Investigating Structural Barriers to Machine Learning Performance in Analog Circuit Design

Jonah Ezekiel
Department of Computer Science
Stanford University
jezekiel@stanford.edu

***Abstract*—Machine Learning (ML) holds significant promise for automating analog circuit design, particularly the inverse design task of mapping circuit performance metrics to circuit parameters. However, recent benchmarks have revealed that ML models perform well on simple circuits, but degrade significantly on more complex circuits, despite similar training procedures and data availability. This paper presents a new framework for investigating structural properties of circuit design spaces that may underlie these performance gaps. We extend the AICircuit benchmark framework with modular tools for scoring, filtering, and analyzing training data, and apply these tools on three hypothesized structural barriers to learnability: local linearity, local variance, and non-injectivity. Our experiments across several circuit types and model architectures suggest that while local linearity and variance show inconsistent or weak relationships to model performance, non-injectivity emerges as a moderately predictive factor for loss. We further evaluate the use of Mixture Density Networks (MDNs) to address non-injectivity by modeling uncertainty. Our findings suggest that point wise structural scores alone are insufficient to explain ML failure in analog design, highlighting the need for richer, possibly joint or higher-order metrics. This work contributes both a diagnostic framework and empirical insights towards understanding when and why ML succeeds or fails in analog circuit design.**

I. Introduction

Analog and radio frequency (RF) circuits underpin a wide range of critical technologies, including cellular communications [2], radar [3], and antenna systems [4]. Unlike digital circuits, which scale predictably with advances in semiconductor technologies, the development of high-performance analog circuits remains heavily reliant on labor-intensive custom design [5]. Analog circuit design traditionally involves two key stages:

- **Topology Selection:** Determining the overall circuit architecture. Topologies can be classified as:
  - *Homogeneous*: Composed of sub-circuits with similar functionality (e.g., amplifiers, mixers).
  - *Heterogeneous*: Composed of sub-circuits with diverse functionalities that interact in complex ways (e.g., multi-block transceivers).
- **Circuit Sizing:** Fine-tuning component parameters (e.g., resistances, capacitances, transistor widths) through extensive parameter sweeps to achieve the desired performance.

Recent work has explored the application of machine learning (ML) tools to automate both topology selection and circuit sizing [6]–[8]. In this paper we focus on the problem of applying ML tools for circuit sizing in the form of inverse design: given a vector of performance metrics $\mathbf{y}$ (e.g., power consumption, bandwidth, gain), the objective is to predict a corresponding set of circuit parameters $\mathbf{x}$ (e.g., resistances, capacitances) using a learned model M such that

$$\mathbf{x} = M(\mathbf{y}).$$

A significant advancement in this area is the AICircuit benchmark [1], which provides

- A large-scale dataset covering both homogeneous and heterogeneous analog circuits, including amplifiers, mixers, oscillators, multi-block transmitters, and receivers.
- Supervised learning tasks linking parameter vectors to their simulated performance metrics.
- Standardized training pipelines and evaluation metrics across multiple model classes.

While AICircuit demonstrates that ML models can match or surpass traditional optimization methods on simple circuits, substantial performance degradation is observed on more complex circuits. The best performing models, neural architectures such as Transformers [10] and multilayer perceptrons, can achieve near perfect accuracy on circuits like low-noise amplifiers but struggle with heterogeneous systems like transceivers and even some homogeneous circuits like power amplifiers [1]. Notably, prior evaluations have primarily focused on mean predictive error, without a detailed examination of the underlying factors contributing to these failures. The AICircuit authors speculate that increasing non-linearity in the design space correlates with diminished learnability, echoing the intuitions of experienced analog designers who typically focus on highly linear, predictable regions of the design space when performing manual sizing [5]. However, this hypothesis has not been quantitatively validated. In this work, we address this gap by

- Extending the AICircuit framework with tools to quantify structural properties of a circuit design space and investigating how these properties affect model training and generalization.
- Implementing modules for three structural characteristics:
  - Local Linearity
  - Local Variance
  - Non-Injectivity

Ultimately we hope our framework for systematic diagnostics can offer a more principled approach to model selection, dataset refinement, and performance expectation in ML-driven circuit design.

II. System Design

Our experimental pipeline is built directly on top of the AICircuit benchmark. We extended the framework with modules for

- **Data characteristic scoring**: Algorithms that assign a numerical score to each data point, quantifying specific structural properties. The framework is modular, allowing new scoring functions to be easily incorporated.
- **Dataset Filtering**: Mechanisms to remove a specified fraction of data points with poor structural scores, enabling comparisons between models trained on filtered versus unfiltered datasets.
- **Result Analysis**: Tools to compute evaluation metrics and generate plots, including:
  - Performance comparisons between filtered and unfiltered models.
  - Score distributions across different circuit types.
  - Scatter plots correlating structural scores with prediction error.

While the scoring and analysis modules are relatively straightforward, the filtering module addresses a more subtle question: how does the presence of structurally problematic points in the training data affect overall model behavior? Simply training a model on the full dataset and analyzing performance post hoc may obscure interactions between problematic and non-problematic points during learning.

A. Structural Properties Investigated

In our work we implement three scoring algorithms, focusing on the following key structural properties:

- **Local Linearity**: Inspired by analog designers' intuition, we hypothesize that circuits with locally linear performance-to-parameter mappings are easier to learn.
- **Variance**: Measures local variability in the target space; higher variance may indicate sensitivity to parameter perturbations.
- **Non-Injectivity**: Captures the extent to which multiple different parameter configurations map to similar performance metrics. Non-injective mappings are hypothesized to degrade supervised learning performance [8].

Prior studies have shown that non-injective datasets can lead neural networks to predict the conditional mean of possible outputs, which results in degraded performance [8]. Mixture Density Networks (MDNs) [9] have been proposed as a solution to model such ambiguities by quantifying predictive uncertainty. We integrate MDNs into the AICircuit framework for comparison with other models and for further studying non-injectivity in the data.

Algorithm 1 Local-Linearity Scoring Algorithm

Require: Dataset $\mathcal{D}$ with features $\mathbf{X}$ and targets $\mathbf{Y}$
Ensure: Local-linearity scores $\mathbf{s}$
Normalize features $\mathbf{X}$ to $[0, 1]$ range
Build KD-tree $\mathcal{T}$ on $\mathbf{X}$
Initialize score list $\mathbf{s} \leftarrow []$
for each point $x_i \in \mathbf{X}$ **do**
  Find k nearest neighbors $\mathcal{N}(x_i)$
  Fit independent linear regressions for each target dimension
  Compute R^2 scores for each regression
  Record mean R^2 as the score for x_i
  Append score to $\mathbf{s}$
end for
return $\mathbf{s}$

Algorithm 2 Variance Scoring Algorithm

Require: Dataset $\mathcal{D}$ with features $\mathbf{X}$ and targets $\mathbf{Y}$
Ensure: Variance scores $\mathbf{s}$
Normalize features $\mathbf{X}$ to $[0, 1]$ range
Build KD-tree $\mathcal{T}$ on $\mathbf{X}$
Initialize score list $\mathbf{s} \leftarrow []$
for each point $x_i \in \mathbf{X}$ **do**
  Find k nearest neighbors $\mathcal{N}(x_i)$
  Compute variance for each target dimension within $\mathcal{N}(x_i)$
  Record mean variance across dimensions
  Append score to $\mathbf{s}$
end for
return $\mathbf{s}$

B. Scoring Algorithm Framework

All scoring algorithms we implement share a common structure:

1) Normalize features and targets to the $[0, 1]$ range to enable meaningful distance computations.
2) Construct a KD-tree [11] in feature space to efficiently query neighboring points.
3) Compute a structural score for each data point based on its local neighborhood.

The three scoring algorithms are detailed in Algorithm 1, Algorithm 2, and Algorithm 3. Note that the local linearity score can take on negative values, representing highly nonlinear regions where the model fits the data worse than a horizontal line at the mean. Further, the range of possible values for the non-injectivity score is by design variable, dependent on the number of output dimensions.

Algorithm 3 Non-Injectivity Scoring Algorithm

Require: Dataset $\mathcal{D}$ with features $\mathbf{X}$ and targets $\mathbf{Y}$
Ensure: Non-injectivity scores $\mathbf{s}$
Normalize features $\mathbf{X}$ and targets $\mathbf{Y}$ to $[0, 1]$ range
Build KD-tree $\mathcal{T}$ on $\mathbf{X}$
Initialize score list $\mathbf{s} \leftarrow []$
for each point $x_i \in \mathbf{X}$ **do**
    Find all neighbors within radius ϵ
    Compute L2 norm between all pairs of target vectors
    Record maximum pairwise L2 norm as the score
    Append score to $\mathbf{s}$
end for
return $\mathbf{s}$

III. Results

Due to computational constraints, we limited our experiments to a subset of circuits and models available in the AICircuit framework. The selected circuits include the Common-Source Voltage Amplifier (CSVA), Power Amplifier (PA), and Voltage controlled oscillator (VCO). The PA and VCO were chosen to represent more complex, difficult-to-learn circuits, while the CSVA serves as a baseline due to its relative simplicity. The models evaluated include Random Forest, Multilayer Perceptron (MLP), Transformer, and Mixture Density Network (MDN). These were selected as the top-performing models in the AICircuit benchmark, with the MDN additionally representaing a new contribution of our work.

For each structural scoring algorithm, we present three types of results: (1) the distribution of scores across different circuits, (2) average validation loss for models trained on the full versus filtered datasets, and (3) scatter plots correlating score and loss. For brevity, plot type (3) is shown only for the CSVA and PA circuits using Transformer and Random Forest models.

A. Linearity

We begin by analyzing the impact of local linearity on model performance. For each circuit considered, we computed local-linearity scores as described in Algorithm 1, filtered out the bottom 25% of data points (those with the lowest R^2 scores), and trained models on both the remaining subset, and the full dataset.

Figure 1 shows the distribution of linearity scores. As expected, the CSVA exhibits a wide range of scores, reflecting its smooth and regime-independant behaviour. The PA, by contrast, shows distinct clusters of high linearity, which we hypothesize correspond to its known operation in discrete modes. The VCO also demonstrates relatively high linearity. These results already cast doubt on the hypothesis that poor local linearity explains model failures: both the PA and VCO circuits, despite being more difficult to learn, are more locally linear than the CSVA.

This is confirmed in Figure 2, where we compare average validation loss across models and datasets. No consistent trend emerges indicating that filtering based on linearity improves generalization. Although, interestingly models trained on the filtered subset perform comparably or even slightly better than those trained on the full dataset. This suggests that the least linear data points may not contribute positively to the learning process.

Finally, Figure 3 presents scatter plots of model loss versus linearity scores for the CSVA and PA circuits. Contrary to our expectations, there is no statistically meaningful negative correlation between score and loss. Increased local linearity does not appear to predict improved model performance.

B. Variance

Next, we analyze the impact of local variance on model performance. For each circuit considered, we computed variance scores as described in Algorithm 2, filtered out the top 25% of data points (those with the highest variance scores), and trained models on both the remaining subset, and the full dataset.

Figure 4 reveals that the PA has significantly higher variance scores than the CSVA or VCO, immediately suggesting variance may be a stronger candidate for explaining model difficulty, at least for the PA. However, like linearity the average validation losses across models trained on the unfiltered versus the filtered dataset show no robust trend, as shown in Figure 5. This indicates that points of high local variance, like points of high local linearity, may not contribute positively to the learning process. Figure 6 presents scatter plots of variance score versus model loss. Although not statistically conclusive, a weak positive correlation is visible: as expected, higher variance appears to correlate with greater loss. The effect is more pronounced in models trained on the filtered dataset, as would be expected if high-variance regions degrade learning stability.

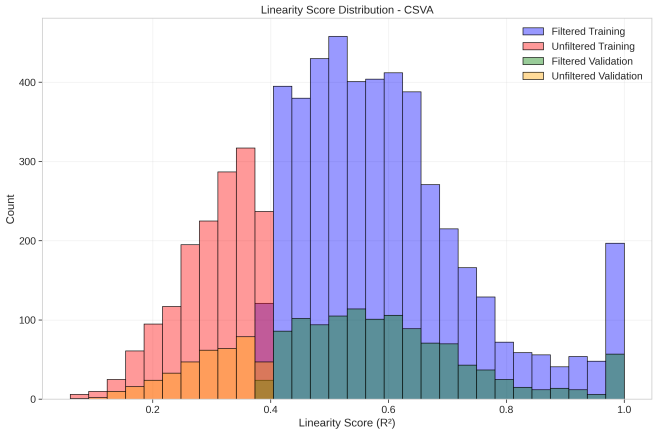
C. Non-Injectivity

Finally, we analyze the impact of non-injectivity on model performance. For each circuit considered, we computed non-injectivity scores as described in Algorithm 3, filtered out the top 25% of data points (those with the highest non-injectivity scores), and trained models on both the remaining subset, and the full dataset.

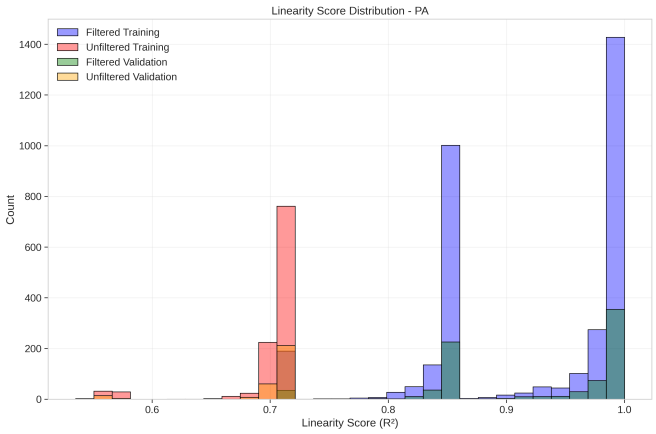
Figure 7 shows no clear pattern across circuit types that would implicate non-injectivity as a main differentiating factor. As in prior sections, Figure 8 confirms that filtered and unfiltered models perform similarly, again with no consistent benefit or harm to using the full dataset. However, the scatter plots in Figure 9 tell a different story. Here we observe a statistically significant positive correlation between non-injectivity score and mode loss, suggesting non-injectivity poses a fundamental learning challenge.

D. Mixture Density Network Results

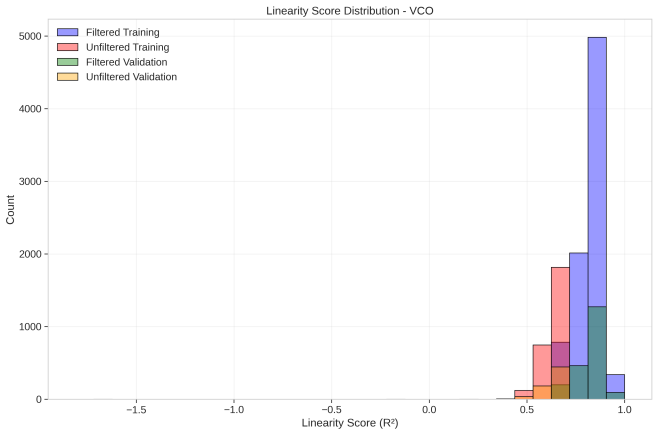
We further investigate the role of non-injectivity by integrating the MDN into our experimental framework. MDNs are specifically designed to model multi-modal output distributions and quantify predictive uncertainty, making them



(a) CSVA

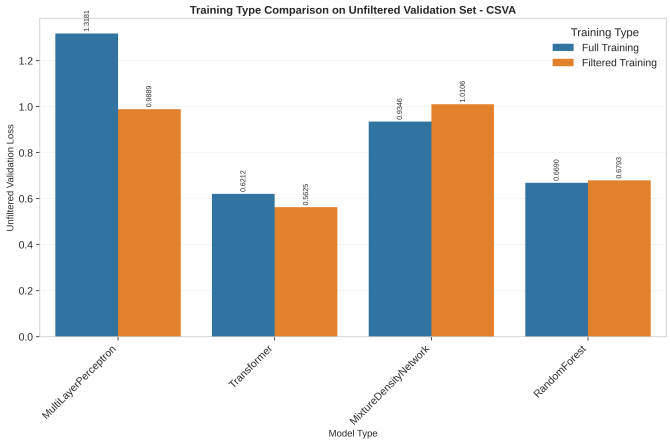


(b) PA

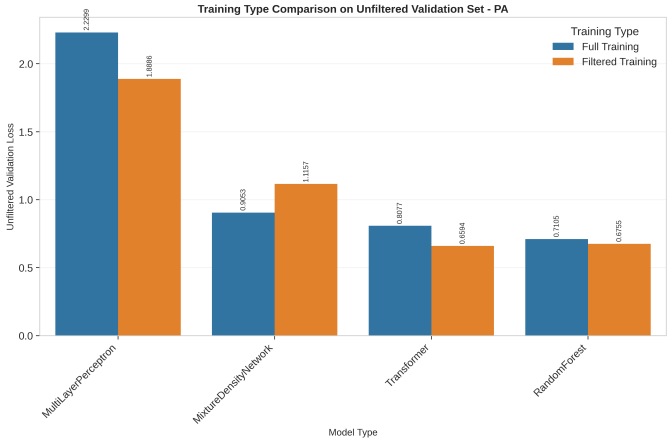


(c) VCO

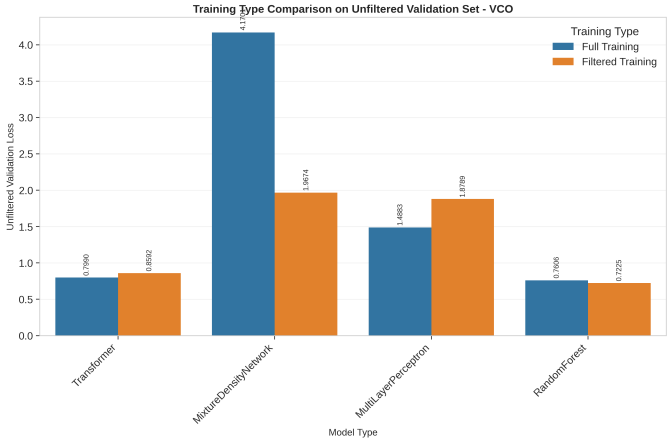
Fig. 1: Distribution of local-linearity scores across the CSVA, PA, and VCO circuits. Higher scores indicate greater local linearity, which is hypothesized to improve model learnability. Colors indicate whether each point was retained (e.g., blue) or removed (e.g., red) during filtering.



(a) CSVA

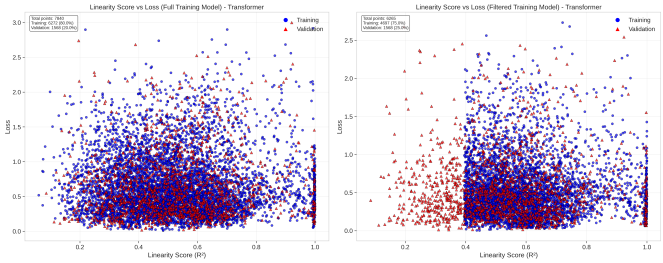


(b) PA

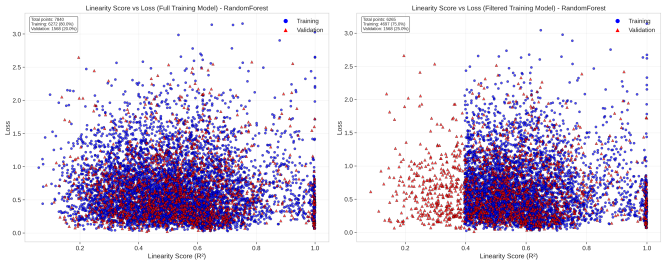


(c) VCO

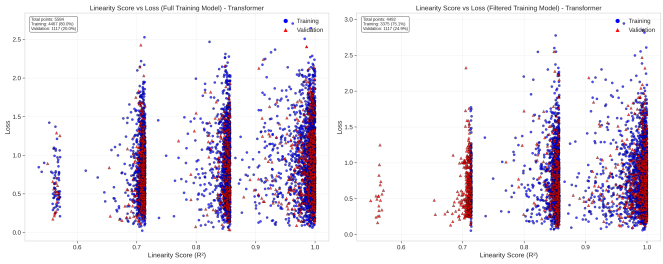
Fig. 2: Comparison of model performance on the full validation set for each circuit, with and without local-linearity-based data filtering. Bars show the average validation loss across models trained on either the full dataset (blue) or the worst-25%-filtered dataset (orange), as scored by local linearity. Lower values indicate better predictive performance.



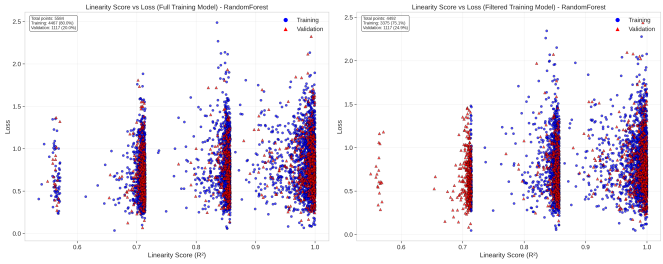
(a) CSVA – Transformer model



(b) CSVA – Random Forest model

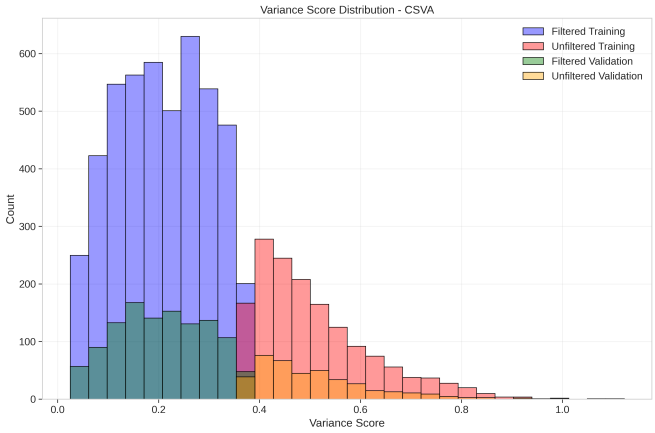


(c) PA – Transformer model

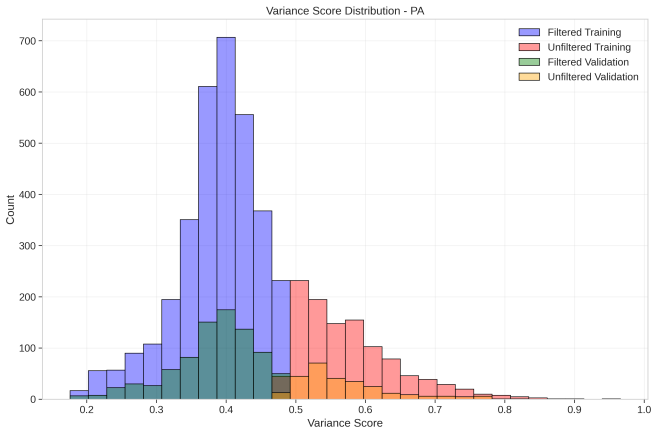


(d) PA – Random Forest model

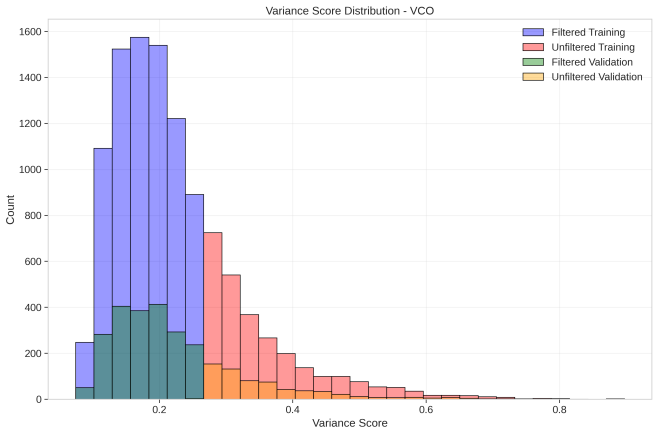
Fig. 3: Scatter plots showing linearity score versus loss for each combination of circuit and model architecture. Each plot compares full-dataset training on the left and filtered training (with the worst 25% of data removed based on linearity score) on the right. Blue points indicate training loss, and red triangles represent validation loss.



(a) CSVA

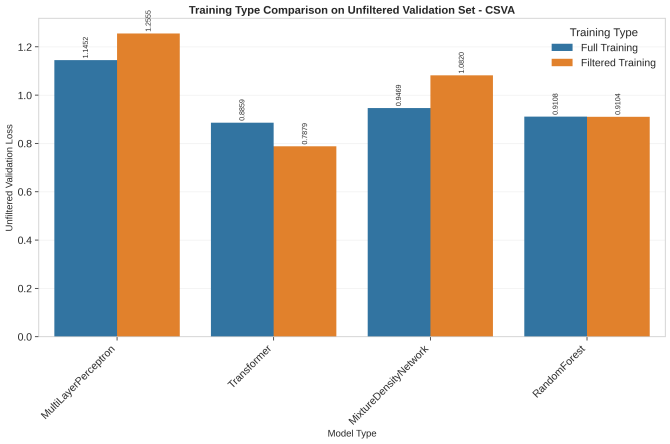


(b) PA

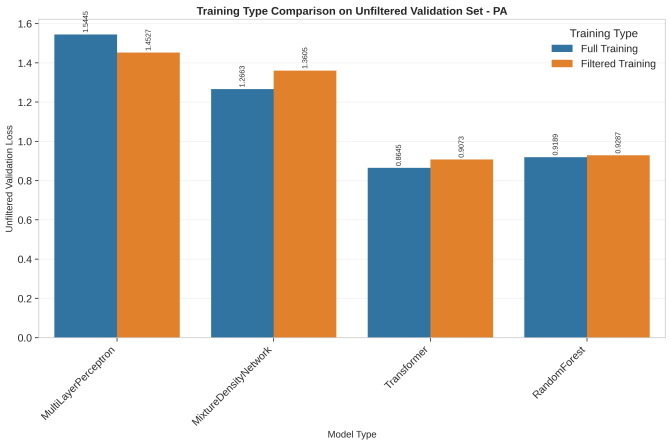


(c) VCO

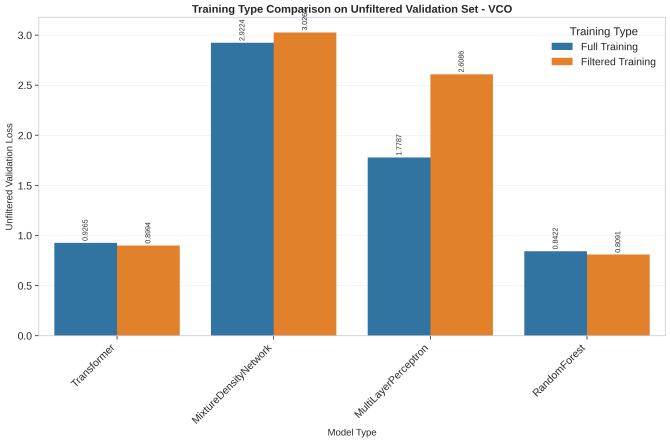
Fig. 4: Distribution of local variance scores across the CSVA, PA, and VCO circuits. Higher scores indicate greater local variance, which is hypothesized to pose a challenge for model learnability. Colors indicate whether each point was retained (e.g., blue) or removed (e.g., red) during filtering.



(a) CSVA

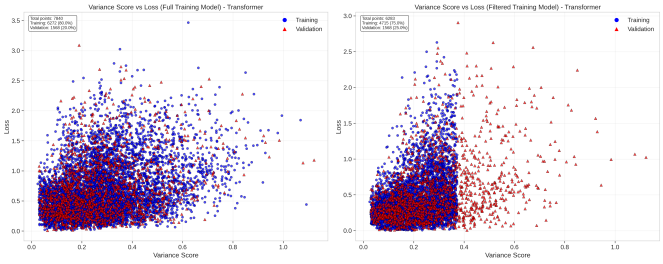


(b) PA

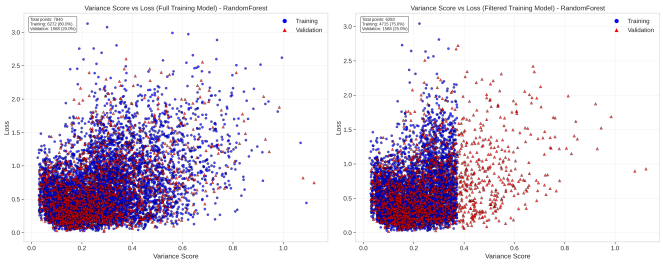


(c) VCO

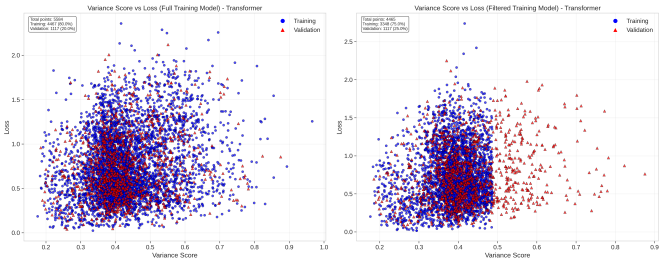
Fig. 5: Comparison of model performance on the full validation set for each circuit, with and without local-variance-based data filtering. Bars show the average validation loss across models trained on either the full dataset (blue) or the worst-25%-filtered dataset (orange), as scored by local variance. Lower values indicate better predictive performance.



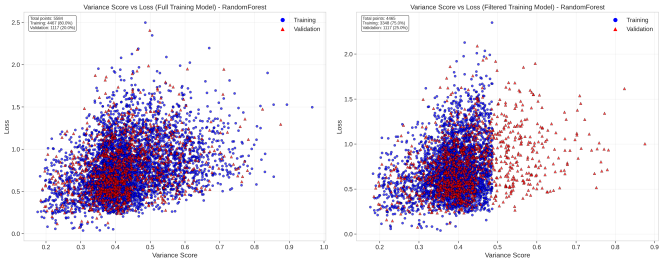
(a) CSVA – Transformer model



(b) CSVA – Random Forest model

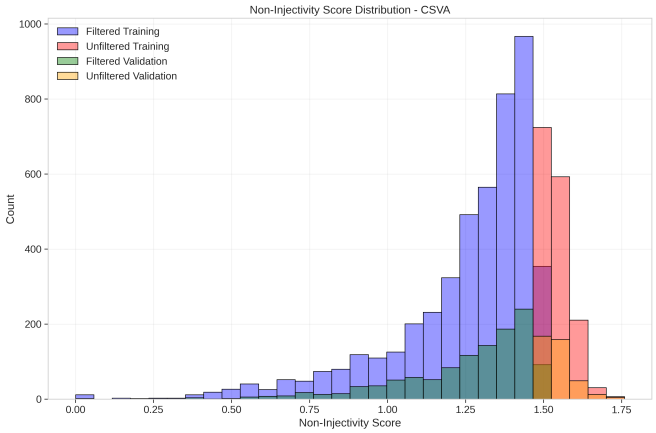


(c) PA – Transformer model

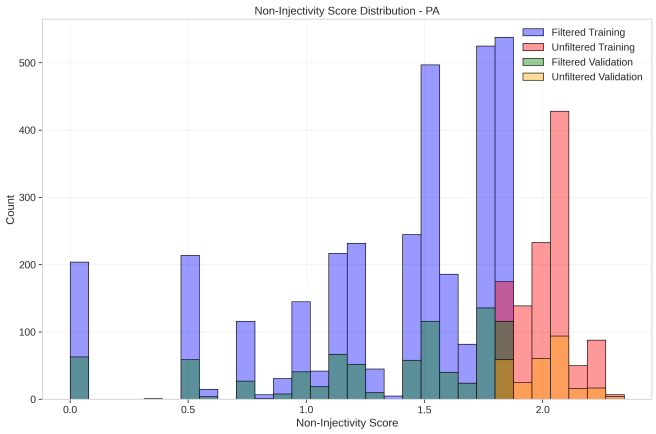


(d) PA – Random Forest model

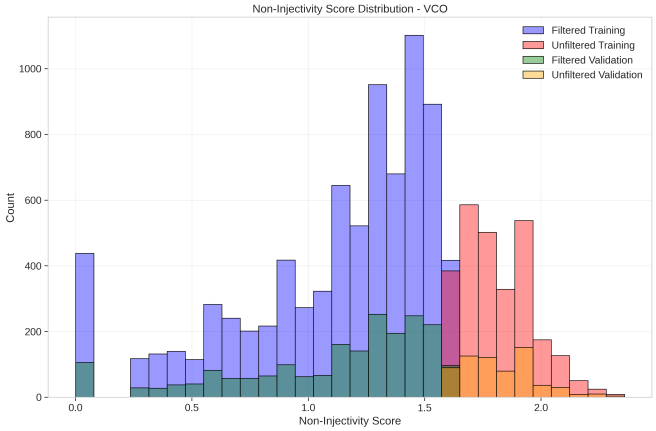
Fig. 6: Scatter plots showing variance score versus loss for each combination of circuit and model architecture. Each plot compares full-dataset training on the left and filtered training (with the worst 25% of data removed based on variance score) on the right. Blue points indicate training loss, and red triangles represent validation loss.



(a) CSVA

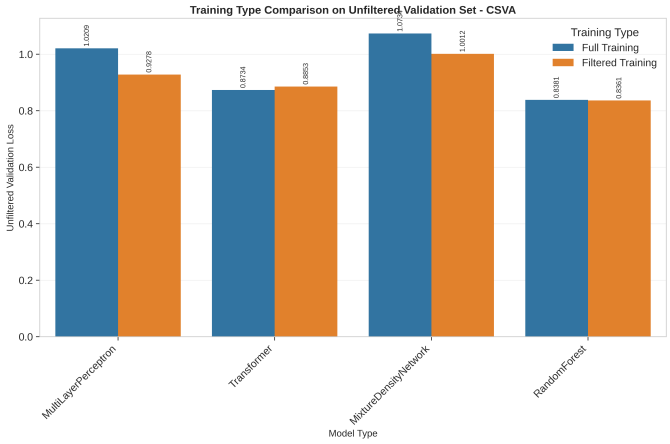


(b) PA

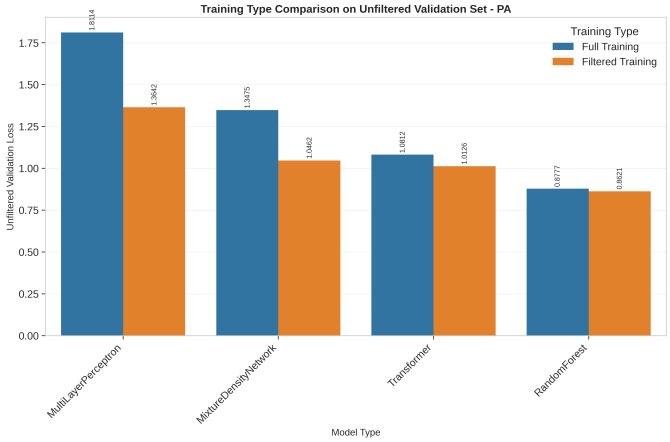


(c) VCO

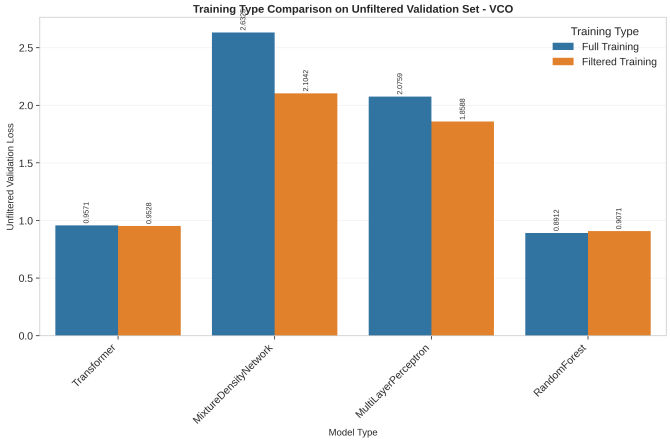
Fig. 7: Distribution of local non-injectivity scores across the CSVA, PA, and VCO circuits. Higher scores indicate greater non-injectivity, which is hypothesized to pose a challenge for model learnability. Colors indicate whether each point was retained (e.g., blue) or removed (e.g., red) during filtering.



(a) CSVA

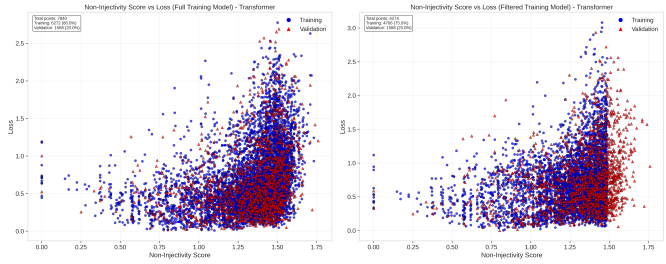


(b) PA

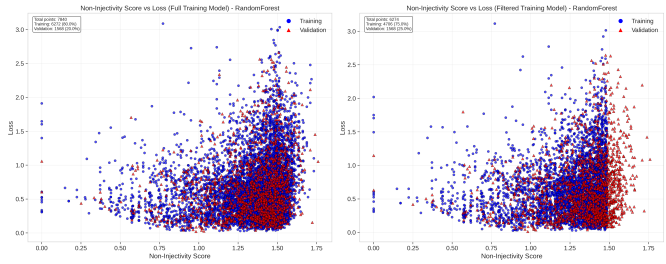


(c) VCO

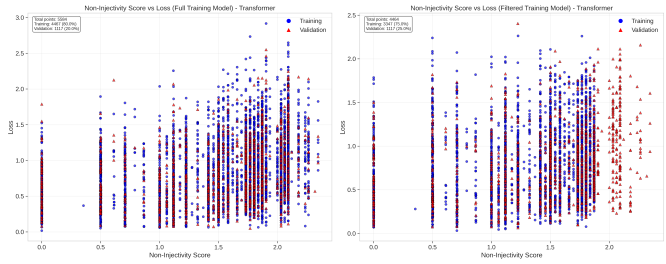
Fig. 8: Comparison of model performance on the full validation set for each circuit, with and without local-non-injectivity-based data filtering. Bars show the average validation loss across models trained on either the full dataset (blue) or the worst-25filtered dataset (orange), as scored by non-injectivity. Lower values indicate better predictive performance.



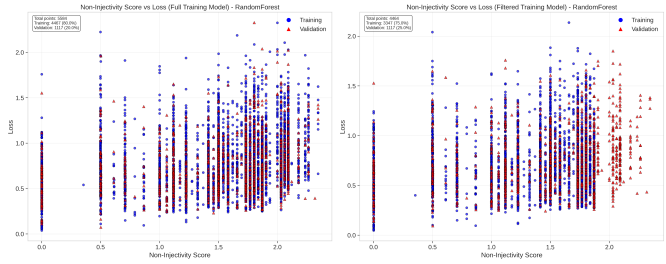
(a) CSVA – Transformer model



(b) CSVA – Random Forest model



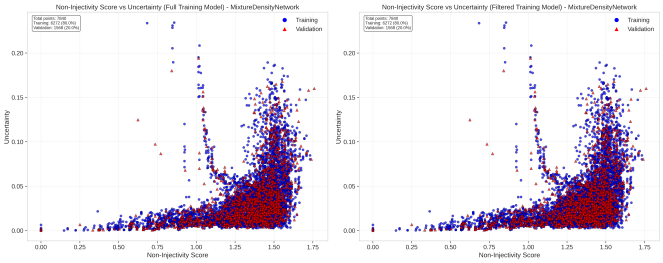
(c) PA – Transformer model



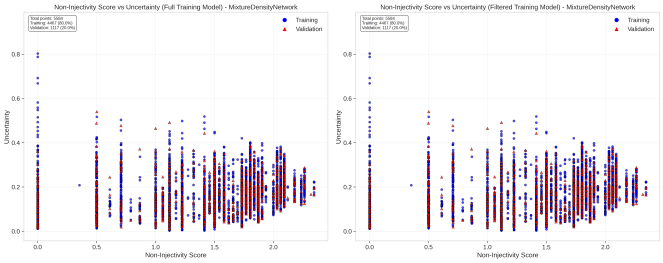
(d) PA – Random Forest model

Fig. 9: Scatter plots showing non-injectivity score versus loss for each combination of circuit and model architecture. Each plot compares full-dataset training on the left and filtered training (with the worst 25% of data removed based on non-injectivity score) on the right. Blue points indicate training loss, and red triangles represent validation loss.

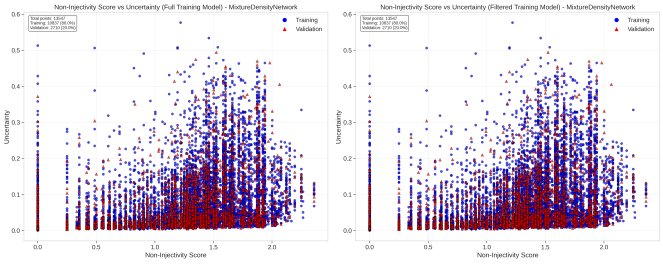
a natural candidate for addressing challenges posed by non-injective mappings. In Figure 10, we plot non-injectivity score versus MDN model uncertainty for each circuit, comparing results from models trained on the full dataset and the filtered dataset. For the CSVA and VCO circuits, we observe a clear positive correlation between non-injectivity score and uncertainty, consistent with our hypothesis. This suggests that MDNs are capable of identifying and adapting to structural ambiguity in the data. However, this relationship breaks down in the case of the PA circuit, where uncertainty does not track with non-injectivity score. The reason for this deviation remains unclear.



(a) CSVA



(b) PA



(c) VCO

Fig. 10: Scatter plots of non-injectivity score versus predictive uncertainty from a Mixture Density Network (MDN) for each circuit type. The left panel shows results from the model trained on the full dataset; the right panel shows results from the model trained after filtering out the top 25% most non-injective points. Blue circles represent training points and red triangles represent validation points.

IV. Conclusion

We have extended the AICircuit framework with modular tools for diagnosing structural properties of analog circuit design spaces and evaluating their impact on machine learning performance. Our contributions include implementations

of scoring algorithms for local-linearity, variance, and non-injectivity, mechanisms for dataset filtering based on these scores, and tools for analyzing and visualizing results.

Our experiments do not reveal any single structural property that statistically predicts model performance across all circuits and models and filtering the data seemingly had no statistical effect on model performance. However, they do demonstrate that such properties can interact with learning dynamics in subtle and context-dependent ways with model performance. Local linearity, while commonly assumed to be beneficial, did not correlate with improved generalization or reduced error. Local variance showed weak but interpretable effects, and non-injectivity stood out as the only property with a statistically significant relationship to model loss.

On a broader level, our findings raise important questions about the nature of failure in ML applied to analog design and beyond. Despite access to vast simulation-accurate datasets and state-of-the-art models, we still have no clear analytic explanation for why performance degrades sharply on complex circuits. This reflects a larger challenge in the field of machine learning: understanding why models succeed or fail remains elusive, in part because the behavior of high-capacity models is difficult to interpret. Our results suggest that point-wise interpretable metrics may be insufficient to capture the full complexity of model brittleness. More sophisticated abstractions, perhaps drawing from causal reasoning or information theory, may be necessary to advance toward deeper understanding.

Future work should explore additional structural properties that may correlate more strongly with model performance, including properties derived from circuit topology or domain-specific priors. Further, rigorous statistical validation is needed to confirm or refute whether linearity, variance, and non-injectivity have meaningful predictive value. This would be especially useful for linearity, which proved non-predictive in our experiments despite being strongly supported by designer intuition, making this a potentially significant result. Another promising direction is to study the joint influence of multiple structural properties: perhaps interactions between linearity, variance, non-injectivity, or other structural properties may together provide an explanation. Expanding the scope of evaluation to include more circuit types and models will also help to distinguish generalizable trends from data-specific artifacts.

Ultimately, this work takes a step towards a more diagnostic science of ML for analog circuit design. While we do not claim to offer conclusive answers, we provide a practical framework for future researchers to probe structure-performance relationships more systematically and invite the community to build on these tools in pursuit of deeper understanding.

REFERENCES

[1] A. Mehradfar et al., "AICircuit: A Multi-Level Dataset and Benchmark for AI-Driven Analog Integrated Circuit Design," 2024.

[2] S. Rangan, T. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, 2014.

[3] S. H. Dokhanchi et al., "A mmWave automotive joint radar-communications system," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[4] W. Hong et al., "Study and prototyping of practically large-scale mmWave antenna systems for 5G cellular devices," *IEEE Communications Magazine*, 2014.

[5] P. E. Allen and D. R. Holberg, "CMOS Analog Circuit Design," 2011.

[6] K. Settaluri et al., "AutoCkt: Deep Reinforcement Learning of Analog Circuit Designs," DATE 2020.

[7] M. Fayazi et al., "Angel: Fully-Automated Analog Circuit Generator," *IEEE Transactions on Circuits and Systems I*, 2023.

[8] D. Krylov et al., "Learning to design analog circuits to meet threshold specifications," ICML 2023.

[9] C. Bishop, "Mixture Density Networks," 1994.

[10] A. Vaswani et al., "Attention is All You Need," NeurIPS 2017.

[11] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, 1975.