while positive values indicate higher errors (worse performance).

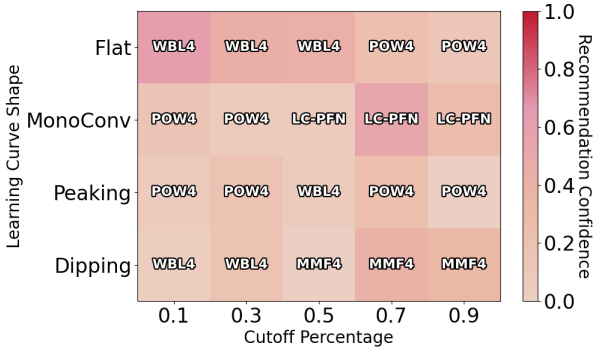


Fig. 20: Model recommendation matrix based on lowest average rank performance, with color intensity indicating recommendation confidence. The recommendations are determined by identifying the model with the lowest average rank for each shape-cutoff combination, while the color intensity represents recommendation confidence calculated from the performance gap between the best and second-best models. A rank difference of 0.5 or greater between the top two models results in maximum confidence (darkest color), indicating a clear performance advantage.

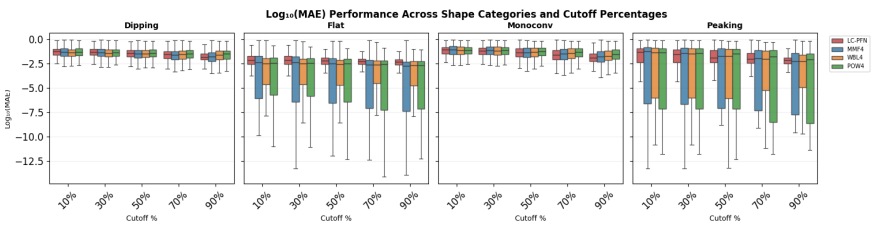


Fig. 22: Performance indicated by MAE across all shapes and cutoffs. 4 shapes x 5 cutoff points = 20 experimental conditions. Lower is better.

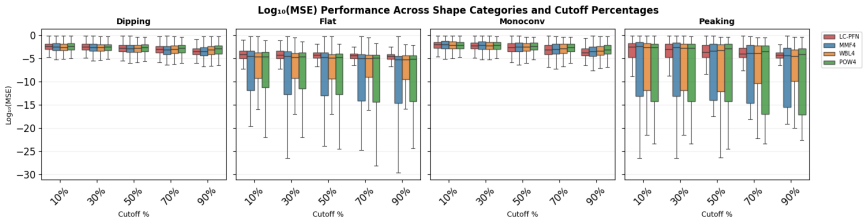


Fig. 23: Performance indicated by MSE across all shapes and cutoffs. 4 shapes x 5 cutoff points = 20 experimental conditions. Lower is better.

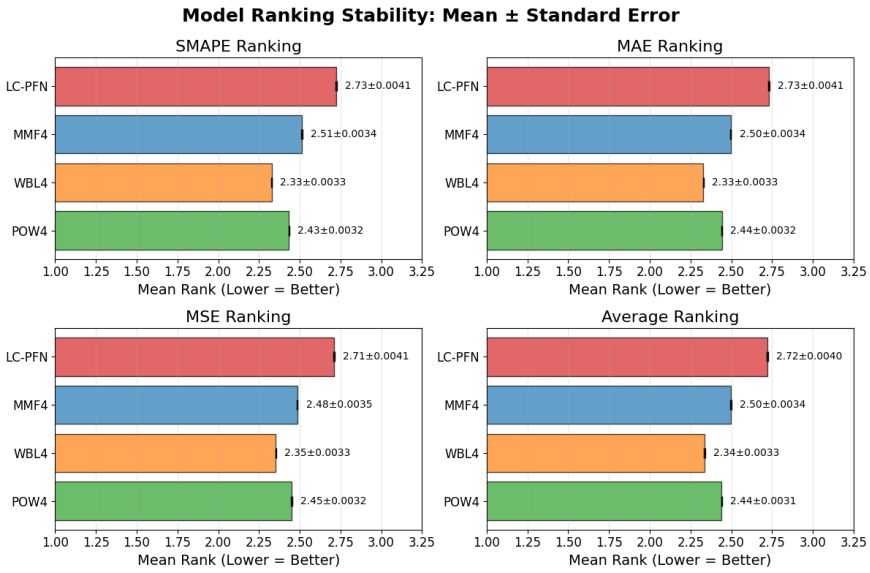


Fig. 24: Model ranking stability across all experimental conditions. Bar charts show mean rankings with standard error bars for each metric (SMAPE, MAE, MSE) and overall average ranking. Rankings are computed within each experimental condition (seed, shape, cutoff percentage, learning curve) where 1 = best performance, 4 = worst performance. Error bars represent standard errors across 5 sampling seeds.

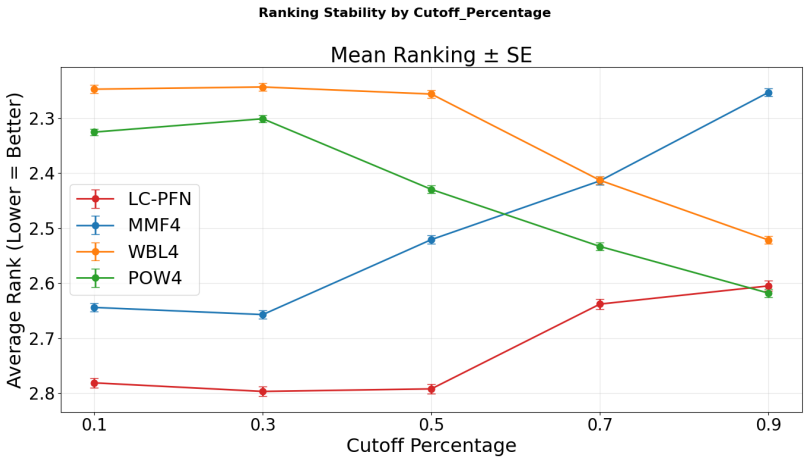


Fig. 25: Model ranking stability across different cutoff percentages. Plot shows mean rankings ± standard error for each model as a function of training data availability (10%-90% of learning curve used for training). All models demonstrate exceptional ranking stability.

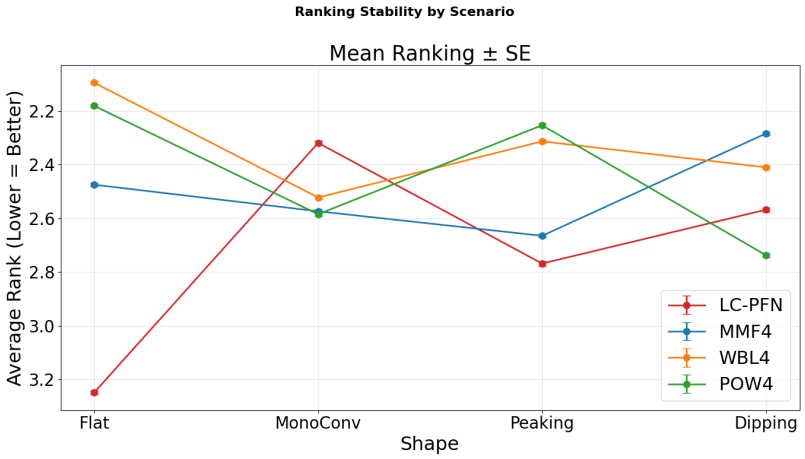
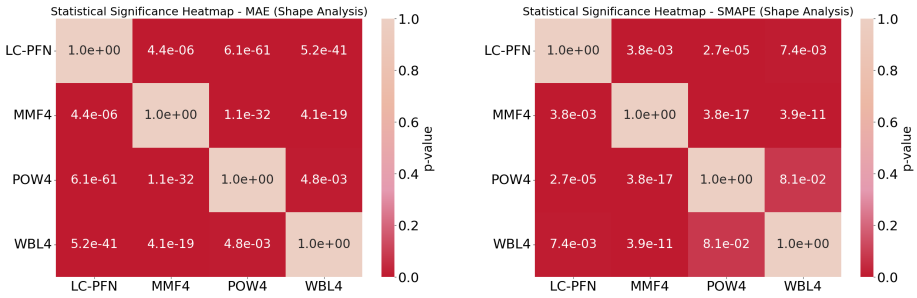


Fig. 26: Model ranking stability across transfer learning scenarios. Plot presents mean rankings ± standard error for each model across four shapes. The minimal standard errors across all models indicate high confidence in model performance rankings with different seeds for sampling.



(a) Statistical significance test using MAE

(b) Statistical significance test using SMAPE

Fig. 27: Statistical significance heatmaps showing p-values for pairwise model comparisons.

D Data and Computational Resources Used

This research uses the publicly available Learning Curve Database (LCDB) 1.1 and OpenML datasets, which are all freely accessible for research purposes. No personally identifiable or sensitive information was involved in this study, as all learning curves represent aggregated performance metrics from ML algorithms. We acknowledge the original contributors to LCDB and OpenML, and encourage users of our code to maintain proper attribution to these foundational resources.

Given the growing concern about the environmental impact of ML research, we report the computational resources used in this study. The LCPFN training required approximately 16 GPU-hours on NVIDIA A40 GPUs (8 hours for each of the two models), consuming an estimated 4.8 kWh of electricity. The evaluation phase utilized 200 CPU-hours across AMD EPYC processors, consuming approximately 1.4 kWh. All computations were performed on TU Delft's DAIC HPC cluster. The estimated total carbon footprint is 3.1 kg CO_2 equivalent, calculated using the methodology from Strubell et al. [5] with global average electricity carbon intensity of 0.5 kg CO_2/kWh.