Extrapolating Learning Curves: When Do Neural Networks Outperform Parametric Models?

Objectives

To analyze the performance of neural network and parametric approaches for sample-size learning curve extrapolation, and determine the conditions under which either approach might offer a consistent advantage.

Introduction

Why This Matters:
How much data do you *really* need to reach your performance goal? It's a key question in ML, and hard to answer, due to the unpredictable shapes of learning curves [4].

Why Sample-Size Curves?
Unlike epoch-based curves, sample-size learning curves guide *early* decisions, like whether to collect more data or which model to try. They're even useful for non-iterative models like k-NN [4, 2].

What's Been Tried:

- **Parametric models** (e.g., MMF4, WBL4) fit many tasks well [3].
- **Neural networks** (e.g., LC-PFNs) can extrapolate from limited samples [1].

Research Gap:
The comparative strengths of these approaches under different conditions remain underexplored.

Research Questions

When do neural networks outperform parametric models in learning curve extrapolation?
We investigate:

1. **Transfer scenario:** Performance on unseen datasets vs. unseen learners vs. both unseen?
2. **Observable window:** How does the amount of observed data impact extrapolation quality?
3. **Curve characteristics:** Do certain learning curve shapes favor neural or parametric approaches?

Methodology

We evaluate **Learning Curve Prior-Data Fitted Networks (LC-PFNs)** against:

- **MMF4:** $(a \cdot b + c \cdot n^d) \,/\, (b + n^d)$
- **WBL4:** $c - b \cdot e^{-a \cdot n^d}$
- **POW4:** $a - b \cdot (d + n)^{-c}$

The experimental framework uses the LCDB1.1 (learning curves from 265 OpenML datasets and 24 learners). Performance evaluation employs: (1) **Symmetric Mean Absolute Percentage Error (SMAPE)** for relative error measurement, (2) **Mean Absolute Error (MAE)**, and (3) **Mean Squared Error (MSE)**.

Results

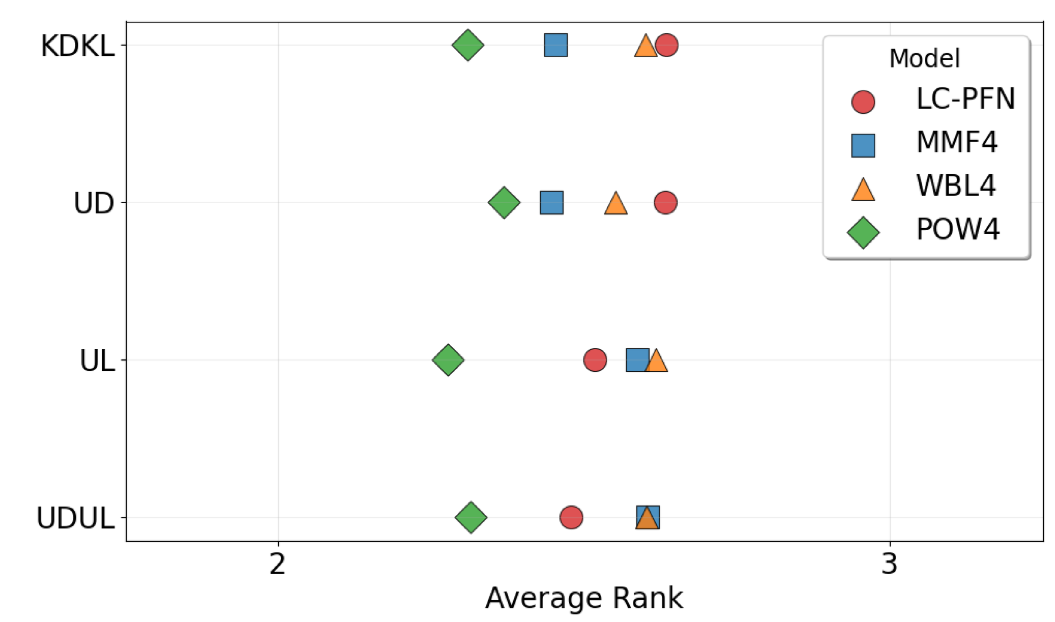

Figure 1: **Average model rankings across four generalization scenarios (lower = better).** POW4 consistently outperforms all other models across all transfer scenario. Surprisingly, LC-PFN ranks worst in familiar scenarios (KDKL, UD) where it should theoretically excel, but shows improved generalization to unseen data (UL, UDUL), suggesting reasonable adaptability despite overall weaker performance.

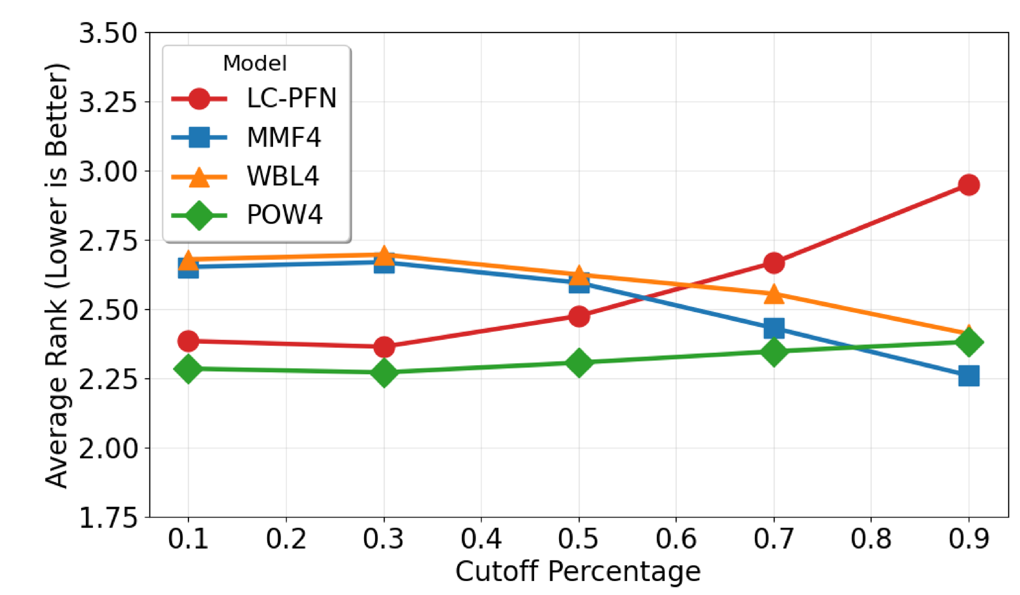

Figure 2: **Model ranking by learning curve cutoff percentage (lower = better).** LC-PFN demonstrates its strongest competitive advantage at early cutoffs (10-50%), ranking second-best when extrapolating from limited data. However, performance degrades as more data becomes available. POW4 maintains consistently superior performance across all cutoffs, while MMF4 and WBL4 improve with more observed data.

Critical Findings

Surprising Result: Parametric models consistently outperform LC-PFN across most scenarios!

- LC-PFN ranks 2nd best at early cutoffs (10-50%) but degrades as more data becomes available. This reveals neural networks have great potential when extrapolating from limited early-stage data!
- LC-PFN shows slightly better performance on irregular curves (peaking/dipping) but still trails parametric models on traditional & flat curves.

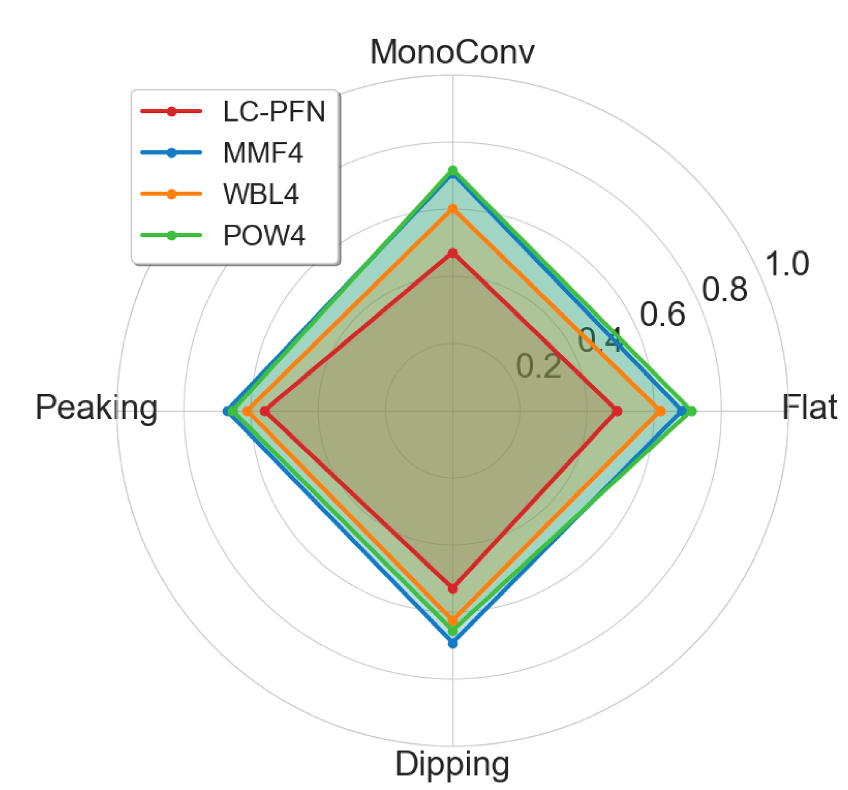

Figure 3: **Normalized performance scores across curve morphologies (higher = better).** The dominance of parametric models may reflect LCDB 1.1's 86% well-behaved curves matching their own monotone assumptions. LC-PFN shows slightly improved performance on irregular curves, suggesting potential beyond dataset bias.

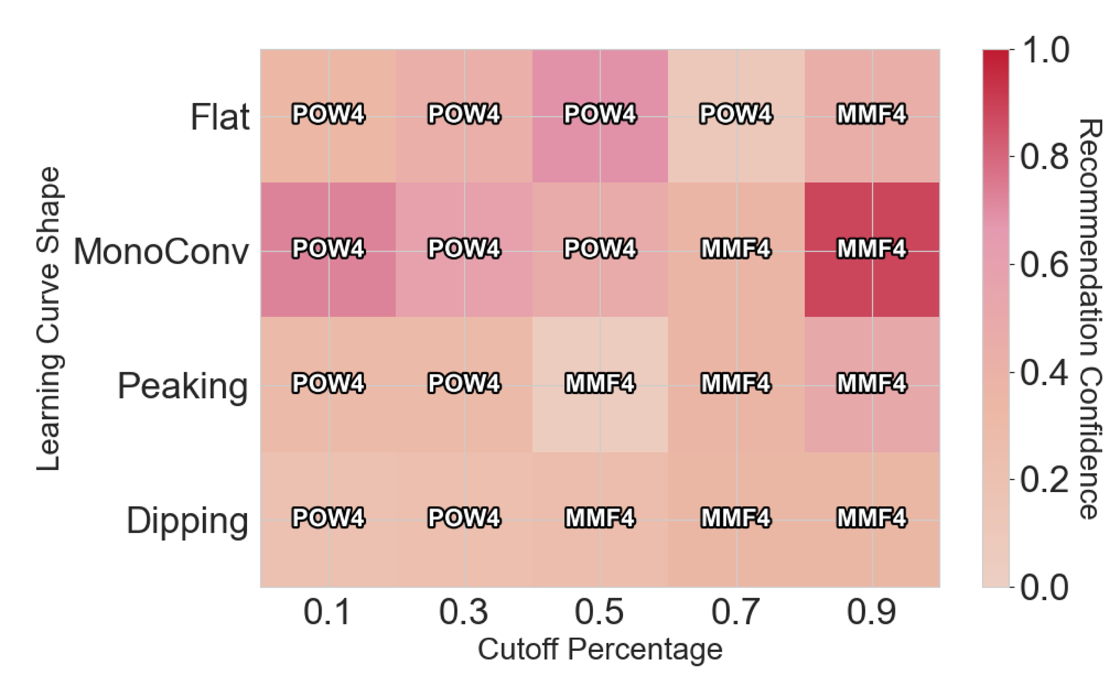

Figure 4: **Recommended model by curve shape and cutoff percentage (color intensity = recommendation confidence).** POW4 dominates early-stage extrapolation (10-30%) across all curve shapes, while MMF4 becomes the preferred choice at higher cutoffs (50-90%), especially for complex Peaking/Dipping behaviors that exhibit non-monotonic patterns. Darker colors indicate higher recommendation confidence.

Conclusions

Main Takeaway: Parametric models (especially POW4) dominate across all scenarios, but LC-PFNs reveal specific niches: (1) strong competitive performance in early-stage extrapolation when data is scarce, (2) relatively better handling of non-traditional curve patterns, and (3) reasonable cross-domain adaptability despite overall weaker performance. Dataset biases toward well-behaved curves may amplify parametric advantages, indicating that practitioners should consider data availability and expected curve behavior when choosing methods.

Impact: This work provides the first comprehensive comparison on realistic, challenging learning curves, offering clear guidance for method selection in practice.

Future Research

Current LC-PFN Limitations:

- Fixed sequence length training - doesn't handle variable-length curves
- This leads to training distribution mismatch with real-world curve diversity
- Training configuration simplicity

Future Directions:

- Develop a variable-length LC-PFN architecture
- Larger, more expressive LC-PFN configuration
- Explore the possibility of hybrid parametric-neural ensemble approaches, to leverage both of their strengths
- Evaluate on dataset with balanced curve morphology distributions to isolate model capabilities from dataset bias

Responsible Research

- All code, random seeds, and pre-trained LC-PFN model weights are publicly available on GitHub to ensure full reproducibility.
- LC-PFN training: approx. 1 GPU-hour, 0.25kWh electricity, 0.125kg CO_2 equivalent.
- LLMs assisted with language refinement and initial code templates only. All research content, analysis, and conclusions are original work.

References & Contact Information

Student: Adelina Cazacu
A.A.Cazacu-1@student.tudelft.nl
Supervisors:
Tom Viering
Cheng Yan
Sayak Mukherjee