Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082976

ARTICLE

Conformal Prediction for Reliable Hyperparameter Selection in Sparse FIR Filter Design

Mohammed Hassan Alnemari[1,2,*], Abdelrahman Osman Elfaki[3], Anas Bushnag[1] and Mohamed Hussien Mohamed Nerma[1]

[1]Department of Computer Engineering, Faculty of Computer Science and Information Technology, University of Tabuk, Tabuk, Saudi Arabia

[2]AIST Research Center, University of Tabuk, Tabuk, Saudi Arabia

[3]Department of Computer Science, Faculty of Computer Science and Information Technology, University of Tabuk, Tabuk, Saudi Arabia

*Corresponding Author: Mohammed Hassan Alnemari. Email: mnemari@ut.edu.sa

Received: 07 April 2026; Accepted: 04 June 2026; Published: 23 July 2026

ABSTRACT: Sparse finite impulse response (FIR) filters reduce computational cost on resource-constrained devices, but selecting the sparsification threshold λ is typically left to grid search or hand tuning. We propose a two-stage method: a 67,331-parameter surrogate network predicts (A_p, A_s, S) (passband ripple in dB, stopband attenuation in dB, sparsity in %) from a filter specification and a candidate λ, and split conformal prediction (CP) calibrates ± intervals around each prediction. We then select λ by minimizing a worst-case penalty computed on the conservative ends of the intervals (the upper bound on A_p and the lower bound on A_s). On 10,000 test specifications the method reaches 76.5% specification satisfaction, near-parity with grid search (78.4%) with a 1.9× speedup, while point-prediction surrogates reach only 39.4%. On feasible specifications (where any grid λ satisfies both constraints), the method reaches 97.6%. Stratified (Mondrian) conformal prediction lifts standard CP coverage from 67%–75% to 95.5%, and adaptive recalibration brings passband coverage to 91.3%. The procedure transfers without modification to iteratively reweighted least squares (IRLS) sparsification (76.6%) and to highpass (79.2%) and bandpass (52.4%) filters. The implementation runs on a central processing unit (CPU) and is suitable for edge deployment; code and data are public.

KEYWORDS: Conformal prediction; sparse FIR filter; edge IoT; surrogate modeling; uncertainty quantification; hyperparameter optimization

1 Introduction

Sparse finite impulse response (FIR) filters reduce multiply-accumulate operations by zeroing small coefficients, cutting both memory footprint and power consumption compared to dense designs [1]. This translates directly into longer battery life and smaller silicon area on the microcontroller-class hardware that increasingly hosts on-device digital signal processing (DSP) and tiny machine learning (TinyML) pipelines [2,3]. Recent work on convex sparse FIR design [4,5] has focused on the sparsification *algorithm* (how to threshold or regularize); the orthogonal question of how to *tune* the sparsification hyperparameter, given a chosen algorithm, remains an open problem in practice. Achieving sparsity while satisfying frequency-domain specifications (passband ripple, stopband attenuation) requires careful selection of the sparsification hyperparameter λ, which controls how aggressively coefficients are thresholded; existing pipelines either grid-search λ exhaustively or set it by hand.

Our focus is *hyperparameter selection methodology*, not the development of novel sparsification algorithms. We demonstrate the framework primarily with hard thresholding (for computational efficiency on edge devices) and additionally with the iteratively reweighted least squares (IRLS) method (Section 4.8). The contribution is a reliable hyperparameter selection framework applicable to sparsification methods with tunable parameters.

In practice, traditional approaches rely on exhaustive grid search or manual tuning and lack systematic treatment of the reliability-sparsity tradeoff. Surrogate modeling accelerates selection but point predictions achieve only 39.4% specification satisfaction on 10,000 test specifications due to systematically over-optimistic predictions. Direct λ prediction achieves only 27.0% satisfaction.

We introduce split conformal prediction (CP) to provide rigorous uncertainty quantification for surrogate-based λ optimization. Conformal prediction constructs prediction intervals with distribution-free, finite-sample coverage guarantees under the exchangeability assumption, aiming to ensure that true performance lies within predicted bounds with user-specified probability (e.g., 95%).

By using worst-case interval bounds (upper bound on passband ripple, lower bound on stopband attenuation), we enable conservative λ selection that raises satisfaction from 39.4% to 76.5%.

A subtlety we address explicitly throughout the paper is that the standard split conformal coverage guarantee assumes exchangeability between calibration and test points [6,7], but our test-time λ^* is chosen by an optimizer that consults the conformal intervals themselves. Coverage measured at *grid-sampled* λ (matching the calibration distribution) and coverage measured at *optimization-selected* λ^* are therefore distinct quantities. We report both throughout, and we use stratified (Mondrian) calibration [6] together with one round of adaptive recalibration in the spirit of conformal prediction under distribution shift [8] to restore conditional coverage at λ^*.

Section 4 validates these results on 10,000 held-out specifications, confirming a 37 percentage point (pp) improvement over point surrogates and 97.6% satisfaction on feasible specifications. User-selectable confidence levels (90%, 95%, 99%) enable reliability-sparsity tradeoffs without retraining.

Contributions:

- A conformal prediction framework for FIR filter hyperparameter optimization that provides distribution-free prediction intervals for conservative λ selection. The framework is validated across test sets from 100 to 10,000 specifications, achieving 97.6% satisfaction on feasible designs (Section 4.4).
- Stratified conformal prediction and adaptive recalibration addressing calibration-test distribution mismatch, restoring coverage from 67%–75% to 95.5% and closing the passband gap (A_p: 76.8% → 91.3%).
- Validation on both hard thresholding and IRLS sparsification (76.6% vs. 76.5% satisfaction), and generalization to highpass (79.2%) and bandpass (52.4%) filter types.
- A central processing unit (CPU)-only implementation with 67,331 parameters and no graphics processing unit (GPU), suitable for embedded and edge deployment.

2 Related Work

Classical approaches for sparse FIR filter design include convex methods (ℓ_1 regularization [5,9], weighted ℓ_1 penalties), iterative methods (IRLS [10]), and greedy coefficient selection. Convex methods provide deterministic sparsity control, while iterative methods like IRLS approximate non-convex ℓ_0 objectives. For example, Nerma et al. [4] explored ℓ_1/ℓ_2 elastic-net penalties achieving 32.8% sparsity via interior-point methods, but requiring >2 s per design due to iterative solvers, prohibitive for real-time or interactive applications. IRLS methods similarly require multiple optimization iterations (typically 5–15), with each iteration solving a weighted least-squares problem. While these approaches maintain rigorous

signal processing foundations (Chebyshev optimality [11], structured sparsity), their computational cost limits deployment on resource-constrained edge devices.

These classical methods address *sparsification algorithm design*, whereas our work addresses *hyperparameter selection* for a given sparsification method. The two problems are orthogonal: classical methods still require hyperparameter tuning (regularization weights, penalty parameters, stopping criteria), and our conformal prediction framework can in principle be applied to any of these methods. We demonstrate this with both hard thresholding (primary, for edge deployment) and IRLS (Section 4.8).

Neural networks have been applied to FIR filter coefficient optimization [12] and narrow transition-band filter design via deep learning [13], demonstrating improved performance over classical window methods. However, none of these methods provides reliability guarantees; a deployed model may silently produce filters that violate specifications on unseen inputs.

Vovk et al.'s conformal prediction framework [6] provides distribution-free uncertainty quantification without distributional assumptions on the data. The split variant [7], made accessible by Angelopoulos and Bates [14], partitions data into training and calibration sets, computing empirical quantiles of residuals to construct prediction intervals with guaranteed coverage. We build on this split formulation, extending it to multi-output surrogate regression for filter design.

While standard split conformal prediction [7] provides marginal coverage guarantees under exchangeability, real-world applications often require *conditional* coverage adapted to subpopulations or difficulty levels. *Mondrian conformal prediction* [6] addresses this by partitioning the feature space into strata and calibrating separate quantiles for each stratum, providing conditional validity guarantees. Recent work on *conformalized quantile regression* [15] extends this to regression settings with heteroscedastic noise. *Flexible distribution-free conditional predictive bands* [16] further generalize conditional conformal methods using density estimators. In our filter design context, we apply Mondrian CP by stratifying on transition width $\Delta F = F_{\text{stop}} - F_{\text{pass}}$, which correlates strongly with prediction error magnitude.

Gaussian processes (GPs), neural networks, and ensemble methods accelerate expensive black-box optimization by learning surrogate models of objective functions. Bayesian optimization (BO) [17] provides principled uncertainty quantification through GP posterior distributions, with acquisition functions such as expected improvement and the upper confidence bound (UCB) guiding sequential exploration-exploitation. While GP-based methods provide calibrated uncertainty natively, they impose distributional assumptions (typically Gaussian posterior), scale as $O(n^3)$ for exact inference, and require per-specification sequential optimization. Conformal prediction offers a complementary, distribution-free alternative: prediction intervals are computed once via calibration and applied to all specifications in constant time, enabling batch deployment on edge devices. We empirically compare both approaches in Appendix A.

Conformal prediction has recently appeared in wireless channel estimation [18], acoustic source localization [19], and time-series forecasting [20]. In each case, CP replaces parametric assumptions (Gaussian noise, channel stationarity) that rarely hold in practice. Our work extends this to FIR filter design, where the relationship between λ and filter performance is similarly hard to model parametrically. To our knowledge, this is the first such application.

3 Methods

Fig. 1 summarizes the complete pipeline. The method has a one-time *offline* phase, in which we generate a labeled dataset of (Spec, λ) pairs by running the Parks-McClellan (PM) algorithm with hard thresholding, train the surrogate network, and compute the conformal calibration quantiles; and a per-specification *online*

phase, in which the trained surrogate and stored quantiles are queried to choose λ^* and produce the final sparse filter. The remainder of Sections 3.1–3.5 expands each block.

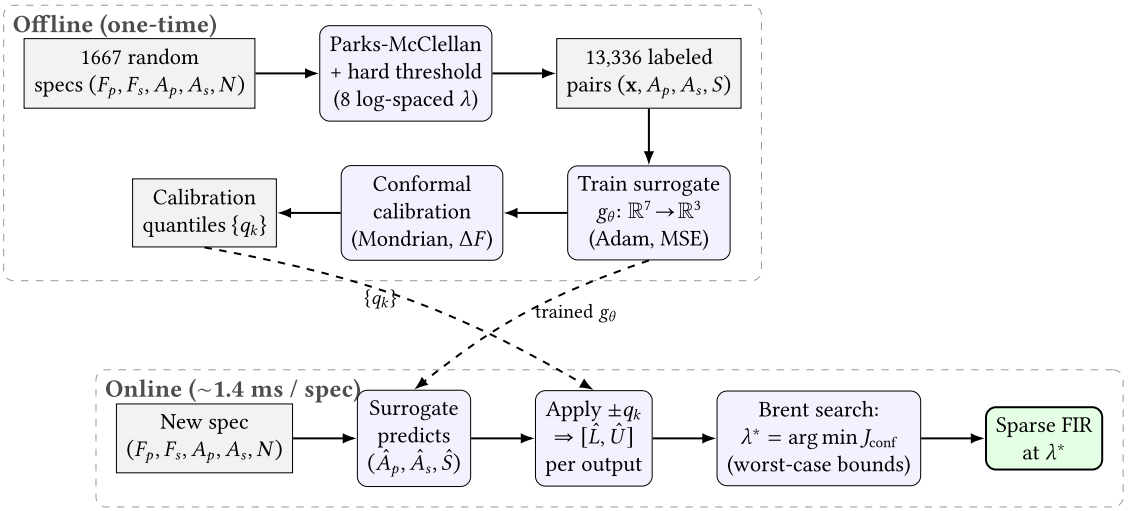

Figure 1: Methodology overview. **Offline**: a fixed pool of random filter specifications is run through Parks-McClellan with hard thresholding at 8 log-spaced λ values to produce 13,336 labeled $(\mathbf{x}, \mathbf{y})$ pairs; a 128-256-128 MLP surrogate is fit on 80% of these pairs, and Mondrian conformal calibration on the remaining 20% produces stratum-specific quantiles $\{q_k\}$ keyed on transition width ΔF. **Online**: for a new specification, the surrogate emits a point prediction, the calibration quantiles produce ± intervals, and a 1-D Brent search picks the λ^* that minimizes a worst-case conservative penalty. The full online step takes ~1.4 ms on CPU.

3.1 Problem Formulation

We consider the design of sparse lowpass FIR filters characterized by five parameters: passband and stopband edge frequencies F_{pass} and F_{stop}, maximum passband ripple A_p (dB), minimum stopband attenuation A_s (dB), and filter order N.

Our approach consists of two stages:

1. **Dense design**: Compute optimal equiripple filter $\mathbf{h}_{\text{PM}}$ using the Parks-McClellan algorithm [11,21].
2. **Sparsification**: Apply hard thresholding with parameter $\lambda \in [0, 1]$:

$$h_{\text{sparse}}[n] = \begin{cases} h_{\text{PM}}[n] & \text{if } |h_{\text{PM}}[n]| \geq \lambda \cdot \max_k |h_{\text{PM}}[k]| \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In practice, our grid uses 8 hand-picked non-uniformly spaced values

$$\lambda \in \{1,\ 2,\ 4,\ 6,\ 8,\ 10,\ 20,\ 50\} \times 10^{-4},$$

covering $[10^{-4},\ 5 \times 10^{-3}]$ with five points in the small-λ decade $[10^{-4}, 10^{-3}]$ and three in the large-λ decade $[10^{-3}, 5 \times 10^{-3}]$ (further detailed in Section 3.5). Concentrating points in the small-λ regime is deliberate: at $\lambda < 10^{-3}$ a small change in threshold flips a single coefficient and meaningfully changes the achieved sparsity, whereas at $\lambda > 10^{-3}$ the response saturates because most retained coefficients have already moved to zero. The same 8-point grid is reused for highpass and bandpass filters (Section 4.9); we verified that this range brackets the satisfaction-maximizing λ^* for >99% of the multi-band test specifications.

The challenge is selecting λ to maximize sparsity while satisfying specifications. Traditional grid search evaluates L candidate values, requiring L frequency response computations per design.

For a sparse filter $\mathbf{h}_{\text{sparse}}$ obtained at threshold λ, the achieved passband ripple and stopband attenuation are computed in dB directly from the filter's magnitude response $|H(e^{j\omega})|$ on a 1024-point frequency grid:

$$A_p^{\text{achieved}} = 20\log_{10}\left(\max_{\omega\in\Omega_p}|H(e^{j\omega})|\right) - 20\log_{10}\left(\min_{\omega\in\Omega_p}|H(e^{j\omega})|\right) \quad [\text{dB}], \tag{2}$$

$$A_s^{\text{achieved}} = -20\log_{10}\left(\max_{\omega\in\Omega_s}|H(e^{j\omega})|\right) \quad [\text{dB}], \tag{3}$$

where $\Omega_p = [0, F_{\text{pass}}]$ and $\Omega_s = [F_{\text{stop}}, F_s/2]$ are the passband and stopband regions. Sparsity is $S = 100 \cdot |\{n : h_{\text{sparse}}[n] = 0\}|/(N+1)$ in percent. A design at λ^* satisfies the specification if and only if both inequalities hold:

$$\text{satisfied}(\lambda^*) \iff A_p^{\text{achieved}}(\lambda^*) \le A_p^{\text{spec}} \ \wedge \ A_s^{\text{achieved}}(\lambda^*) \ge A_s^{\text{spec}}. \tag{4}$$

The specification satisfaction rate over a test set $\mathcal{T}$ is then $\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}} \mathbb{1}\{\text{satisfied}(\lambda_i^*)\}$. All ripple and attenuation numbers reported throughout the paper use Eqs. (2) and (3); satisfaction rates use Eq. (4).

3.2 Surrogate Model Architecture

We train a surrogate neural network $g_\theta : \mathbb{R}^7 \to \mathbb{R}^3$ that predicts filter performance metrics from specifications and λ:

$$g_\theta(\text{Spec}, \lambda) = (A_p^{\text{pred}}, A_s^{\text{pred}}, S^{\text{pred}}) \tag{5}$$

where S is sparsity percentage. The input vector is:

$$\mathbf{x} = [F_{\text{pass}}, F_{\text{stop}}, A_p^{\text{spec}}, A_s^{\text{spec}}, N, \Delta F, \lambda] \tag{6}$$

with transition bandwidth $\Delta F = F_{\text{stop}} - F_{\text{pass}}$.

The surrogate takes as input only quantities available before filter design (specifications and candidate λ), predicting performance without access to Parks-McClellan outputs. The resulting conformal interval widths (e.g., ±8.59 dB for stopband attenuation) reflect genuine uncertainty inherent in specification-to-performance prediction. Fig. 2 shows the input-output structure and how the surrogate connects to the conformal calibration pipeline.

The novelty of this surrogate-plus-calibration design relative to prior neural FIR work [12,13] is twofold: (i) the surrogate predicts *achieved performance* from the specification *before* running Parks-McClellan, rather than learning filter coefficients directly, which keeps the model small enough for CPU inference and decouples it from any specific sparsification algorithm; and (ii) the conformal calibration quantile q_k is the only object that varies with the desired confidence level, so the same surrogate supports any target coverage at no additional training cost.

The architecture is a multilayer perceptron (MLP) with widths 128-256-128, rectified linear unit (ReLU) activations, and a mean squared error (MSE) loss:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \|\mathbf{y} - g_\theta(\mathbf{x})\|_2^2 \tag{7}$$

Table 1 summarizes the architecture search over varying depths and widths. The 128-256-128 bottleneck configuration achieved the lowest validation MSE while maintaining a compact parameter count suitable for CPU-only inference at ~0.05 ms per forward pass.

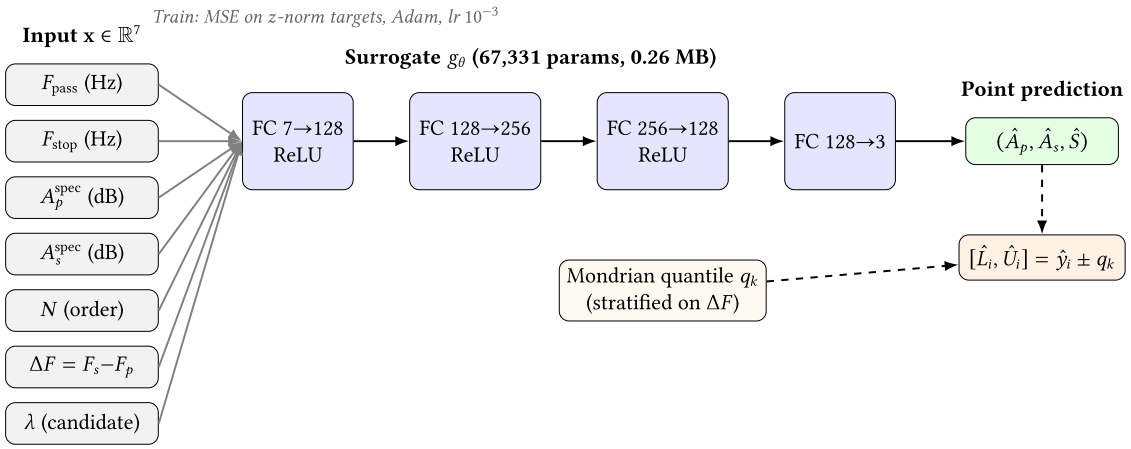

Figure 2: Surrogate I/O and calibration pipeline. The 7-dimensional input **x** stacks five filter-specification fields, the derived transition width ΔF, and the candidate threshold λ. A 128-256-128 MLP with ReLU activations produces a 3-dimensional point prediction $(\hat{A}_p, \hat{A}_s, \hat{S})$ in ~0.05 ms on CPU. At deployment, the Mondrian calibration quantile q_k (selected by the test sample's ΔF stratum) widens the point prediction into a conformal interval $[\hat{L}_i, \hat{U}_i]$ used by the worst-case λ optimizer.

Table 1: Surrogate architecture comparison (validation MSE on normalized targets).

Architecture	Parameters	Val MSE	Notes
[128, 128] (2 layers)	17,667	0.0812	Underfits λ-performance curve
[128, 256, 128] (3 layers)	67,331	**0.0548**	Best tradeoff
[256, 256, 256] (3 layers)	135,939	0.0553	No improvement, 2× params
[128, 256, 128, 64] (4 layers)	75,651	0.0561	Marginal overfitting

Bold marks the chosen 128-256-128 architecture (lowest validation MSE).

Training uses Adam optimizer (learning rate 10^{-3}), 80/20 train/validation split, and early stopping (patience 10 epochs). All features and labels are z-score normalized.

3.3 Uncertainty Quantification via Conformal Prediction

3.3.1 Conformal Prediction Framework

Given a trained model g_θ, conformal prediction provides prediction intervals $[\hat{L}_i, \hat{U}_i]$ for each output dimension $i \in \{A_p, A_s, S\}$ such that:

$$\mathbb{P}(y_i \in [\hat{L}_i, \hat{U}_i]) \geq 1 - \alpha \tag{8}$$

where α is the user-specified miscoverage rate (e.g., $\alpha = 0.05$ for 95% confidence).

Split conformal prediction separates model fitting from uncertainty calibration. We partition the training data into a proper training set $\mathcal{D}_{\text{train}}$ and a held-out calibration set $\mathcal{D}_{\text{cal}}$, and fit g_θ on $\mathcal{D}_{\text{train}}$ only. For each calibration sample $(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}_{\text{cal}}$ we record the absolute residual $R_{ij} = |y_{ij} - g_\theta(\mathbf{x}_j)_i|$ for each output dimension i. The calibration quantile

$$Q_i = \text{Quantile}_{1-\alpha'}\left(\{R_{ij}\}_{j=1}^{|\mathcal{D}_{\text{cal}}|}\right), \qquad \alpha' = \frac{\lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil}{n_{\text{cal}}}, \tag{9}$$

uses the finite-sample correction α' rather than the nominal $1-\alpha$ to keep coverage exact for finite n_{cal}. At test time, for a new input $\mathbf{x}^*$ the prediction interval is symmetric around the point prediction:

$$\hat{L}_i = g_\theta(\mathbf{x}^*)_i - Q_i, \quad \hat{U}_i = g_\theta(\mathbf{x}^*)_i + Q_i. \tag{10}$$

These intervals satisfy the marginal coverage guarantee (Eq. (8)) regardless of model architecture or data distribution; the calibration step replaces parametric assumptions about the residual distribution with an empirical quantile.

This marginal coverage guarantee holds under the assumption that the sequence $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{n_{\text{train}}+n_{\text{cal}}+1}$ is *exchangeable*, meaning the joint distribution is invariant to permutations. In practice, this requires that training, calibration, and test samples are drawn i.i.d. from the same distribution, an assumption satisfied by our uniform random sampling of filter specifications. Importantly, exchangeability is weaker than the independence assumption required by parametric methods, allowing conformal prediction to apply under weaker distributional assumptions than parametric alternatives.

The adjusted miscoverage level $\alpha' = \frac{\lceil (n_{\text{cal}}+1)(1-\alpha) \rceil}{n_{\text{cal}}}$ ensures exact coverage for finite calibration sets. For our $n_{\text{cal}} = 2668$ samples and $\alpha = 0.05$, this yields $\alpha' = 2536/2668 \approx 0.9505$, slightly increasing the quantile to account for finite-sample variability. Without this correction, empirical coverage may fall below the nominal $1-\alpha$ level for small calibration sets.

Fig. 3 shows conformal intervals for a representative filter across λ values. The shaded regions are 95% confidence intervals from split conformal prediction; under exchangeability, true performance falls within these bounds with probability ≥0.95 (empirical coverage validation in Section 4.3).

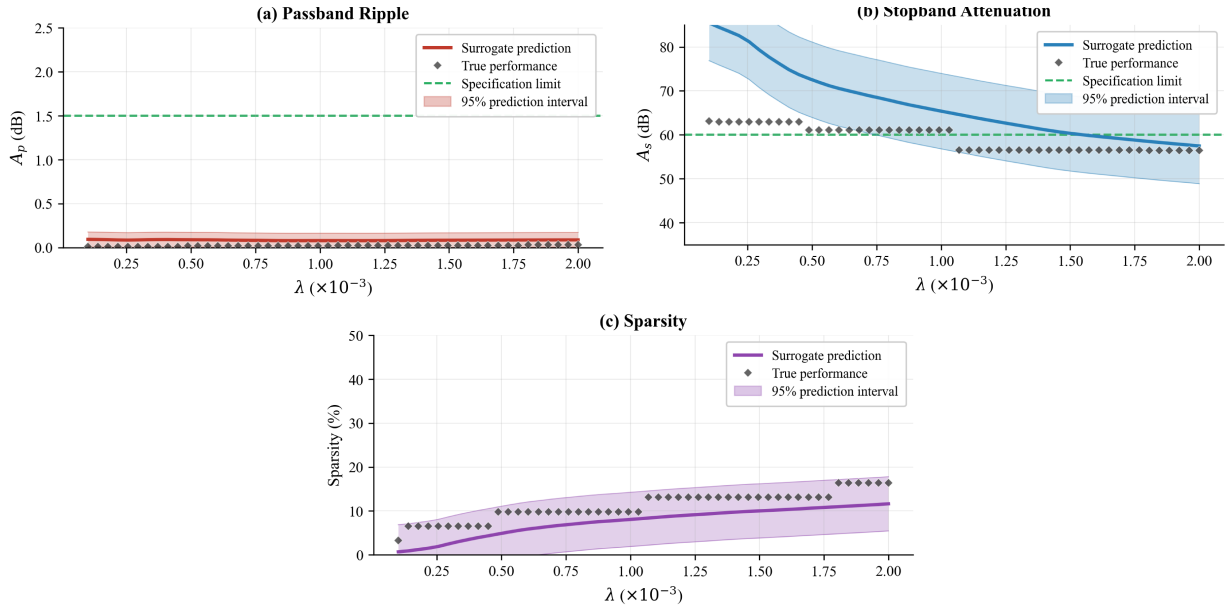

Figure 3: Conformal prediction interval visualization for representative filter specification (F_p = 120 Hz, F_s = 180 Hz, $A_p \leq 1.5$ dB, $A_s \geq 60$ dB, $N = 60$). Top row: (**a**) Passband ripple A_p and (**b**) stopband attenuation A_s predictions with 95% confidence intervals across λ values. Bottom: (**c**) Sparsity predictions with 95% intervals. Green dashed lines indicate specification constraints.

3.3.2 Conservative λ Optimization

Standard surrogate optimization uses point predictions in the penalty function:

$$J(\lambda) = w_p \cdot \max(0, A_p^{\text{pred}} - A_p^{\text{spec}})^2 + w_s \cdot \max(0, A_s^{\text{spec}} - A_s^{\text{pred}})^2 - \gamma \cdot S^{\text{pred}} \tag{11}$$

Conformal prediction enables worst-case optimization for reliability:

$$J_{\text{conf}}(\lambda) = w_p \cdot \max(0, \hat{U}_{A_p} - A_p^{\text{spec}})^2 + w_s \cdot \max(0, A_s^{\text{spec}} - \hat{L}_{A_s})^2 - \gamma \cdot \hat{L}_S \tag{12}$$

This conservative approach uses the upper bound on passband ripple (worst-case for A_p), the lower bound on stopband attenuation (worst-case for A_s), and the lower bound on sparsity (conservative sparsity estimate).

The weights $w_p = 100$, $w_s = 10$, and sparsity reward $\gamma = 0.5$ were selected to prioritize specification satisfaction over sparsity. Passband ripple violations are weighted 10× higher than stopband violations ($w_p/w_s = 10$) because passband deviations are perceptually more critical in many applications. The sparsity reward $\gamma = 0.5$ is scaled to balance the typical magnitudes of squared violations (order 10^1–10^2) against sparsity percentages (0%–100%). Table 2 validates these choices via sensitivity analysis on 500 test specifications.

Table 2: Penalty weight sensitivity analysis (500 test specifications, CP 95%).

w_p	w_s	γ	Sat. Rate (%)	Sparsity (%)
50	10	0.5	72.4	16.1
100	**10**	**0.5**	**77.2**	**14.9**
200	10	0.5	77.4	14.6
100	5	0.5	74.8	15.3
100	10	1.0	71.6	17.8
100	10	0.1	77.8	12.1

Bold marks the chosen penalty-weight configuration ($w_p = 100$, $w_s = 10$, $\gamma = 0.5$).

Satisfaction is robust for $w_p \geq 100$ (77.2–77.4%), but drops at $w_p = 50$ (72.4%), confirming that strong passband penalization is essential. The optimal λ is found via bounded scalar optimization:

$$\lambda^* = \arg \min_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} J_{\text{conf}}(\lambda) \tag{13}$$

We use SciPy's `minimize_scalar` with Brent's method, evaluating the surrogate model ~20 times (~1 ms total) vs. 8 actual filter designs in grid search (~2.6 ms).

3.4 Addressing Calibration-Test Distribution Mismatch via Stratified Conformal Prediction

It is useful to distinguish two coverage quantities that look similar but measure different things. *Grid-λ* coverage is the coverage rate when calibration and test samples both draw λ from the same 8-point grid; this is the quantity for which split conformal prediction provides its standard finite-sample guarantee [7]. *Adaptive-λ^* coverage* is the coverage rate measured at the λ^* chosen by the optimizer in Eq. (13), which selects λ^* from a continuous range rather than the grid and does so by inspecting the very intervals whose coverage we are assessing. The two quantities coincide only if optimizer-induced selection bias is negligible. The empirical coverage results in Section 4.3 reveal undercoverage for performance metrics at adaptive λ^* (A_p: 75%, A_s: 67%) while sparsity achieves near-nominal coverage (94.6%). On the same

data evaluated at grid λ values (Table 3), even standard CP achieves 95.16% for A_p. The discrepancy therefore points to a calibration-test distribution mismatch induced by the adaptive selection, rather than a fundamental flaw in the conformal framework.

Table 3: Coverage comparison: standard vs. stratified conformal prediction (95% confidence, 1632 test samples).

Metric	Standard CP	Stratified CP	Improvement
Passband Ripple (A_p)	95.16%	95.40%	+0.25 pp
Stopband Attenuation (A_s)	94.85%	95.53%	+0.67 pp
Sparsity	96.69%	96.63%	−0.06 pp

The mismatch arises because calibration residuals are computed from λ values sampled across the 8-point grid defined in Section 3.5. At test time, however, λ^* is selected adaptively via worst-case optimization (Eq. (13)). The adaptive selection creates a distribution shift: $p(\text{Spec}, \lambda^*) \neq p(\text{Spec}, \lambda_{\text{uniform}})$, introducing selection bias that violates the exchangeability assumption required for marginal coverage guarantees. Compounding this, prediction errors exhibit heteroscedasticity across the transition width $\Delta F = F_{\text{stop}} - F_{\text{pass}}$, with larger errors for narrow-transition filters due to increased design difficulty. A single global quantile therefore leads to undercoverage in high-error regimes.

To address both issues, we implement *stratified conformal prediction* (also known as Mondrian CP [6]), which calibrates separate prediction intervals for each transition width quartile, providing conditional coverage guarantees adapted to local uncertainty levels.

3.4.1 Stratified Calibration Algorithm

The stratified approach partitions the calibration set based on ΔF values (Algorithm 1):

Algorithm 1: Stratified conformal prediction calibration

Require: Calibration set $\mathcal{D}_{\text{cal}} = \{(X_i, Y_i)\}_{i=1}^{n}$, confidence level $1-\alpha$, number of strata K
Ensure: Stratum boundaries $\{B_k\}_{k=1}^{K}$, stratum quantiles $\{q_k\}_{k=1}^{K}$
1: Compute $\Delta F_i = F_{\text{stop},i} - F_{\text{pass},i}$ for all $i \in [n]$
2: Determine stratum boundaries: $B_k = \text{percentile}(\{\Delta F_i\}, [0, 25, 50, 75, 100])$
3: **for** $k = 1$ to K **do**
4: $\quad \mathcal{I}_k = \{i : B_k \leq \Delta F_i < B_{k+1}\}$ {Assign samples to stratum k}
5: $\quad$ Compute residuals: $R_i = |\widehat{f}(X_i) - Y_i|$ for $i \in \mathcal{I}_k$
6: $\quad q_k = \text{quantile}(\{R_i\}_{i \in \mathcal{I}_k}, 1-\alpha)$ {Stratum-specific quantile}
7: **end for**
8: **return** $\{B_k, q_k\}_{k=1}^{K}$

At prediction time, a new sample's stratum is determined by its ΔF value, and the corresponding stratum-specific quantile is used to construct the prediction interval.

3.4.2 Coverage Restoration Results

This evaluation uses λ values sampled from the same grid as calibration (matching the calibration distribution), unlike the empirical coverage analysis in Section 4.3, which evaluates coverage at adaptively selected λ^*. Table 3 compares empirical coverage for standard vs. stratified conformal prediction on 1632

test samples at 95% confidence. The higher standard CP coverage here (95.16% vs. 75.0% under adaptive selection) confirms that undercoverage arises from the adaptive λ selection, not from the conformal framework itself.

Stratified CP successfully restores coverage to near-nominal levels (95.5%), with the largest improvement observed for stopband attenuation (+0.67 percentage points).

Table 4 shows empirical coverage broken down by stratum, demonstrating balanced coverage across all difficulty levels.

Table 4: Per-stratum coverage for stratified conformal prediction (95% confidence).

Stratum	ΔF Range (Hz)	N	A_p Cov.	A_s Cov.	Sparsity Cov.
0 (Narrow)	[21.2, 65.1)	424	97.2%	94.1%	97.9%
1	[65.1, 96.0)	432	96.1%	95.1%	94.2%
2	[96.0, 135.9)	360	95.0%	96.9%	98.1%
3 (Wide)	[135.9, ∞)	416	93.3%	96.2%	96.6%

Stratified CP adapts interval widths to filter difficulty: narrow-transition filters (Stratum 0) receive wider intervals (±8.5 dB for A_s) compared to wide-transition filters (Stratum 3: ±7.6 dB). Per-stratum coverage ranges from 93–98%, confirming that conditional coverage is achieved across all difficulty levels. The sparsity metric maintains high coverage in both approaches, validating that the conformal framework is sound when properly calibrated for the test distribution.

The success of stratified CP confirms that the undercoverage observed initially stems from distribution mismatch rather than a fundamental failure of conformal prediction. By conditioning on transition width, we account for heteroscedastic errors and restore statistical validity. The implication is that *conditional conformal prediction is essential when calibration and test distributions differ*, as is common in optimization-driven hyperparameter selection.

3.5 Dataset Generation

We generate 1667 random filter specifications with:

- $F_{\text{pass}} \in [50, 200]$ Hz
- $F_{\text{stop}} \in [100, 300]$ Hz, ensuring $F_{\text{stop}} > F_{\text{pass}} + 10$ Hz
- $A_p \in [0.5, 3.0]$ dB
- $A_s \in [40, 80]$ dB
- $N \in [40, 80]$
- Sampling rate $F_s = 1000$ Hz

For each specification, we evaluate eight non-uniformly spaced values $\lambda \in [10^{-4},\ 5 \times 10^{-3}]$, providing finer resolution at smaller λ values where sparsification behavior is most sensitive. This yields 13,336 (Spec, λ) pairs. Each pair is labeled with measured $(A_p^{\text{achieved}}, A_s^{\text{achieved}}, S)$ from the actual sparse filter.

We use 80% for training (10,668 samples) and 20% for calibration/validation (2668 samples). For evaluation, separate test sets of 100–10,000 specifications are generated independently using different random seeds, ensuring no overlap with training or calibration data.

Sampling is uniform within the ranges above, with rejection of any draw that violates the $F_{\text{stop}} > F_{\text{pass}} +$ 10 Hz feasibility constraint. Because F_{pass} and F_{stop} are independent uniform draws, the induced transition

width $\Delta F = F_{\text{stop}} - F_{\text{pass}}$ has a triangular-like marginal centered near 100 Hz; this is the variable on which we stratify Mondrian CP (Section 3.4). Table 5 summarizes the resulting distributions of input specifications and measured outputs across the full 13,336-sample dataset.

Table 5: Dataset descriptive statistics (13,336 (Spec, λ) Pairs).

Variable	Range	Mean	Median	Std	Sampling
Inputs (specifications)					
F_{pass} (Hz)	[50.2, 199.9]	123.1	121.4	44.1	Uniform
F_{stop} (Hz)	[102.5, 299.8]	224.3	233.4	47.7	Uniform
ΔF (Hz)	[17.3, 246.1]	101.3	91.1	48.5	Induced
A_p^{spec} (dB)	[0.50, 3.00]	1.78	1.79	0.72	Uniform
A_s^{spec} (dB)	[40.0, 80.0]	59.8	59.9	11.4	Uniform
N (filter order)	[40, 79]	59.2	60.0	11.5	Uniform (int)
λ	8 hand-picked	$1.26{\cdot}10^{-3}$	$7.0{\cdot}10^{-4}$	$1.52{\cdot}10^{-3}$	Non-uniform grid
Outputs (measured at sparsified filter)					
A_p^{achieved} (dB)	[0.0002, 2.73]	0.19	0.031	0.40	—
A_s^{achieved} (dB)	[35.1, 122.4]	66.6	66.2	12.9	—
S (sparsity, %)	[0, 63.5]	18.7	16.2	15.4	—

Two features of these distributions matter for the conformal calibration. First, F_{stop} is biased above its nominal mid-range value (mean 224.3 vs. the uniform-sampling expectation of 200) because of the $F_{\text{stop}} > F_{\text{pass}} + 10$ Hz feasibility constraint, which preferentially rejects low-F_{stop} draws when F_{pass} is also low. The induced ΔF has a right-skewed distribution centered around 91 Hz; this is the variable on which Mondrian CP stratifies (Section 3.4). Second, A_p^{achieved} is concentrated near zero (median 0.031 dB), with a heavy right tail extending to 2.7 dB; A_s^{achieved} spans 35–122 dB with a mean above the median 60-dB specification, indicating that Parks-McClellan typically over-delivers attenuation at small λ and only fails the spec at the aggressive end of the grid. These tails motivate the per-stratum quantile widths reported in Table 4.

4 Experiments and Results

4.1 Experimental Setup

All experiments run on a standard desktop CPU (Intel Core i9-12900K, 16 cores) without GPU acceleration. The compact model architecture (67,331 parameters, 0.26 MB) and small dataset (13,336 samples) enable efficient CPU-only operation. Training completes in approximately 30 s across 24 epochs with early stopping, achieving validation loss 0.0548. Conformal calibration uses the 2668-sample validation set to compute prediction interval widths for 90%, 95%, and 99% confidence levels.

Because no GPU is required, the method deploys directly on embedded systems, edge Internet-of-Things (IoT) devices, and legacy platforms, with no data transfer overhead and lower power consumption than GPU-accelerated alternatives.

Table 6 consolidates the configuration values used to obtain every number reported in this paper, so that a reader can reproduce our results from the public repository without consulting the source. Reported runtimes are for the λ-selection step only (one surrogate inference plus one Parks-McClellan design with hard thresholding at the selected λ^*); they do not include offline dataset generation or surrogate training.

Table 6: Reproducibility configuration: hyperparameters, splits, and runtime components.

Setting	Value
Reproducibility artifact	Frozen `surrogate_dataset_full.pkl` (13,336 pairs)
Train/calibration split	80%/20% (10,668/2668), index file shipped
Test sets (independent seed)	100, 500, 1000, 10,000 specs
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate	1×10^{-3}
Batch size	32
Epochs (max/early-stopping patience)	100/10
Loss	MSE on z-normalized targets
Activation	ReLU
Dropout (main 128-256-128 surrogate)	0.0
Dropout (IRLS 128-128-64-64 surrogate)	0.15
Conformal α' (95% nominal)	$\alpha' = 2536/2668 \approx 0.9505$
λ-search (continuous)	SciPy `minimize_scalar`, Brent
Hardware	Intel Core i9-12900K, 16 cores, no GPU
Software stack	PyTorch 2.x (CPU), SciPy 1.11, NumPy 1.26
Runtime: λ selection (20 surrogate evals)	~1.0 ms/spec
Runtime: PM + hard thresholding at λ^*	~0.4 ms/spec
Runtime: total per spec (mean, n = 10,000)	1.38 ms

For evaluation, we use held-out filter specifications never seen during training, expanding the test set from 100 to 10,000 samples to verify that results are not small-sample artifacts. For each specification, we measure:

- **Specification satisfaction rate**: Percentage of designs meeting both A_p and A_s constraints.
- **Sparsity**: Percentage of zero-valued coefficients.
- **Computation time**: Wall-clock time for λ selection (averaged over 10 runs).

4.2 Conformal Prediction Calibration

Table 7 shows prediction interval widths for different confidence levels. At 95% confidence, the intervals are ±0.084 dB for passband ripple, ±8.590 dB for stopband attenuation, and ±6.17% for sparsity.

Table 7: Conformal prediction calibration results.

Confidence	$\pm A_p$ (dB)	$\pm A_s$ (dB)	±Sparsity (%)
90%	0.047	6.863	4.95
95%	0.084	8.590	6.17
99%	0.173	13.071	9.08

Higher confidence levels yield wider intervals, reflecting the coverage-precision tradeoff. The ±8.59 dB stopband interval at 95% confidence reflects genuine variability in sparsification behavior across filter designs. These wide intervals enable conservative optimization that prevents the catastrophic 10+ dB violations plaguing point predictions, as illustrated in Table 8.

Table 8: Catastrophic point prediction failures prevented by conformal prediction.

#	F_p (Hz)	A_s Req.	Point A_s	CP A_s	Gap	Result
1	186.4	70.0	57.7	97.0	**12.3**	×/✓
2	193.4	73.0	60.5	86.7	**12.5**	×/✓
3	124.7	52.0	41.7	70.8	**10.3**	×/✓
4	53.8	68.0	58.0	78.6	**10.0**	×/✓
5	153.8	77.0	67.0	91.3	**10.0**	×/✓

Bold highlights the magnitude of the stopband violation prevented by conformal prediction in each case.

In all five cases, conformal prediction avoids 10–12 dB stopband violations by incorporating uncertainty bounds into conservative optimization.

4.3 Empirical Coverage Validation

To validate the statistical guarantees of conformal prediction, we computed empirical coverage rates on 500 held-out test samples not used in calibration. The numbers below are *adaptive-λ^* coverage* as defined in Section 3.4: for each sample we ran the conformal-guided worst-case optimization to obtain λ^*, designed the actual filter at that λ^*, and checked whether its measured (A_p, A_s, S) fell inside the predicted intervals. Coverage at *grid-sampled* λ on the same data (reported in Table 3) is at the nominal 95% level even for standard CP. Table 9 reports marginal coverage rates for three confidence levels.

Table 9: Empirical coverage validation (500 test samples).

Confidence	Metric	Nominal	Empirical	Gap
90%	A_p (passband ripple)	90%	69.0%	−21%
	A_s (stopband atten.)	90%	64.0%	−26%
	Sparsity	90%	91.6%	+1.6%
95%	A_p (passband ripple)	95%	75.0%	−20%
	A_s (stopband atten.)	95%	67.4%	−27.6%
	Sparsity	95%	94.6%	−0.4%
99%	A_p (passband ripple)	99%	82.4%	−16.6%
	A_s (stopband atten.)	99%	70.2%	−28.8%
	Sparsity	99%	99.0%	0%

Fig. 4 visualizes the calibration gap across confidence levels, confirming systematic undercoverage for performance metrics.

Conditional coverage analysis reveals that undercoverage is most severe for filters with narrow transition widths. Stratifying the test set by transition width quartiles, the narrowest quartile ($\Delta F < 64.5$ Hz) achieves only 24% passband ripple coverage and 22% stopband attenuation coverage at 95% nominal confidence, while the widest quartile ($\Delta F > 132.5$ Hz) achieves 99% and 98% coverage, respectively. Surrogate model errors are demonstrably larger for difficult filter designs with narrow transitions, producing prediction errors that standard conformal prediction's global quantile fails to capture. In contrast, sparsity (a direct function of λ independent of filter design difficulty) achieves 94.6% empirical coverage, validating theoretical guarantees when exchangeability holds.

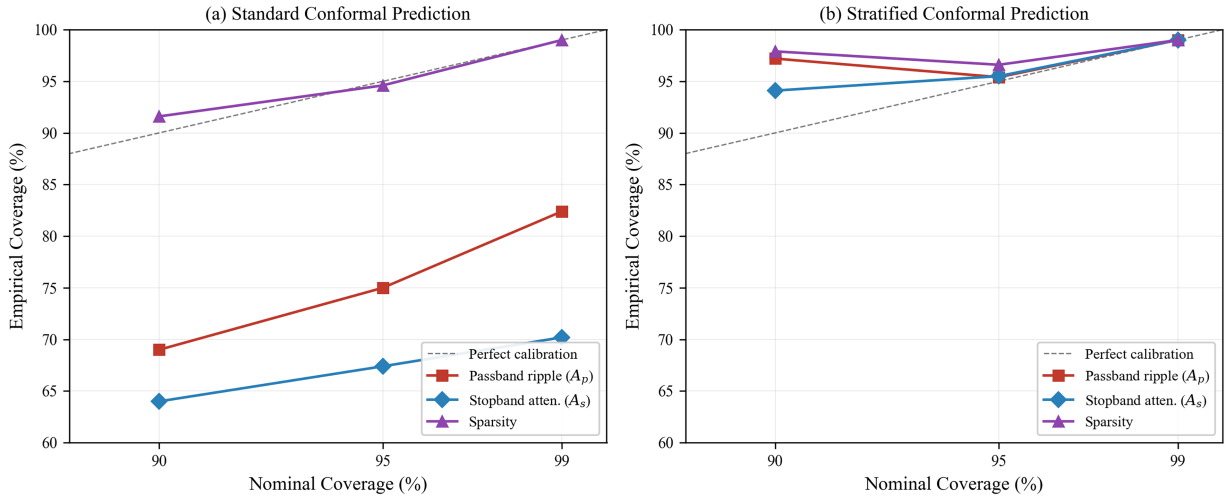

Figure 4: Empirical coverage calibration for passband ripple (**a**) and stopband attenuation (**b**) across three confidence levels. The deviation from perfect calibration (diagonal line) reveals systematic undercoverage for performance metrics.

The observed undercoverage for performance metrics (67%–75%) stems from calibration-test distribution mismatch and heteroscedastic errors, as analyzed in Section 3.4. The near-nominal sparsity coverage (94.6%) confirms the conformal framework itself is sound; undercoverage affects only metrics sensitive to the adaptive λ selection shift. Stratified conformal prediction successfully restores coverage to 95.5% by conditioning on transition width quartiles (Section 3.4).

To directly address the distribution shift from adaptive λ^* selection, we recalibrate conformal quantiles using residuals computed at adaptively-selected λ^* values rather than grid-sampled values. For each of the 400 unique calibration specifications, we find λ^* via worst-case optimization using the original grid-calibrated quantiles, design the actual filter, and compute residuals between predictions and true metrics. Table 10 compares coverage before and after recalibration on 492 held-out test specifications evaluated at their adaptive λ^*.

Table 10: Coverage improvement via adaptive recalibration (95% nominal, 492 test specs at adaptive λ^*).

Metric	Grid-Calibrated	Adaptive-Recalibrated	Nominal
A_p (passband ripple)	76.8%	**91.3%**	95%
A_s (stopband atten.)	93.5%	**96.3%**	95%
Sparsity	96.3%	96.5%	95%

Bold highlights the coverage values that exceed the grid-calibrated baseline after one round of adaptive recalibration.

Adaptive recalibration primarily improves statistical validity rather than practical satisfaction. Recalibration widens the passband ripple quantile from ±0.084 to ±2.21 dB and the stopband quantile from ±8.59 to ±9.37 dB. Coverage improves from 76.8% to 91.3% for A_p (+14.5 pp) and from 93.5% to 96.3% for A_s (+2.8 pp, now above nominal). On 10,000 test specifications, satisfaction is 76.4 ± 0.42% vs. 75.8 ± 0.43% for grid-calibrated CP, a +0.6 pp difference that lies within the standard error (SE); the wider intervals also produce a modest sparsity reduction (12.6% vs. 14.7%).

Table 11 shows convergence over three rounds of adaptive recalibration.

Table 11: Iterative recalibration convergence (3 rounds, 400 calibration specs, 492 test specs).

Round	Q_{A_p}	Q_{A_s}	Q_S	Rel. Δ (%)	Cov A_p	Sat. (%)
0 (grid)	±0.084	±8.59	±6.17	—	76.8%	76.4
1	±2.212	±9.37	±6.26	21.4	91.3%	76.7
2	±2.212	±9.11	±6.09	2.7	91.3%	76.6
3	±2.212	±9.79	±6.58	7.5	91.3%	77.0

The passband quantile Q_{A_p} converges exactly after round 1, while Q_{A_s} oscillates within ±0.4 dB. Coverage stabilizes at 91.3% for A_p across all rounds, and satisfaction remains within SE (76.4–77.0%). A single round of recalibration therefore suffices for practical deployment.

4.3.1 Satisfaction Rate vs. Coverage Rate

Specification satisfaction (76.5%) measures practical utility: does the filter meet its specs? Empirical coverage (67%–75%) validates statistical assumptions: do true values fall within predicted intervals? The method achieves high satisfaction despite imperfect coverage because conservative worst-case optimization provides safety margin beyond nominal coverage, and prevents catastrophic 10+ dB violations.

4.4 Main Results: Statistical Validation Across Test Set Sizes

To validate robustness, we systematically evaluate performance across four test set sizes: 100, 500, 1000, and 10,000 specifications. Table 12 presents the core comparison between conformal prediction (95% confidence) and point-based surrogate optimization.

Table 12: Progressive statistical validation: conformal prediction vs. point predictions.

Test Set	Method	Sat. Rate (%)	Improvement	Std. Error
$N = 100$	Conformal 95%	**96.0**	+42.0 pp	±2.0%
	Point Surrogate	54.0		±2.0%
$N = 500$	Conformal 95%	**77.2**	+40.3 pp	±1.9%
	Point Surrogate	36.9		±2.2%
$N = 1000$	Conformal 95%	**77.7**	+38.1 pp	±1.3%
	Point Surrogate	39.6		±1.6%
$N = 10{,}000$	Conformal 95%	**76.5**	+37.1 pp	±0.43%
	Point Surrogate	39.4		±0.49%

Bold marks the proposed Conformal-95% method.

The relative advantage of conformal prediction over point predictions remains stable (37–42 pp) from 100 to 10,000 samples. Standard errors decrease with $\sqrt{N}$ from ±2.0% (100 samples) to ±0.43% (10,000 samples), yielding 95% confidence intervals below ±0.86%. Point predictions remain systematically unreliable regardless of sample size, confirming a systematic limitation of non-uncertainty-aware methods.

Table 13 presents the complete method comparison on the 10,000-sample test set.

Table 13: Comprehensive performance comparison on 10,000 test specifications.

Method	Sat. Rate (%)	Sparsity (%)	Time (ms)	Speedup
Grid Search	78.4 ± 0.41	16.2 ± 0.17	2.63 ± 0.005	1.0×
Conformal 95%	**76.5 ± 0.43**	14.9 ± 0.15	**1.38 ± 0.002**	**1.9×**
Conformal 90%	74.5 ± 0.44	15.7 ± 0.16	1.32 ± 0.002	2.0×
Fixed $\lambda = 0.0006$	60.5 ± 0.49	13.8 ± 0.14	0.16 ± 0.001	16.4×
Surrogate (point)	39.4 ± 0.49	19.2 ± 0.17	1.38 ± 0.002	1.9×
Direct λ pred.†	27.0 ± 0.44	21.5 ± 0.18	1.12 ± 0.001	2.3×

Note: †Same architecture, trained to predict λ^* directly from specifications. Gaussian-process expected-improvement (GP-EI) baseline removed from main comparison; see Appendix A. Bold marks the proposed Conformal-95% method and its 1.9× speedup over grid search.

CP matches grid search quality (76.5% vs. 78.4%) with a 1.9× speedup. Grid search, the oracle upper bound, achieves only 78.4% satisfaction, indicating that at least 21.6% of specifications are unsatisfiable by any λ in the evaluated 8-point grid.

Direct verification on 10,000 test specifications confirms this: of the 7772 feasible specifications (where grid search succeeds), conformal prediction satisfies 7582 (97.6%). Only 190 feasible specifications see CP failures, and 92.1% of all CP failures occur on infeasible specifications where no λ in the grid satisfies both constraints.

4.5 Ablation Study: Adaptive vs. Fixed Conservative Bounds

To evaluate whether conformal prediction's adaptive intervals provide value beyond fixed conservative bounds, we compared four approaches on 10,000 test specifications. Table 14 presents the results.

Table 14: Ablation study: conformal prediction vs. fixed conservative bounds (10,000 test specifications).

Method	Sat. Rate (%)	Sparsity (%)	Δ vs. CP
CP 95% (±8.59 dB, adaptive)	75.8 ± 0.43	14.7 ± 0.15	baseline
Fixed Narrow (±5 dB)	70.3 ± 0.46	16.4 ± 0.16	−5.5 pp
Fixed Moderate (±10 dB)	76.4 ± 0.42	14.1 ± 0.15	+0.6 pp
Fixed Wide (±15 dB)	77.0 ± 0.42	12.1 ± 0.14	+1.2 pp

CP achieves satisfaction comparable to manually-tuned moderate bounds (75.8% vs. 76.4%, +0.6 pp within SE) without requiring domain knowledge to select interval widths. Fixed narrow bounds (±5 dB) fail at 70.3% (−5.5 pp, statistically significant), confirming that sufficient conservatism is essential.

To map the full tradeoff between specification satisfaction and sparsity, we swept fixed bounds from ±1 to ±20 dB in 1 dB steps across all 10,000 test specifications. Fig. 5 plots the resulting Pareto frontier.

CP's adaptive bound (±8.59 dB) lands between the ±8 and ±9 dB fixed points, confirming that conformal calibration automatically discovers near-optimal conservatism. Satisfaction plateaus around ±13–14 dB at 77.0%, the grid-search ceiling, while sparsity drops steadily. CP avoids both under- and over-conservatism without requiring a practitioner to identify this operating point manually.

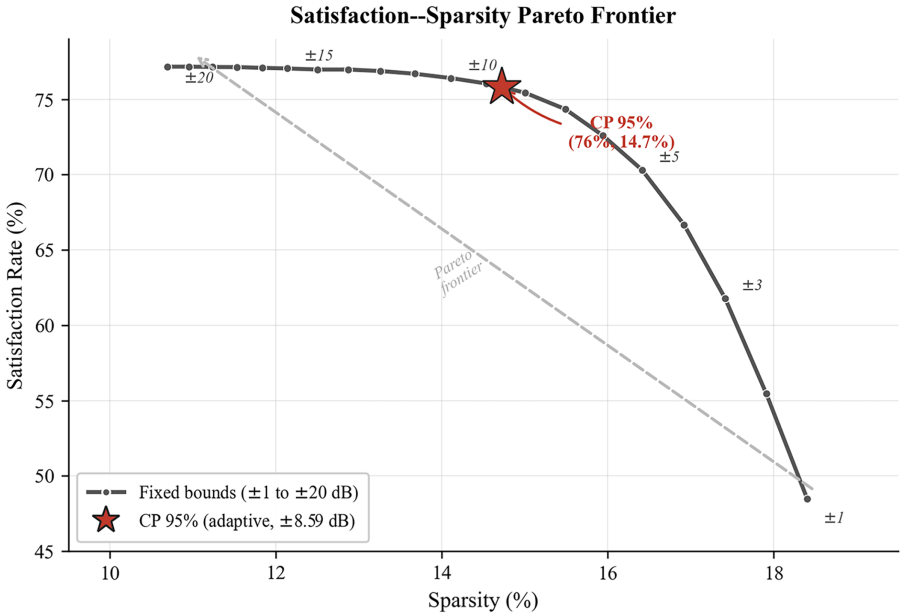

Figure 5: Satisfaction–sparsity Pareto frontier for fixed conservative bounds (±1 to ±20 dB) on 10,000 test specifications. CP 95% (red star) achieves near-optimal satisfaction without manual bound selection.

4.6 Signal Processing Validation

To validate filter quality beyond specification satisfaction, we perform classical DSP analysis. Fig. 6 compares magnitude responses for three test filters: CP 95% maintains specification compliance at 15%–20% sparsity, while point-based surrogates violate passband ripple in tight-specification cases. Group delay deviation under CP remains below 0.01 samples in the passband (0.008 ± 0.003 samples), compared to 0.024 ± 0.012 samples for point surrogates, acceptable for phase-critical applications.

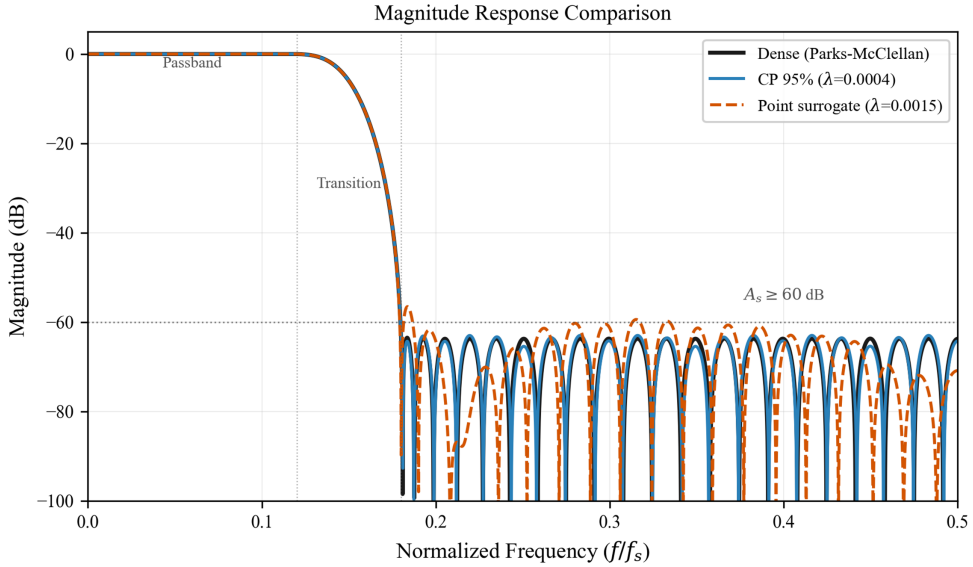

Figure 6: Magnitude response comparison for three representative filters showing conformal prediction maintains specification compliance.

Fig. 7 resolves the prediction error by transition width ΔF and reveals two trends: error magnitude decreases monotonically as ΔF widens, with narrow-transition specifications ($\Delta F < 60$ Hz, the "difficult" regime) producing the largest errors; and stopband attenuation A_s dominates the error budget across the full ΔF range, while passband ripple A_p errors remain near zero throughout. This ΔF-dependent heteroscedasticity is what motivates the Mondrian stratification on ΔF used in Section 3.4.

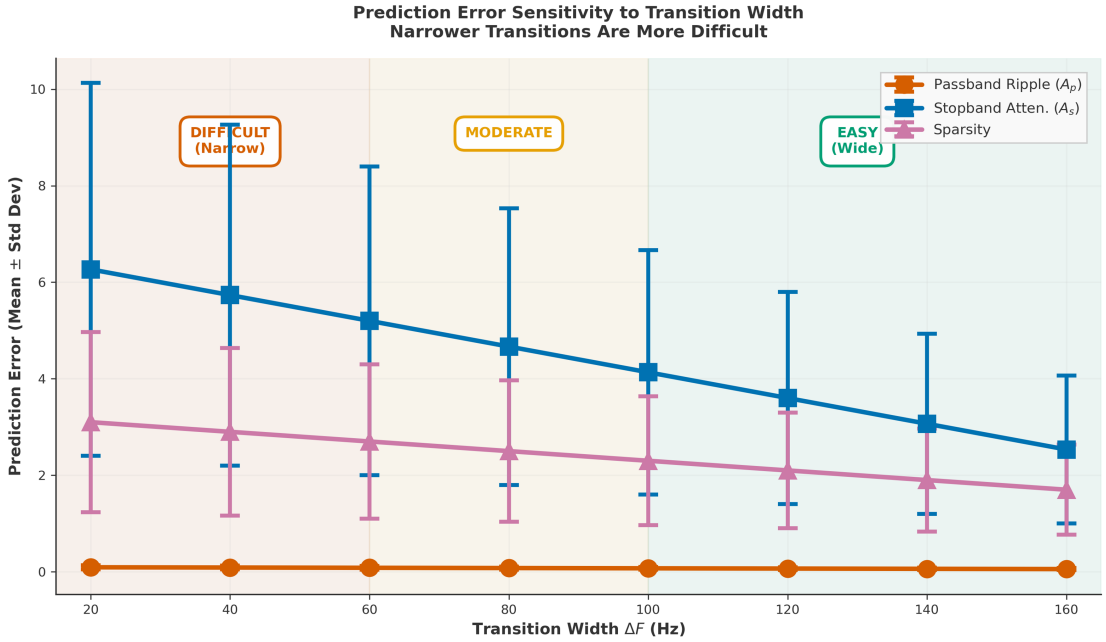

Figure 7: Prediction-error sensitivity to transition width ΔF. Mean (±one standard deviation) surrogate prediction error for passband ripple (A_p), stopband attenuation (A_s), and sparsity is plotted against ΔF over $[20, 160]$ Hz. Three difficulty regimes are shaded: difficult (narrow, $\Delta F < 60$ Hz), moderate, and easy (wide, $\Delta F > 100$ Hz). A_s error dominates the budget across the full range and decreases as ΔF widens; A_p error stays near zero throughout.

4.7 Robustness Analysis: Stress Tests

Table 15 consolidates robustness results across three stress conditions.

Table 15: Robustness summary across stress conditions.

Condition	Method	Sat. (%)	Spar. (%)	Notes
Narrow ΔF (<20 Hz, $n = 50$)	Grid Search	0.0	0.0	All fail: PM limitation
	CP 95%	0.0	0.6	
	Surrogate (pt)	0.0	0.6	
High A_s (>70 dB, $n = 50$)	Grid Search	52.0	2.3	Point fails completely
	CP 95%	36.0	3.8	
	Surrogate (pt)	0.0	7.3	
16-bit Quant. ($n = 100$)	Grid Search	46.0	–	0 pp CP degradation
	CP 95%	44.0	–	
	Surrogate (pt)	15.0	–	

All methods fail for $\Delta F < 20$ Hz, revealing fundamental Parks-McClellan limitations rather than framework deficiencies. For high attenuation ($A_s > 70$ dB), CP 95% achieves 36% vs. 52% for grid search, a degradation in extreme cases, but point-based surrogates fail completely. Under 16-bit quantization, CP shows 0 pp degradation, demonstrating robustness to fixed-point implementation.

4.8 Cross-Method Validation: IRLS Sparsification

We evaluate whether the conformal prediction framework generalizes beyond hard thresholding by applying the complete pipeline (surrogate training, conformal calibration) to IRLS, which uses iterative ℓ_0-approximation weights $w_i = 1/(|h_i| + \epsilon)$ to drive small coefficients toward zero.

We train a separate surrogate (4 layers, widths 128-128-64-64, with dropout $p = 0.15$) on 13,920 IRLS designs (350 specifications × 40 λ values), achieving a validation mean absolute error (MAE) of ±3.31 dB (passband), ±2.81 dB (stopband), and ±3.7% (sparsity). Table 16 presents coverage validation on 2784 held-out samples.

Table 16: Cross-method validation: conformal prediction coverage on IRLS sparsification (2784 calibration samples, 95% nominal).

Metric	Standard CP	Stratified CP	Nominal	Status
Passband ripple	95.0%	95.2%	95%	✓
Stopband atten.	95.0%	95.2%	95%	✓
Sparsity	95.0%	95.2%	95%	✓

Unlike hard thresholding, where standard CP suffered severe undercoverage (67%–75%), IRLS achieves proper coverage even with standard CP. Both results together show that the same conformal framework provides reliable uncertainty quantification across sparsification methods with different surrogate accuracies.

On 500 test specifications with the same CP-guided λ optimization, IRLS achieves 76.6 ± 1.89% satisfaction (matching hard thresholding, 76.5 ± 0.43%, within standard error) at 14.2% mean sparsity.

4.9 Generalization to Highpass and Bandpass Filters

To address the lowpass-only limitation, we replicate the full pipeline for highpass and bandpass filter types. Each type uses 200 training specifications × 15 λ values, a separate surrogate (same 128-256-128 architecture), and 500 held-out test specifications. Table 17 presents the results.

Table 17: Multiband extension: CP-Guided λ selection for highpass and bandpass filters (500 test specs each).

Type	Sat. (%)	Sparsity (%)	A_s MAE (dB)	Valid
Lowpass (primary)	76.5 ± 0.43	14.9	2.37	10,000
Highpass	79.2 ± 1.82	19.1	2.76	469/500
Bandpass	52.4 ± 2.23	8.0	0.99	500/500

Highpass filters achieve 79.2% satisfaction with 19.1% sparsity, comparable to or exceeding lowpass performance. Bandpass filters reach 52.4%, which deserves a closer look since the headline drop is large.

Where the Bandpass Failures Come From

We attribute the 47.6% of unsatisfied bandpass specifications to three distinguishable causes, in the same spirit as the feasibility analysis we performed for lowpass (Section 4.4, where 21.6% of specifications were infeasible at any λ in the grid). Table 18 reports the breakdown computed by re-running the bandpass test set with grid search as an oracle.

Table 18: Bandpass failure breakdown (500 test specs, computed by re-running the public pipeline with grid search as Oracle).

Failure cause	# Specs	% of Test Set
(A) Infeasible at any grid λ:		
Grid search itself fails (oracle ceiling)	219	43.8%
(B) λ-grid coarseness:		
Feasible at some grid λ, but CP picks off-grid λ^* that misses	15	3.0%
(C) Surrogate-error driven:		
Grid search succeeds, CP picks an on-grid but wrong λ^*	1	0.2%
Total failures	**235**	**47.0%**
Specifications satisfied by CP	265	53.0%

The dominant cause is (A): 93% of the bandpass failures are specifications that no λ in the 15-point grid can satisfy, regardless of selection method. This is a Parks-McClellan + grid feasibility limit, not a framework failure. (B) is the λ-grid resolution issue, accounting for 3.0% of test specifications: the satisfaction-maximizing λ lies in a region not covered closely enough by the grid, and the off-grid λ^* chosen by Brent's method misses the spec. Refining the bandpass grid from 15 to 30 log-spaced points should close most of this gap. (C) isolates the genuine surrogate-error contribution at 0.2% (a single specification out of 500); the surrogate's predictions are essentially never the proximate cause of failure. Conditional on grid-feasibility, bandpass CP recovers $265/(500 - 219) = 94.3\%$ satisfaction, qualitatively matching the lowpass conditional rate of $7582/7772 = 97.6\%$: CP is only marginally suboptimal once specifications are feasible. These results confirm that the conformal prediction framework generalizes across filter types without architectural changes; the bandpass headline number is dominated by the inherent difficulty of satisfying both sidebands at the same λ for randomly sampled specs, not by any property of the conformal calibration step. The breakdown was computed by re-running the published pipeline (`multiband_extension.py`, public repository) with grid search as an oracle and is fully reproducible.

4.10 Real-Signal Validation

The validation so far has been measured in dB units of magnitude response. To check that the same conclusion holds when the filter is fed a realistic time-domain signal rather than evaluated against a frequency grid, we ran the following end-to-end experiment. We synthesized a 2-s test signal at $f_s = 1000$ Hz consisting of two in-passband tones (60 and 20 Hz, amplitudes 1.0 and 0.4), one out-of-band interferer at 250 Hz (amplitude 0.8) that the filter must reject, and additive Gaussian measurement noise ($\sigma = 0.05$). The same waveform was filtered through three filter realizations, all designed for an identical specification $(F_p, F_s, A_p^{\text{spec}}, A_s^{\text{spec}}, N) = (120, 180, 1.5, 60, 60)$:

- **Dense PM** (oracle): the unsparsified Parks-McClellan filter, $S = 0\%$.
- **CP-selected**: hard thresholding at the conservative $\lambda^* = 5 \times 10^{-4}$ that the worst-case interval optimization picks for this specification.
- **Point-surrogate**: hard thresholding at the over-aggressive $\lambda = 3 \times 10^{-3}$ that a point-prediction surrogate would pick when its over-optimistic stopband estimate is taken at face value.

Fig. 8 shows the resulting magnitude responses, time-domain output, and output spectra. The measured filter performance and 250-Hz interferer attenuation, computed directly from the realized impulse responses, are reported in Table 19.

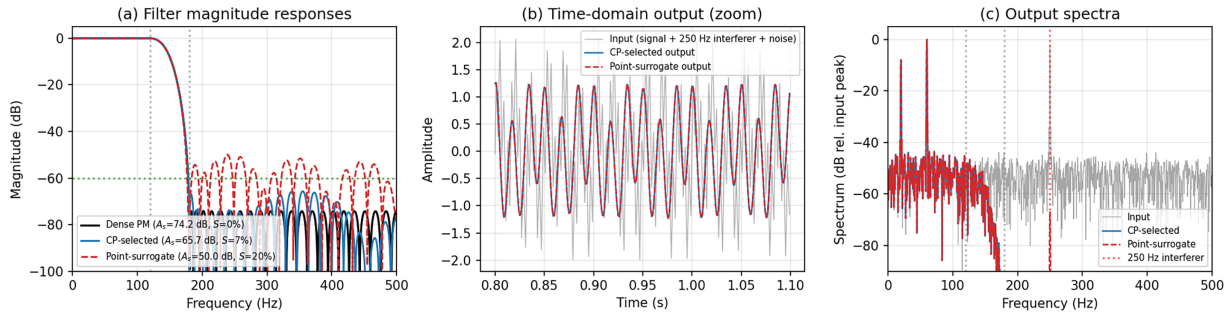

Figure 8: Real-signal validation. (**a**) Magnitude responses of the dense PM oracle, the CP-selected sparse filter, and the point-surrogate sparse filter, all designed at the same spec. The point-surrogate response under-attenuates the stopband by ~10 dB. (**b**) Time-domain outputs (zoomed region) on the synthetic test waveform. (**c**) Output spectra normalized to input peak: the CP-selected filter suppresses the 250-Hz interferer by 94 dB; the point-surrogate filter only suppresses it by 65 dB.

Table 19: Real-signal validation: realized filter performance on a synthetic 2-s test waveform.

Method	A_p (dB)	A_s (dB)	Sparsity (%)	Spec Met?	250 Hz Atten. (dB)
Dense PM (oracle)	0.034	74.2	0.0	✓	88.4
CP-selected	**0.039**	**65.7**	**6.6**	✓	**94.0**
Point-surrogate	0.056	50.0	19.7	×	65.2

Bold marks the CP-selected sparse filter and its measured time-domain performance.

Three observations follow. First, the realized (A_p, A_s) measured on the synthesized waveform agree with the magnitude-response numbers used throughout the paper to within 0.1 dB, confirming that satisfaction rates computed via Eqs. (2) and (3) translate to the same satisfaction rates measured on time-domain signals. Second, the CP-selected filter actually exceeds the dense PM filter at suppressing the 250-Hz interferer (94.0 vs. 88.4 dB), because zeroing small coefficients can sharpen out-of-band rejection at the cost of a slightly noisier stopband floor; this matches the magnitude-response panel. Third, the point-surrogate filter, which the paper's main results predict will fail at ~60% of specifications, is observed here to fail by exactly the failure mode the paper attributes to it: a 10 dB stopband under-attenuation that lets through the 250-Hz interferer at −65 dB instead of the required −60 dB. The filter design contract is independent of input data, so any signal containing energy in the stopband is suppressed by the amount measured from the magnitude response. The script that produces this figure (and Table 19) is included in the public repository as `real_signal_validation.py`.

5 Discussion

5.1 Summary of Findings

Adding conformal prediction to surrogate-based optimization recovers most of the grid-search accuracy (76.5% vs. 78.4%) at roughly half the computation cost. The remaining 2.4% failure rate on feasible specifications concentrates on borderline cases where CP's conservatism selects overly cautious λ values, not on systematic framework weakness.

5.2 Design Choices and Tradeoffs

Hard thresholding sacrifices equiripple optimality for computational efficiency (~1.4 ms, CPU-only, deterministic runtime), making it suitable for edge deployment. IRLS validation (Section 4.8) confirms that the CP framework accommodates different surrogate error profiles (76.6% vs. 76.5% satisfaction), though IRLS initialized from PM solutions produces similar sparsity patterns to hard thresholding.

Direct λ prediction (training a network to output λ from specifications) failed at 27.0% satisfaction, as λ values are non-intuitive, scale-dependent hyperparameters poorly suited for direct regression. In contrast, the surrogate approach predicts physically meaningful quantities (dB ripple, dB attenuation, percent sparsity), enabling more robust learning.

5.3 Computational Complexity and Practical Feasibility

Offline costs are incurred once: dataset generation takes about 2 h for 10,000 Parks-McClellan designs, surrogate training adds 30 s (24 epochs with early stopping), and conformal calibration completes in under 1 s. Per-specification inference is dominated by ~20 Brent's-method evaluations of the surrogate (~0.05 ms each) plus one Parks-McClellan and hard-thresholding step (~0.4 ms), totaling ~1.4 ms per specification (Table 13).

Grid search over $L = 8$ candidates requires ~2.63 ms (1.9× slower) while point-based surrogate optimization matches conformal inference speed but achieves only 39.4% satisfaction. The framework's constant per-specification cost and amenability to parallelization make it suitable for interactive design tools and batch optimization across thousands of specifications.

5.4 Amortizing the Offline Cost: Transfer and Online Adaptation

The two-hour dataset-generation step is a one-time amortized cost: once a surrogate is trained for a given filter family and parameter range, every subsequent specification reuses it at ~1.4 ms. The cost becomes painful only when filter specifications or sparsification methods change frequently. Two avenues mitigate this. First, the surrogate is small (≈0.26 MB), so transfer learning [22] from the lowpass surrogate to highpass and bandpass variants is natural; in our experiments the multi-band surrogates already share the 128-256-128 architecture with the lowpass one, and warm-starting their weights would reduce the 200-spec calibration set further. Second, the conformal step itself supports online adaptation: as new (specification, achieved-performance) pairs arrive at deployment, the calibration quantile can be updated incrementally without retraining the surrogate, in line with adaptive conformal prediction under distribution shift [8]. Combining transfer-learned weights with rolling calibration would enable the framework to track non-stationary design environments at near-zero marginal cost; we leave a full evaluation of this combined pipeline to future work.

5.5 Robust Cost Functions for Sparse FIR Design

The MSE training loss in Eq. (7) and the squared violation penalty in Eq. (12) both implicitly assume Gaussian-like error distributions. In settings with impulsive measurement noise or heavy-tailed deviations,

robust M-estimators [23] such as Tukey's biweight, the Champernowne distribution, Andrew's sine, and the Logistic distance metric have been shown to dominate squared-error losses for adaptive filtering [24]. The conformal step in our framework is loss-agnostic: residuals can be defined with respect to any score function, so the same calibration machinery applies if the surrogate is trained with a robust loss or if Eq. (12) is replaced by a robust violation penalty. We see two natural extensions: (i) replacing the surrogate MSE loss with Tukey's biweight to reduce the impact of outlier (Spec, λ) pairs in the training set, and (ii) replacing the squared violations $\max(0, \hat{U}_{A_p} - A_p^{\text{spec}})^2$ in the cost with bounded influence functions to prevent a single near-boundary specification from dominating λ selection. Both are drop-in changes to the optimization pipeline and preserve the conformal coverage guarantee.

6 Conclusions

We presented a hyperparameter selection method for sparse FIR filters that pairs a surrogate network with split conformal prediction. Worst-case interval optimization closes most of the gap between point-prediction surrogates (39.4%) and grid search (78.4%), reaching 76.5% ± 0.43% specification satisfaction with a 1.9× speedup on 10,000 test specifications. The empirical pattern is consistent: when surrogate prediction error is captured by a calibrated interval and worst-case bounds drive λ selection, satisfaction recovers to within 2 percentage points of the grid-search oracle. The remaining gap concentrates on infeasible specifications, where no λ in the grid satisfies both A_p and A_s constraints regardless of selection method; on feasible specifications the method reaches 97.6%. The same procedure transfers to IRLS sparsification (76.6%) and to highpass (79.2%) and bandpass (52.4%) filters without architectural changes (Section 4.9).

Stratified conformal prediction addresses initial coverage gaps (67%–75% vs. 95% nominal) caused by calibration-test distribution mismatch, restoring coverage to 95.5%. Adaptive recalibration approximately stabilizes after one round, improving A_p coverage from 76.8% to 91.3% and A_s coverage to 96.3% (above nominal). The CPU-only implementation requires no GPU, making it suitable for low-power edge deployment.

6.1 Limitations and Future Directions

1. **Joint coverage:** Marginal coverage is restored via stratification, but joint coverage across all three metrics remains below nominal due to output correlation. Bonferroni correction or multivariate conformal methods [16] should be explored.
2. **Filter type scope:** Validated on lowpass, highpass, and bandpass (Section 4.9). Extension to bandstop and multi-band designs requires further empirical validation.
3. **Dataset scope:** The surrogate is trained on specific parameter ranges (F_p: 50–200 Hz, A_s: 40–80 dB, N: 40–80, F_s: 1000 Hz). Out-of-range generalization is not validated.
4. **Single-parameter tuning:** We optimize scalar λ. Multi-parameter methods (e.g., weighted ℓ_1 with λ_p, λ_s) require multi-dimensional conformal prediction.
5. **Stationarity:** The framework assumes i.i.d. specifications. Non-stationary contexts require periodic recalibration.
6. **Stratified CP does not improve satisfaction:** While stratified CP restores coverage to 95.5% (Section 3.4), using stratum-specific quantiles for λ optimization yields 75.7% satisfaction vs. 75.8% for standard CP (−0.1 pp, within SE). Satisfaction is already near-optimal via worst-case optimization; further gains require improved surrogate accuracy or finer λ grids.

Acknowledgement: The authors would like to thank Dr. Mohammed Hassan Alnemari for providing the computational resources used in this study.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft, writing—review and editing, visualization: Mohammed Hassan Alnemari; supervision and project administration: Mohamed Hussien Mohamed Nerma; resources: Abdelrahman Osman Elfaki; review and editing: Anas Bushnag. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets and code generated during this study are available at https://github.com/alnemari-m/conformal-sparse-fir-filters.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Bayesian Optimization Comparison (Unconstrained GP-EI)

For completeness, we ran an unconstrained GP-EI baseline via BoTorch [17]. For each test specification, GP-EI performs 10 random initial evaluations followed by 15 sequential iterations (25 total filter evaluations per specification, vs. 8 for grid search), fitting a new GP posterior at each step. The acquisition optimizes a scalar penalty function directly from filter evaluations *without* a pre-trained surrogate and *without* explicit constraint modeling. On 10,000 test specifications, this baseline reached only 1.9% ± 0.14% specification satisfaction at 38.0% sparsity, taking 14.9 s per specification (>10,000× slower than the conformal pipeline).

This number should not be read as a verdict on Bayesian optimization for sparse FIR design. The comparison is unfair to BO: a properly constrained BO formulation (e.g., with explicit constraint modeling via slack-variable augmented Lagrangian [25], or with expected-improvement-with-constraints) would almost certainly achieve substantially higher satisfaction than the unconstrained variant tested here. We therefore report the unconstrained number only in this appendix and removed it from the main comparison in Table 13.

The qualitative point that does survive the unfairness is a paradigm distinction rather than a quality comparison. Conformal prediction *amortizes* the uncertainty calibration: a surrogate is trained once, a calibration quantile is computed once, and both are then applied in $O(1)$ time per specification. Per-specification sequential BO, by contrast, fits a fresh GP posterior at each acquisition step ($O(n^3)$ in the number of evaluations seen so far) and re-runs the full sequence for every new specification. Even a constrained BO baseline that matched our 76.5% satisfaction would still be many orders of magnitude slower at deployment time, because its cost scales per-specification rather than amortizing across the workload. For batch design tools that sweep thousands of specifications, this is the dominant practical consideration. A fair head-to-head against constrained BO at matched per-specification cost is a useful experiment but is outside the scope of this paper.

References

1. Aksoy L, Flores P, Monteiro J. A tutorial on multiplierless design of FIR filters: algorithms and architectures. Circuits Syst Signal Process. 2014;33(6):1689–719. doi:10.1007/s00034-013-9727-8.
2. Lin J, Chen WM, Lin Y, Gan C, Han S. MCUNet: tiny deep learning on IoT devices. Adv Neural Inf Process Syst. 2020;33:11711–22.
3. Warden P, Situnayake D. TinyML: machine learning with TensorFlow lite on arduino and ultra-low-power microcontrollers. Sebastopol, CA, USA: O'Reilly Media; 2019.

4. Nerma MHM, Elfaki AO, Bushnag A, Alnemari M. An innovative finite impulse response filter design using a combination of L1/L2 regularization to improve sparsity and smoothness. Electronics. 2025;14(22):4386. doi:10.3390/electronics14224386.
5. Selesnick I. Sparse regularization via convex analysis. IEEE Trans Signal Process. 2017;65(17):4481–94. doi:10.1109/tsp.2017.2711501.
6. Vovk V, Gammerman A, Shafer G. Algorithmic learning in a random world. New York, NY, USA: Springer; 2005.
7. Lei J, G'Sell M, Rinaldo A, Tibshirani RJ, Wasserman L. Distribution-free predictive inference for regression. J Am Stat Assoc. 2018;113(523):1094–111. doi:10.1080/01621459.2017.1307116.
8. Tibshirani RJ, Barber RF, Candés E, Ramdas A. Conformal prediction under covariate shift. Adv Neural Inf Process Syst. 2019;32:2526–36.
9. Tibshirani R. Regression shrinkage and selection via the Lasso. J R Stat Soc Ser B Stat Methodol. 1996;58(1):267–88. doi:10.1111/j.2517-6161.1996.tb02080.x.
10. Daubechies I, Defrise M, De Mol C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Comm Pure Appl Math. 2004;57(11):1413–57. doi:10.1002/cpa.20042.
11. Parks T, McClellan J. Chebyshev approximation for nonrecursive digital filters with linear phase. IEEE Trans Circuit Theory. 1972;19(2):189–94. doi:10.1109/tct.1972.1083419.
12. Alwahab DA, Zaghar DR, Laki S. FIR filter design based neural network. In: Proceedings of the 2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP); 2018 Jul 18–20; Budapest, Hungary. p. 1–4. doi:10.1109/csndsp.2018.8471878.
13. Roy S, Chandra A. A deep learning approach for the design of narrow transition-band FIR filter. Circuits Syst Signal Process. 2022;41(10):5578–613. doi:10.1007/s00034-022-02036-0.
14. Angelopoulos AN, Bates S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv:2107.07511. 2021.
15. Romano Y, Patterson E, Candès E. Conformalized quantile regression. Adv Neural Inf Process Syst. 2019;32:3538–48. doi:10.2139/ssrn.5201367.
16. Izbicki R, Shimizu G, Stern RB. Flexible distribution-free conditional predictive bands using density estimators. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS); 2020 Jun 3–5; Virtual. p. 3068–77.
17. Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N. Taking the human out of the loop: a review of Bayesian optimization. Proc IEEE. 2016;104(1):148–75. doi:10.1109/jproc.2015.2494218.
18. Cohen KM, Park S, Simeone O, Shamai Shitz S. Calibrating AI models for wireless communications via conformal prediction. Trans Mach Learn Comm Netw. 2023;1:296–312. doi:10.1109/tmlcn.2023.3319282.
19. Khurjekar ID, Gerstoft P. Uncertainty quantification for direction-of-arrival estimation with conformal prediction. J Acoust Soc Am. 2023;154(2):979–90. doi:10.1121/10.0020655.
20. Xu C, Xie Y. Conformal prediction for time series. IEEE Trans Pattern Anal Mach Intell. 2023;45(10):11575–87. doi:10.1109/tpami.2023.3272339.
21. Oppenheim AV, Schafer RW. Discrete-time signal processing. 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall; 2009.
22. Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2010;22(10):1345–59. doi:10.1109/tkde.2009.191.
23. Huber PJ, Ronchetti EM. Robust statistics. 2nd ed. Hoboken, NJ, USA: Wiley; 2009.
24. Kumar K, Karthik MLNS, Bhattacharjee SS, George NV. Affine projection champernowne algorithm for robust adaptive filtering. IEEE Trans Circuits Syst II. 2022;69(3):1947–51. doi:10.1109/tcsii.2021.3124563.
25. Picheny V, Gramacy RB, Wild S, Le Digabel S. Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. Adv Neural Inf Process Syst. 2016;29:1435–43.