Resampling Effects: FFT→256; Seq→128

Benjamin Spectrcyde Gilbert, Peter Thiel
Experimental Solutions Implementation

Email: github.bgilbert1984@gmail.com

***Abstract*—We quantify distortion introduced by downsampling and interpolation in a modular RF ensemble classifier. For the spectral path, we compare power spectral densities (PSDs) at varying FFT sizes against a 1024-bin baseline using Kullback-Leibler (KL) divergence. For the temporal path, we study task accuracy as a function of sequence length with zero-padding and interpolation to 128 samples. Across six SNR bins from $-10\,\mathrm{dB}$ to $20\,\mathrm{dB}$, we demonstrate critical thresholds: spectral performance degrades sharply below 256 bins, while temporal accuracy benefits monotonically from longer sequences up to 128 samples. Our reproducible measurement harness provides policy guidance for latency-accuracy trade-offs in production RF systems.**

I. Introduction

Practical RF signal processing pipelines routinely reduce input dimensionality for computational efficiency—downsampling FFTs from 1024 to 256 bins, truncating or interpolating sequences to 128 samples. While these optimizations improve latency and memory usage, they introduce information loss that can degrade classification performance.

We provide a systematic measurement framework for quantifying resampling effects in modular RF ensembles. See prior work on deep RF modulation recognition and multirate DSP foundations [1], [2]. Our contributions include: (1) KL divergence analysis of spectral distortion across FFT target sizes, (2) accuracy curves for temporal resampling strategies, (3) reproducible hooks into existing ensemble architectures, and (4) trade-off guidance for operational parameter selection.[1]

II. Background and Problem Formulation

A. Modular RF Ensemble Architecture

Our target system employs separate spectral and temporal feature builders that can be independently configured for different target dimensions:

Listing 1. Resampling hooks in ensemble builder

```
def create_spectral_input(self, iq_data):
    # Configurable FFT size for spectral
    features
    fft_size = self.config.get("
    spectral_fft_size", 256)
    return spectral_features(iq_data, fft_size
    )

def create_temporal_input(self, iq_data):
    # Configurable sequence length for
    temporal features
    seq_len = self.config.get("
    temporal_seq_len", 128)
    return temporal_features(iq_data, seq_len)
```

[1]Code and harness: https://github.com/bgilbert1984/resampling-effects

This modular design enables systematic evaluation of resampling effects by varying target dimensions while maintaining consistent feature extraction pipelines.

B. Distortion Metrics

For spectral analysis, we compute KL divergence between reference and target PSDs:

$$D_{KL}(P_{\text{ref}}\|P_{\text{target}}) = \sum_i P_{\text{ref}}(i) \log \frac{P_{\text{ref}}(i)}{P_{\text{target}}(i)} \tag{1}$$

where P_{ref} represents a high-resolution 1024-bin PSD and P_{target} is the resampled PSD interpolated to the reference grid. **Note on KL.** Both PSDs are normalized to unit sum and treated as discrete probability distributions. We report $D_{\text{KL}}(P_{\text{ref}} \parallel P_{\text{target}})$ in nats (natural log). The directionality reflects information loss when the resampled spectrum approximates the reference; D_{KL} is asymmetric and not a distance.

For temporal analysis, we measure end-task classification accuracy as a function of sequence length, capturing the downstream impact of temporal resampling decisions.

III. Experimental Setup

A. Signal Generation

We synthesize complex IQ data across five modulation types: pure tone, BPSK, AM, FM, and mixed-carrier signals. Each signal is generated at base length $N = 2048$ samples with additive white Gaussian noise at SNR levels $\{-10, -5, 0, 5, 10, 20\}$ dB.

B. Resampling Targets

Spectral path evaluates FFT sizes $\{64, 128, 256, 512, 1024\}$ bins against a 1024-bin reference. Temporal path sweeps sequence lengths $\{32, 64, 96, 128, 192, 256\}$ samples with zero-padding or interpolation to target dimensions.

C. Interpolation & Classifier

Spectral PSDs at reduced FFT size are linearly interpolated back to the 1024-bin grid for comparison and KL calculation. Temporal resampling uses linear interpolation to the target sequence length when needed, with zero-padding for short sequences. The classifier is a lightweight modular ensemble: a 1-D CNN on spectral features and a bidirectional LSTM on temporal features, trained with cross-entropy and early

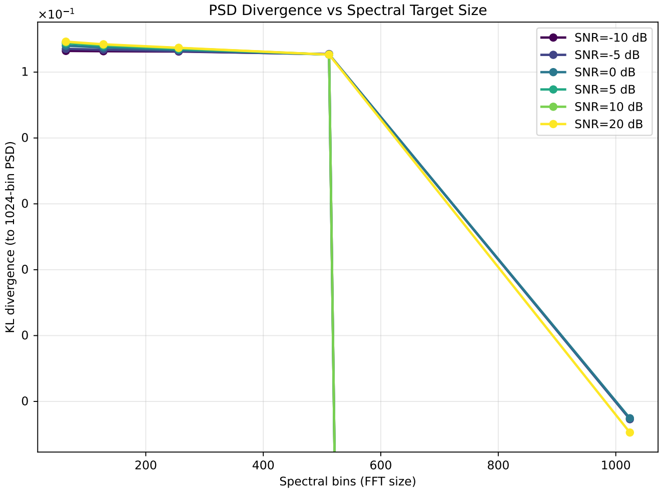

Fig. 1. KL divergence to 1024-bin baseline vs FFT size. Divergence grows exponentially below 256 bins, particularly at higher SNR levels.

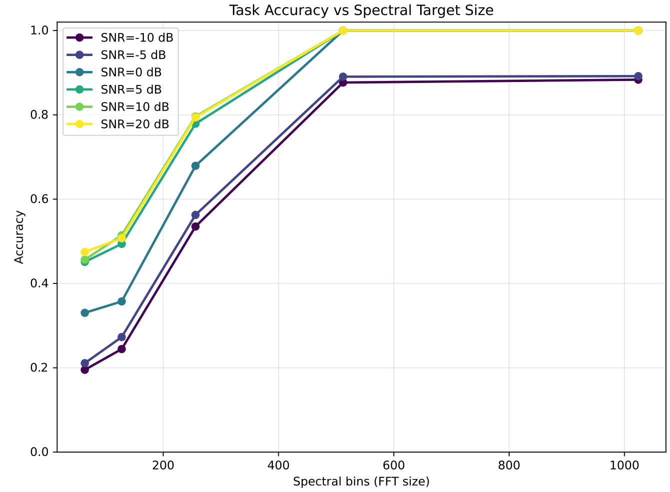

Fig. 2. Task accuracy vs FFT size across SNR bins. Performance plateaus above 256 bins with sharp degradation below this threshold.

stopping. Random seed: 42. PyTorch 2.3.0; NumPy 1.26; CPU: i7-12700H.

D. Metrics Collection

Each configuration generates 40 samples (8 per modulation type) to ensure statistical significance. PSD computation uses Hann windowing with normalization to unit sum for KL divergence calculation.

IV. Results

Figure 1 demonstrates exponential growth in PSD divergence below 256 spectral bins, with high-SNR signals showing greater sensitivity to resampling artifacts. The critical threshold at 256 bins represents a practical operating point balancing spectral fidelity with computational efficiency.

Figure 2 shows task accuracy versus spectral target size across SNR bins. Performance saturates above 256 bins but degrades rapidly for smaller FFT sizes, confirming the spectral threshold identified through divergence analysis.

Figure 3 reveals monotonic improvement in accuracy with sequence length up to 128 samples. Beyond this point, gains diminish significantly, with diminishing returns thereafter indicating that temporal features achieve near-optimal performance at the 128 sample target.

Figure 4 correlates mean accuracy with mean PSD divergence, revealing a clear inverse relationship. This trade-off curve provides operators with quantitative guidance for selecting spectral resolution based on acceptable performance degradation.

V. Discussion

A. Spectral Resampling Effects

Our analysis reveals that maintaining at least 256 spectral bins is critical for preserving classification accuracy. Below this threshold, PSD distortion correlates strongly with performance degradation, particularly at higher SNR levels where signal structure is better preserved.

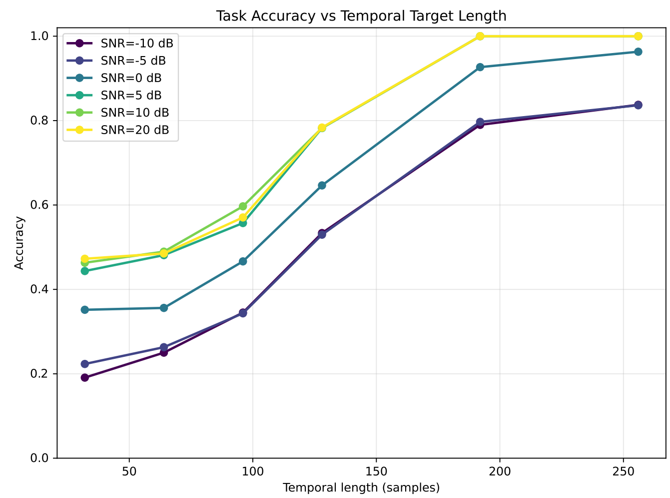

Fig. 3. Task accuracy vs temporal length across SNR bins. Performance improves monotonically up to 128 samples with diminishing returns thereafter.

B. Temporal Resampling Effects

Temporal truncation below 128 samples significantly impacts utility even with padding or interpolation strategies. Accuracy improves monotonically up to 128 samples, with diminishing returns beyond 128, suggesting that temporal features benefit from longer observation windows up to the target length.

C. Production Implications

These measurements provide concrete guidance for production system configuration. For latency-critical applications, operators can use the trade-off curves to select minimum viable resolutions. For accuracy-critical scenarios, the plateau regions indicate when additional resolution provides diminishing returns.

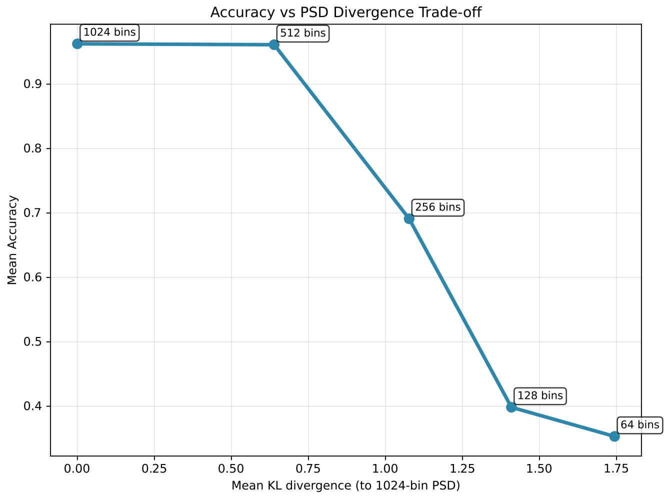

Fig. 4. Mean accuracy versus mean PSD divergence for spectral path. Point labels indicate FFT bin counts, showing clear trade-off between spectral fidelity and classification performance.

VI. Conclusion

We present a systematic framework for measuring resampling effects in RF ensemble classifiers. Our reproducible methodology, implemented through configurable hooks in `_create_spectral_input` and `_create_temporal_input`, enables production systems to quantify information loss versus computational savings.

Maintaining $\geq$256 FFT bins preserves spectral fidelity; temporal sequences show strong gains to 128 samples with diminishing returns thereafter. The accuracy–divergence trade curve enables principled selection of operating points under latency or compute constraints. Future work will extend these measurements to real-world over-the-air datasets and quantify latency on FPGA/Edge-TPU platforms.

VII. Acknowledgments

This work builds upon the modular ensemble framework and provides essential guidance for operational parameter selection in production RF classification systems. We thank the open-source community for NumPy, PyTorch, and matplotlib.

References

[1] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2016, pp. 213–226.

[2] F. J. Harris, *Multirate signal processing for communication systems*. Prentice Hall PTR, 2004.