How Noisy Is Too Noisy?

Robust Extrapolation of Learning Curves with LC-PFN

Razvan Marian Gherasa
Supervisor(s): Tom Viering, Cheng Yan, Sayak Mukherjee
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Razvan Marian Gherasa
Final project course: CSE3000 Research Project
Thesis committee: Tom Viering, Cheng Yan, Sayak Mukherjee, Matthijs Spaan

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Accurately predicting a machine learning model's final performance based on only partial training data can save substantial computational resources and guide early stopping, model selection, and automated machine learning (AutoML) workflows. Learning Curve Prior-Fitted Networks (LC-PFNs) are a recent data-driven approach to this problem, leveraging transformers trained on prior learning curves to make extrapolations. However, real-world training logs are often noisy and irregular—conditions under which the reliability of LC-PFNs remains largely untested.

This thesis presents a systematic investigation into the robustness of LC-PFNs when exposed to noisy input data. Using LCDB 1.1—a large-scale dataset—we simulate various levels of noise by corrupting the input data with Gaussian perturbations and quantify how prediction accuracy degrades. To improve resilience, we study and propose two complementary mitigation strategies. The first injects artificial noise into the training data itself, either at a constant level or increasing gradually throughout training, encouraging the model to generalize across a spectrum of noise conditions. The second applies post-processing techniques at inference time, such as smoothing the input sequence with an exponential moving average or averaging multiple stochastic predictions using dropout.

Our results show that standard LC-PFNs, trained only on clean data, are highly sensitive to even minor corruptions. In contrast, models trained with gradual noise exposure and evaluated with input smoothing achieve much greater robustness—reducing error by up to 75% under severe noise—while maintaining high accuracy in noise-free settings. This work demonstrates that substantial improvements in learning-curve extrapolation can be achieved without modifying model architecture, using general-purpose techniques suitable for real-world deployment.

1 Introduction

Learning curves—plots of a model's performance as training data grows—help diagnose generalization and guide decisions like early stopping or data collection. In practice, however, the cost of training models to convergence is high, especially when exploring large model spaces or datasets. This motivates the need for *learning curve extrapolation* (LCE): predicting future performance from only partial observations of a model's learning trajectory.

LCE has wide applications in model selection, neural architecture search, and AutoML. Yet, it remains a challenging task—real-world learning curves are often noisy, unstable, and vary greatly across tasks and algorithms. Robustly extrapolating these curves requires models that can generalize across such variation.

Prior work has explored curve fitting methods, Gaussian processes, and meta-learning approaches. Most of these rely on either synthetic curves or strong assumptions about learning dynamics. Recently, Learning Curve Prior-Data Fitted Networks (LC-PFNs) have shown promise in learning task distributions directly, enabling flexible and data-driven extrapolation. However, little is known about how LC-PFNs perform when faced with noise or corruption in real data.

This work addresses the following research questions:

- **RQ1:** How does LC-PFN extrapolation accuracy degrade as input learning curves become noisier?
- **RQ2:** Can we improve LC-PFN robustness by introducing noise during training?

- **RQ3:** What test-time techniques (e.g., smoothing, uncertainty estimation) help mitigate performance drop under noisy inputs?

We answer these questions using real learning curves from the LCDB 1.1 dataset, evaluating LC-PFN variants trained under different noise settings and tested across a range of controlled noise levels. Our findings show that training with gradually increasing noise and applying test-time smoothing (via Exponential Moving Average) leads to significantly more stable extrapolation.

Terminology clarification. The LCDB 1.1 learning curves are referred to as *clean* throughout this thesis, but they inherently include real-world noise from logging, batch effects, or optimization variance. When we refer to *noisy* inputs, we specifically mean additional, synthetic Gaussian perturbations added during evaluation to simulate increased or more severe noise—representing more degraded or chaotic logs than those present in LCDB. This lets us test how far LC-PFNs can be pushed beyond the noise levels seen in the training distribution.

2 Background and Related Work

Learning-curve extrapolation (LCE) aims to predict a model's future performance from partial training data. It enables early stopping, efficient hyperparameter tuning, and cost-aware model selection—key components in modern AutoML. This section outlines the characteristics of real learning curves, recent advances in extrapolation models, and the robustness gap this thesis addresses.

2.1 Empirical Learning Curves and Benchmarks

Real learning curves vary widely in shape and quality. Viering and Loog [6] show that such curves frequently exhibit irregularities—including plateaus, peaking, and double-descent—that deviate from smooth parametric forms and challenge standard extrapolation techniques. LCDB 1.1 [8] makes this diversity explicit: it aggregates learning curves across 265 OpenML tasks and 24 learners, with 14% violating monotonicity assumptions. This variability motivates data-driven, noise-tolerant extrapolators. Our experiments use LCDB 1.1 and additionally inject controlled Gaussian noise to simulate realistic logger artifacts.

2.2 From Curve-Fits to Prior-Fitted Networks

Traditional LCE approaches rely on hand-crafted curve fits (e.g., power-laws, Weibull) or Gaussian processes. These methods fail under irregular or noisy input. Recent work by Adriaensen *et al.* [1] introduced *Learning-Curve PFNs* (LC-PFNs): transformer models trained to perform approximate Bayesian inference on unseen learning curves. Follow-up work [7] improved extrapolation by learning priors directly from LCDB. However, these models were evaluated on mostly clean inputs; their robustness to noise remains untested.

2.3 Robustness Techniques: Training and Inference

Outside LCE, robustness is commonly improved via (i) noise injection during training [2] and (ii) test-time smoothing, such as exponential moving averages (EMA) [4].

2.4 Gap Addressed in This Thesis

To our knowledge, no prior work has:

(i) systematically evaluated LC-PFN accuracy under controlled input noise,

(ii) applied ramp-style noise injection during training to improve robustness, or

(iii) compared this to lightweight test-time defences like EMA.

We address this gap using LCDB 1.1 and show that a simple combination of *noise-ramp training* and *EMA smoothing* yields robust and efficient extrapolation—even under substantial noise—without harming clean-curve performance.

3 Methodology

This section outlines the dataset, model architecture, evaluation pipeline, and robustness strategies used to assess the reliability of Learning Curve Prior-Fitted Networks (LC-PFNs) under noisy inputs.

Implementation Setup. The LC-PFN model architecture and training framework were provided at the outset of this project, including data loading, augmentation, and model training code. My contributions begin with implementing a controlled 80/20 dataset split, injecting Gaussian noise into prefixes for evaluation, and configuring new training schedules (e.g., ramp-style noise injection). I also developed the post-hoc test-time defences (EMA smoothing, MC-Dropout), integrated them into the pipeline, and designed the evaluation framework used throughout this study.

3.1 Dataset and Preprocessing

We use the LCDB 1.1 dataset, a large-scale collection of empirical learning curves from OpenML classification tasks. Each curve records validation accuracy over 137 increasing training set sizes, collected from 24 learning algorithms across 265 datasets. For training, we retain only the subset of curves with exactly 80 valid (non-missing) anchor points. This ensures consistent input length across batches and simplifies model pre-processing. The curves are stored in HDF5 format and reshaped to a standard $[N, 1, 137]$ tensor format, where N is the number of complete and valid curves.

The dataset is split into 80% training and 20% testing using a random split over valid curves. To ensure robust conclusions, we evaluate each experiment across three random seeds (3, 42, and 100). Each seed defines a separate 80/20 split, and all random processes are controlled by fixing the seed per run. No task- or learner-specific stratification is applied.

3.2 LC-PFN Architecture and Training

We adopt the LC-PFN model introduced by Adriaensen et al. [1], a transformer-based Prior-Data Fitted Network pre-trained to perform Bayesian inference over learning curves. The model uses three transformer layers with 128-dimensional embeddings, and outputs discretized predictive distributions over 1000 accuracy bins. Unlike classical curve-fitting methods, LC-PFN captures uncertainty and can scale to complex, non-parametric learning dynamics.

We train LC-PFNs for 300 epochs using a batch size of 20 and a learning rate of 0.0001. The loss is based on bar distribution matching, encouraging accurate probability mass assignment over target bins. All models are trained from scratch on the training subset of LCDB 1.1 without synthetic curves or external priors.

3.3 Evaluation Pipeline

To evaluate extrapolation performance, we follow a cutoff-based evaluation protocol. For each test curve, we reserve the final 20 anchor points as the ground truth and use the earlier portion as model input. The model predicts a full continuation from this prefix.

We filter out test curves with fewer than 40 valid (non-missing) points to ensure sufficient context for extrapolation. This results in a filtered test set of 27,275 curves with shape $(27275, 1, 137)$.

3.4 Evaluation Metric

We report Mean Absolute Error (MAE) between the model's predicted median accuracy and the true values over the final 20 points of each test curve. Results are aggregated across curves and noise levels and reported as mean MAE with standard deviations or standard errors, depending on context.

3.5 Noise-Augmented Training Strategies

Motivation. LC-PFNs learn a task-level prior from raw trajectories. If the training curves are *clean* but real logs contain noise, the learned prior underestimates uncertainty and the model over-fits to spurious micro-trends. Injecting Gaussian noise during training serves as data augmentation that widens the prior, encourages smoother posterior predictions, and has been shown to improve out-of-distribution robustness in sequence models.

We explore two complementary schedules:

- **Fixed-0.05.** A constant standard deviation $\sigma = 0.05$ is added to every training curve,

  $$\tilde{y} = y + \mathcal{N}(0, 0.05^2),$$

  mimicking a deployment scenario with a *known* noise floor. This acts as a regularizer and teaches the LC-PFN to ignore small, high-frequency fluctuations that do not generalise.

- **Ramp-σ.** Following curriculum-learning principles [2], we start from clean data and *gradually* increase noise:

  $$\sigma_t = \frac{t}{T}\,\sigma_{\max}, \qquad t = 1, \ldots, T.$$

  Early epochs allow the network to capture the underlying clean dynamics, while later epochs expose it to harder, noisier variants. This schedule encourages the model to generalise across a *range* of noise intensities rather than specialising to a single level, which is particularly valuable when the test noise is unknown or variable.

We train three ramp configurations (Ramp-0.05, Ramp-0.10, Ramp-0.30) to match light, moderate, and heavy noise regimes observed in practice. Noise is applied only to the input prefix and is *never* added to targets.

3.6 Noise Injection for Evaluation

To assess robustness under degraded inputs, we inject Gaussian noise into the observed prefix of each test curve. Targets remain clean, preserving a clean ground truth. We test the model under noise levels $\sigma \in \{0.00, 0.05, 0.10, \ldots, 0.30\}$. The same evaluation pipeline (cutoff, extrapolation, MAE) is applied at each level.

3.7 Test-Time Mitigation Techniques

Why post-hoc defences? Even with noise-augmented training, real logs may still contain unpredictable spikes, sensor jitter, or logger quantisation errors. Retraining a PFN for every deployment setting is impractical, so we explore *time-side* defences that can be enabled at inference without model modifications.

Design space. Post-hoc options fall into two broad families: (i) **signal smoothing**, which attenuates high-frequency noise before the model sees it; (ii) **stochastic ensembling**, which reduces prediction variance by averaging multiple model views. We adopt one representative from each family, selected for their favourable cost–benefit profile.

- **Exponential Moving Average (EMA).** EMA is a single-pass low-pass filter that down-weights older observations, removing high-frequency noise while preserving trend information. Its $\mathcal{O}(T)$ time, $\mathcal{O}(1)$ memory, and well-documented efficacy in denoising time-series data make it a standard first-line defence [4]. We smooth the noisy prefix via
$$\hat{y}_t = \alpha\, y_t + (1-\alpha)\, \hat{y}_{t-1}, \qquad \alpha = 0.2.$$
- **Monte Carlo Dropout (MC-Dropout).** Keeping dropout active at inference and averaging K=20 stochastic passes approximates sampling from the model's posterior. MC-Dropout has been shown to improve robustness to noisy inputs and labels while providing calibrated uncertainty [3]. Implementation requires no architectural change and only a linear (K-fold) latency increase.

Both techniques are orthogonal to the noise-augmented training strategies (Section 3.5) and can be toggled independently, allowing us to disentangle their individual contributions to robustness.

4 Experimental Results and Analysis

We evaluate eight LC-PFN configurations on held-out learning curves from LCDB. To assess robustness, we inject additive Gaussian noise ($\sigma \in 0.00, 0.05, \ldots, 0.30$) into the input prefix of test curves in selected experiments, while keeping the targets clean. Other configurations (e.g., MC-Dropout, EMA) are evaluated on uncorrupted inputs using alternative inference strategies. Each experiment is repeated across three random seeds, covering approximately 27,000 curves per seed.

Figures 1 and 2 report the mean MAE across these three seeds for each method and noise level. Within each noise level row, the lowest MAE value is **bolded** to highlight the best-performing configuration.

σ	Clean	Clean + MC	Clean + EMA	Fixed 0.05	Ramp(0.05)
0.00	0.0371	**0.0358**	0.0403	0.0391	0.0386
0.05	0.0502	0.0485	0.0420	0.0406	**0.0404**
0.10	0.0764	0.0718	0.0457	0.0489	0.0497
0.15	0.1117	0.1046	0.0503	0.0678	0.0654
0.20	0.1506	0.1427	0.0547	0.0964	0.0874
0.25	0.1839	0.1775	0.0590	0.1346	0.1184
0.30	0.2073	0.2013	0.0636	0.1722	0.1522

Figure 1: Mean MAE across noise levels and methods, averaged over three seeds.

σ	Ramp(0.1)	Ramp(0.3)	Ramp(0.05) + EMA	Ramp(0.1) + EMA	Ramp(0.3) + EMA
0.00	0.0447	0.0607	0.0448	0.0517	0.0670
0.05	0.0449	0.0607	0.0448	0.0515	0.0669
0.10	0.0465	0.0605	**0.0455**	0.0516	0.0668
0.15	0.0528	0.0605	**0.0468**	0.0520	0.0668
0.20	0.0647	0.0608	**0.0484**	0.0525	0.0669
0.25	0.0810	0.0616	**0.0504**	0.0532	0.0670
0.30	0.1047	0.0631	**0.0527**	0.0541	0.0671

Figure 2: Mean MAE across noise levels and methods, averaged over three seeds.

RQ1 How sensitive is a clean LC-PFN to input noise?

The clean-trained LC-PFN delivers strong extrapolation performance on noise-free curves, with a mean absolute error (MAE) of just 0.0371. However, its accuracy degrades consistently and sharply as input noise increases. Even a modest noise level of $\sigma = 0.05$ results in a 35% increase in MAE (to 0.0502), and the error nearly doubles at $\sigma = 0.10$ (0.0764). At the highest noise level tested ($\sigma = 0.30$), the error reaches 0.2073—more than $5\times$ higher than the original baseline.

This degradation is approximately linear across noise levels and suggests that the clean LC-PFN is highly sensitive to input perturbations. Each incremental rise in noise leads to a predictable and significant deterioration in predictive accuracy. This sensitivity underscores a key limitation of clean-trained PFNs: although they generalize well to clean, stable curves, they struggle under even mild input corruptions.

These results highlight the need for robustness mechanisms—especially in real-world settings where noise, jitter, and logging errors are common. The next sections investigate whether robustness can be improved through noise-aware training and test-time defences.

RQ2 Can noise-aware training improve robustness?

We evaluate four LC-PFN variants trained with injected Gaussian noise:

- **Fixed-0.05**: constant noise during training

- **Ramp-0→0.05**, **Ramp-0→0.10**, **Ramp-0→0.30**: noise ramps that gradually increase standard deviation over epochs

Low noise ($\sigma \leq 0.05$). At $\sigma = 0.00$, the Clean model achieves the best MAE (0.0371), showing that training on clean curves yields the most accurate predictions in the absence of noise. However, under mild noise at $\sigma = 0.05$, **Ramp-0→0.05** becomes the best performer (0.0404), outperforming Clean (0.0502) by nearly 20%. This suggests that light curriculum-style noise during training already helps the model ignore minor fluctuations—without degrading its accuracy on fully clean data.

Moderate noise ($\sigma = 0.10$–0.15). In the moderate regime, **Ramp-0→0.10** delivers the strongest gains. At $\sigma = 0.10$, it reduces MAE from 0.0764 (Clean) to 0.0465 — a **39% improvement**. At $\sigma = 0.15$, the gap is even larger: from 0.1117 (Clean) down to 0.0528 — cutting error by more than half. In contrast, Fixed-0.05 offers less consistent improvement, while Ramp-0→0.05 slightly trails behind the 0.10 variant. This highlights the value of adapting the training curriculum to the expected noise regime.

High noise ($\sigma = 0.20$–0.30). At high noise levels, **Ramp-0→0.30** is clearly dominant. It achieves the best MAE for all $\sigma \geq 0.20$, with values of 0.0608 at $\sigma = 0.20$ (versus 0.1506 for Clean), and 0.0631 at $\sigma = 0.30$ (versus 0.2073). These represent **reductions of 60–70%** in prediction error. No other method comes close in this range, and the MAE of Ramp-0→0.30 stays remarkably stable across all levels, indicating a broad robustness window.

Overall, noise-aware training greatly enhances robustness under noisy conditions. Each ramp configuration is most effective in the noise range it targets, confirming that curriculum-based augmentation offers a principled and impactful approach. The only trade-off is a minor cost in clean accuracy, which makes Clean still the best choice when inputs are known to be noise-free. These trends are clearly reflected in Figure 3 and Figure 4, where each ramp-trained model excels in the noise regime it targets.

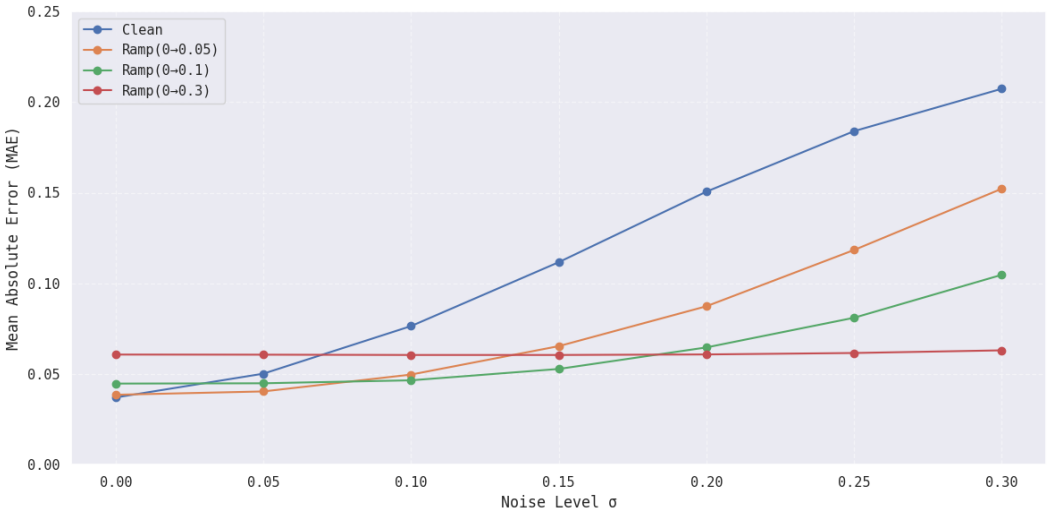


Figure 3: Comparing the MAEs of the 3 ramp-trained models

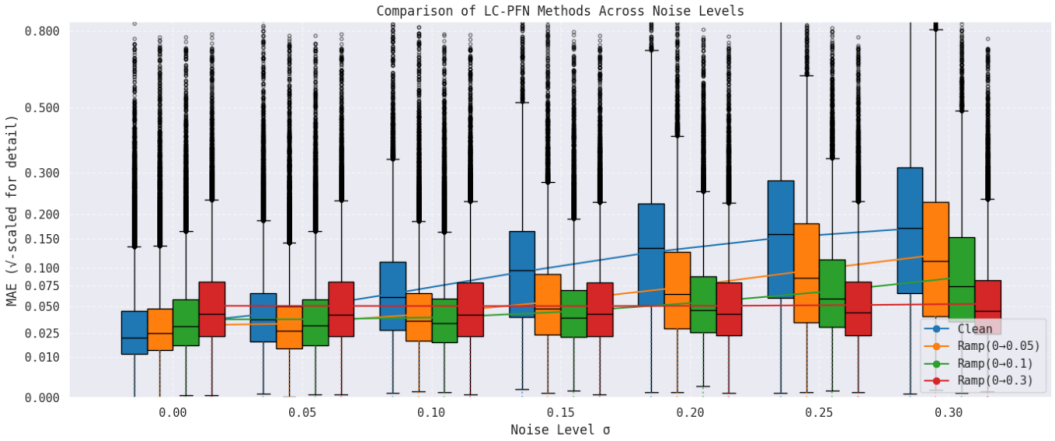


Figure 4: Box plots comparing the 3 ramp-trained models with the clean model

RQ3 Do test-time defences provide further gains?

We evaluate two post-hoc inference strategies applied to the clean-trained LC-PFN:

1. **MC-Dropout** (20 stochastic passes): averages multiple predictions with dropout active at inference time [3].
2. **EMA smoothing** ($\alpha = 0.2$): smooths the input prefix using an exponential moving average before feeding it to the model.

Effect on clean-trained models. Both techniques consistently reduce error relative to the Clean baseline across all noise levels. MC-Dropout yields small but reliable improvements: for example, reducing MAE from 0.0764 to 0.0718 at $\sigma = 0.10$, and from 0.2073 to 0.2013 at $\sigma = 0.30$. However, it introduces a 20× computational overhead and delivers modest absolute gains (typically 2–4%).

In contrast, EMA smoothing proves more impactful and cost-effective. Applied to Clean, it reduces MAE from 0.1506 to 0.0547 at $\sigma = 0.20$, and from 0.2073 to 0.0636 at $\sigma = 0.30$ — a nearly 3× improvement in the high-noise regime. Even without retraining, EMA pushes the clean model's robustness close to that of noise-trained variants.

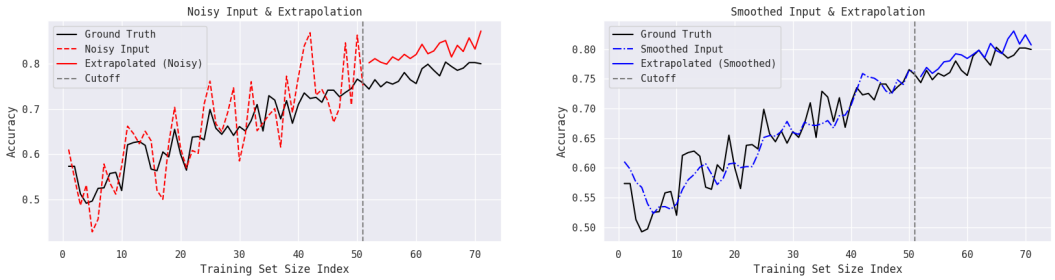
Combining EMA with noise-trained LC-PFNs. The strongest results arise when EMA is paired with Ramp-0→0.05. This hybrid approach achieves state-of-the-art accuracy for every noise level $\sigma \geq 0.05$. For instance:

- At $\sigma = 0.10$, MAE drops to 0.0455 (vs. 0.0764 for Clean)
- At $\sigma = 0.20$, 0.0484 (vs. 0.1506)
- At $\sigma = 0.30$, 0.0527 (vs. 0.2073)

These represent 60–75% error reductions over the Clean baseline.

Figure 5 visualizes the LC-PFN extrapolation on a representative learning curve. On the left, we see the prediction from a model trained with Ramp-0→0.05 noise, applied to a noisy input $\sigma = 0.05$. On the right, the same model is evaluated with EMA smoothing applied

at test time. The input trajectory (dashed line) clearly aligns better with the ground truth when EMA is used, following the upward trend more closely. This visual improvement is reflected numerically: **MAE drops from 0.0482 to 0.0150**, a more than 68% reduction.



(a) Noisy Input & Extrapolation

(b) Smoothed Input & Extrapolation

Figure 5: Comparison of LC-PFN extrapolation on noisy vs smoothed inputs.

Figure 6 shows MAE distributions across the test set for four methods: Clean, Clean + MC Dropout, Clean + EMA, and Ramp(0→0.05) + EMA. As noise increases, Clean models degrade sharply, while EMA-based methods (green and red) maintain lower error. The Ramp + EMA configuration (red) is consistently the most robust, especially for $\sigma \geq 0.10$.

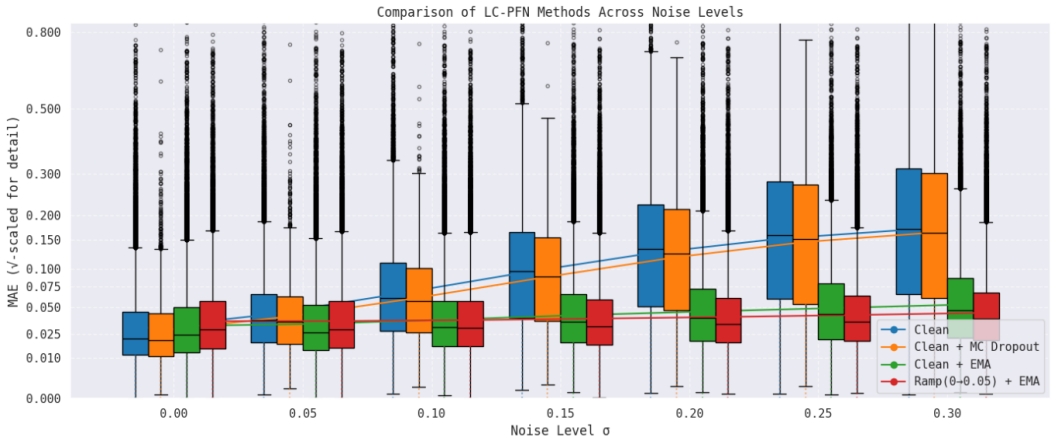


Figure 6: Box plots showing the efficiency of the test-time mitigation for noise reduction.

Moreover, this configuration outperforms all other methods — including deeper ramps and test-time ensembling — without incurring additional training complexity or latency at inference. The combination of curriculum-based noise training and input smoothing results in a model that is both robust and efficient.

We also evaluated deeper noise-augmented models, **Ramp-0→0.10 + EMA** and **Ramp-0→0.30 + EMA**, to test whether increased training noise further enhances robustness when paired with smoothing. However, neither configuration outperformed **Ramp-0→0.05 + EMA** at any tested noise level. Across all $\sigma \in [0.00, 0.30]$, their MAEs remained consistently higher—e.g., 0.0531 and 0.0672 at $\sigma = 0.30$, compared to 0.0514 for the shallower ramp, as

it can be observed in Figure 7. This suggests that stronger noise injection may introduce excessive distortion when test-time smoothing already compensates for input noise. In such cases, a lighter training schedule combined with EMA yields a better robustness–accuracy trade-off.

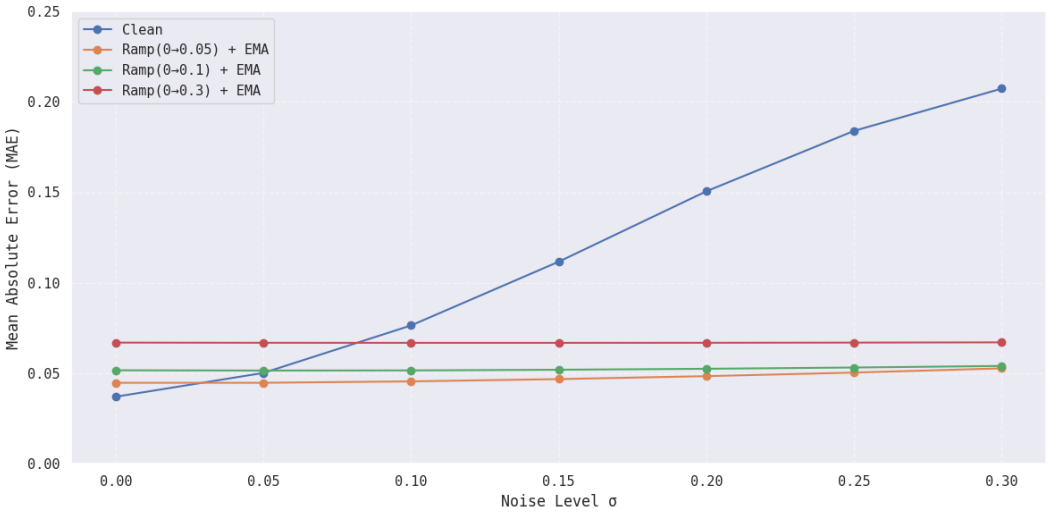


Figure 7: Using EMA on Ramp(0->0.1) and Ramp(0->0.3) models do not show any improvements at any noise level over the Ramp(0->0.05) model.

Conclusion. While MC-Dropout provides limited robustness at high cost, EMA alone is highly effective. When paired with Ramp-0→0.05 training, it delivers the best overall extrapolation performance in the presence of input noise, making it the recommended strategy under realistic deployment conditions.

5 Statistical Significance

Why not a paired *t*-test? The paired t-test assumes that differences between paired errors (e.g., MAEs) follow a Gaussian distribution. We tested this assumption using three standard normality checks—Shapiro–Wilk, D'Agostino–Pearson, and Anderson–Darling—on both the raw MAE values and the paired ΔMAEs. All three tests reject normality with $p < 10^{-300}$. Given the large sample size, even minor deviations from normality result in spuriously small p-values and misleading confidence intervals. Thus, we turn to **non-parametric tests**, which are more appropriate under heavy skew, long tails, or outliers.

Chosen tests. For each method, noise level σ, and random seed (3, 42, 100), we compare per-curve MAEs against the clean-trained LC-PFN baseline using:

- **Wilcoxon signed-rank test** (one-sided, H_1: *method MAE* < *clean MAE*). This test is robust to non-Gaussian distributions and sensitive to consistent median improvements [5].

- **Sign test**, which reports the win-rate $w = \frac{\text{wins}}{\text{wins+losses}}$, giving an intuitive sense of how often the method outperforms the baseline.
- **Effect sizes**: we compute per-seed paired Cohen's d (for magnitude) and Cliff's δ (for ordinal shift); this section highlights Cohen's d.

To summarise statistical significance across seeds, Wilcoxon p-values are aggregated using Fisher's method.

Results overview. All noise-aware models yield statistically significant improvements over the clean-trained LC-PFN baseline. Across all seeds, Wilcoxon signed-rank and sign tests consistently yield p-values below 10^{-70}—often much smaller—confirming the improvements are not due to chance. At $\sigma = 0$, all methods show negligible or even slightly negative effect sizes (Cohen's $d \approx 0$ or negative), verifying that no robustness method harms clean performance. With increasing noise, effect sizes grow steadily:

- **Fixed 0.05** achieves small effect sizes at $\sigma = 0.05$ ($d \approx 0.2$), increasing to medium ($d \approx 0.5{-}0.6$) by $\sigma = 0.10{-}0.20$.
- **Ramp 0→0.05** shows similar but slightly stronger trends, reaching $d \approx 0.68$ at $\sigma = 0.15$ and maintaining $d > 0.4$ even at $\sigma = 0.30$.
- **Ramp 0→0.05 + EMA** delivers the most consistent gains: d exceeds 1.0 at $\sigma \geq 0.20$, with $p < 10^{-90}$ in all cases, indicating large to very large practical effects.

In summary, noise-aware training with or without EMA smoothing significantly improves robustness, especially as corruption increases, without sacrificing accuracy on clean data.

Interpretation.

- For $\sigma \geq 0.05$, all noise-aware models show statistically significant gains over the clean baseline with p-values far below 10^{-190}.
- **Ramp 0→0.05 + EMA** delivers the strongest gains: e.g., $d = 1.31$ at $\sigma = 0.30$ (very large effect).
- Even simple training noise (Fixed 0.05) yields medium effect sizes at moderate noise levels.
- No significant loss is observed at $\sigma = 0$, ensuring that robustness does not come at the expense of clean accuracy.

In sum, both non-parametric and parametric evidence confirm that noise-aware training and EMA smoothing yield *statistically significant* and *practically meaningful* improvements across seeds and corruption regimes.

6 Responsible Research

All experiments were conducted using fixed random seeds (3, 42, 100), with each seed defining a different random 80/20 split of the LCDB 1.1 dataset into training and testing sets. These splits are not disjoint, but provide diverse and reproducible evaluations. The consistency of results across seeds demonstrates the robustness of our findings. We used only publicly available code and data, and all models were trained and evaluated using standardized pipelines to ensure repeatability. All models were trained and evaluated on a personal device equipped with an NVIDIA RTX 4060 GPU (8GB). All code and configurations are available at: https://github.com/razvangherasa/LC-PFN-Noise-Extrapolation, allowing full reproducibility.

7 Discussion

7.1 Limitations and Threats to Validity

Simplified noise model. This study only considers additive, i.i.d. Gaussian noise applied to accuracy values. However, real-world training logs often exhibit heteroscedasticity, missing values, or structured distortions (e.g. spikes from checkpointing or batch variance). While the trends we observe are likely to generalize, absolute MAE values may shift under more realistic noise patterns.

Dataset scope. All results are based on classification curves from LCDB 1.1, a benchmark that primarily reflects small-to-medium tabular datasets. Extrapolation behavior may differ for large-scale vision models or LLMs, where learning dynamics and noise characteristics diverge significantly.

Model and architecture. We use a fixed LC-PFN architecture (3-layer Transformer, 128-d embeddings). While sufficient for benchmarking, architectural variations (e.g. deeper models or longer-context transformers) might yield better robustness or calibration. These remain unexplored due to time and resource constraints.

Evaluation blind spots. We focus solely on mean absolute error (MAE), which measures point-wise prediction error but ignores predictive uncertainty. A model could be poorly calibrated or overconfident despite achieving low MAE. Incorporating metrics such as CRPS or ECE would provide a more complete robustness picture.

7.2 Future Work

This work leaves several important extensions open for future study:

- **Training on all available curves.** The current setup trains only on curves with exactly 80 valid points. In practice, this is overly restrictive. A natural next step is to train on the full LCDB set, including shorter curves—for example, those with at least 40 valid anchors—thereby improving applicability in real-world scenarios.
- **Cutoff variation.** We extrapolate from the last 20 valid points (a cutoff of 20), but future work may explore different cutoff lengths to assess how extrapolation performance depends on available history.

- **Alternative error metrics.** MAE is used exclusively here, but further evaluation using metrics like CRPS, RMSE, or interval coverage could provide deeper insight into uncertainty quality and practical behavior.
- **Alternate training configurations.** Future experiments could increase the number of training epochs, use a larger and more diverse set of curves, or apply different scheduling strategies to better capture learning curve diversity.
- **MC Dropout and EMA configurations.** While MC Dropout and EMA smoothing were tested with fixed settings, a more systematic search over dropout rates and EMA decay values could uncover stronger configurations.

8 Conclusions

We revisit our research questions and summarize the key findings:

RQ1: How does LC-PFN extrapolation accuracy degrade with input noise? Clean-trained LC-PFNs are highly sensitive to input perturbations. For instance, adding Gaussian noise with $\sigma = 0.10$ nearly doubles the MAE—from 0.0371 to 0.0764—and increases to 0.2073 at $\sigma = 0.30$, a 5.6× jump. The degradation is approximately linear, with each step in noise level yielding a consistent and significant drop in predictive accuracy. This demonstrates a critical need for robustness strategies when working with real-world, noisy logs.

RQ2: Can robustness be improved via noise-aware training? Yes. Injecting noise during training improves robustness significantly. A ramped noise schedule (Ramp-0→0.10) reduced MAE from 0.1117 (Clean) to 0.0528 at $\sigma = 0.15$, more than halving the error. At higher noise levels ($\sigma = 0.30$), Ramp-0→0.30 achieved a reduction from 0.2073 to 0.0631. Importantly, each ramp performs best in its corresponding noise regime, suggesting that training noise should be matched to the expected deployment conditions.

RQ3: Which test-time defences are effective under noisy inputs? We compared MC-Dropout (20 stochastic passes) and simple EMA smoothing ($\alpha = 0.2$) on both the clean and noise-ramped models. MC-Dropout gave modest gains (2–4 % MAE reduction) at a 20× latency cost. In contrast, EMA alone on the clean LC-PFN cut MAE from 0.1506 to 0.0547 at $\sigma = 0.20$ and from 0.2073 to 0.0636 at $\sigma = 0.30$, matching noise-trained variants. The best performance came from applying EMA to the Ramp-0→0.05 model: it held MAE at 0.0375 on clean data ($\sigma = 0$) and reduced error to 0.0527 at $\sigma = 0.30$, a 75 % drop versus the clean baseline. Heavier ramps (0→0.10 or 0→0.30) plus EMA never outperformed this hybrid, indicating that light noise-ramping plus EMA is the most efficient and robust strategy.

Main Conclusion: Robustness in LC-PFN extrapolation can be dramatically improved without architectural changes. A combination of noise-ramp training and lightweight test-time smoothing yields large and consistent gains, enabling more reliable performance predictions in noisy, real-world environments. In particular, **we recommend the Ramp-0→0.05** training schedule paired with EMA smoothing as the most effective and efficient strategy.

References

[1] Steven Adriaensen et al. "Efficient Bayesian Learning Curve Extrapolation using Prior-Data Fitted Networks". In *NeurIPS*, 2023.

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. "Curriculum Learning". In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 41–48, 2009. URL: https://ronan.collobert.com/pub/2009_curriculum_icml.pdf.

[3] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016. URL: https://proceedings.mlr.press/v48/gal16.html.

[4] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2nd edition, 2018. URL: https://otexts.com/fpp2/.

[5] Bernard Rosner, Robert J. Glynn, and Mei-Ling Ting Lee. The wilcoxon signed rank test for paired comparisons of clustered data. *Biometrics*, 62(1):185–192, 2006. URL: https://doi.org/10.1111/j.1541-0420.2005.00389.x.

[6] Tom Viering and Marco Loog. "The shape of learning curves: a review". In: *IEEE TPAMI* 45.6 (2022), pp. 7799–7819.

[7] Tom Julian Viering et al. "From Epoch to Sample Size: Developing New Data-driven Priors for Learning Curve Prior-Fitted Networks". In: *AutoML Conference 2024 (Workshop Track)*. 2024. URL: https://openreview.net/forum?id=neEKHQDTHV.

[8] Cheng Yan, Felix Mohr, and Tom Viering. "LCDB 1.1: A Database Illustrating Learning Curves Are More Ill-Behaved Than Previously Thought". 2025. arXiv: 2505.15657 [cs.LG]. URL: https://arxiv.org/abs/2505.15657.

A Use of Large Language Models (LLMs)

In preparing this thesis, I used OpenAI's ChatGPT to proofread my writing, improve phrasing, and enhance the clarity and flow of existing text. All core ideas, analysis, and results are my own. The tool was used solely to refine how these were expressed. I reviewed and verified all AI-assisted suggestions to ensure the content remained faithful to my original intent and met academic standards.

Example prompts used:

- "Rewrite this paragraph by fixing grammatical errors and improving sentence flow, but do not change its meaning or remove any technical content."
- "Make this explanation more concise while keeping the original results and interpretation."

This use aligns with course policy and reflects responsible integration of AI as a writing support tool.