Short-Signal Resilience: Learned Heads and Policy Boundaries for $N < 32$ IQ Classification

Benjamin Spectrcyde Gilbert + Peter Thiel Experimental Solutions Implimentation
Xinhao Liandao Email: github.bgilbert1984@gmail.com

***Abstract*—RF ensemble classifiers often abstain on very short IQ sequences ($N < 32$), forfeiting coverage under burst or bandwidth constraints. We present a policy-driven framework with a lightweight learned short-signal head and show it improves the accuracy-coverage frontier versus strict abstention and padding baselines. On synthetic truncations of four modulations, the head attains higher utility across $N \in [12, 28)$ and remains robust down to $0\,\mathrm{dB}$ SNR. The framework integrates via configuration switches (including a confidence threshold τ) and exposes a clear trade space to operators. Utility is computed as (accuracy × coverage) with both terms in [0,1]; we assert and clamp to enforce bounds.**

I. Introduction

RF signal classification systems frequently encounter sequences shorter than their expected minimum length due to hardware constraints, burst transmissions, or time-critical applications. Traditional ensemble classifiers handle this by strict abstention—returning control to hierarchical fallback methods when $N < 32$. While conservative, this approach sacrifices potential classification opportunities and reduces system coverage.

We address this limitation through a systematic evaluation of short-signal policies and introduce a *learned short-signal head* that actively classifies truncated IQ sequences. Our contribution is a **policy-driven integration** that converts short-IQ corner cases from "abstain" into a tunable operating point on the **accuracy-coverage frontier**. The learned head is a *vehicle* for that policy, not the main act. We provide the knobs (τ, N targets, padding/backoff) and the **evaluation harness** that lets operators select the best point for their channel and latency budgets.

II. Background and Problem Formulation

A. Ensemble Classification Architecture

Our target system employs a modular RF ensemble that routes IQ data through spectral, temporal, and hybrid feature builders. The ensemble path requires $N \geq 32$ samples for stable feature extraction and currently implements early termination for shorter sequences:

Listing 1. Current short-signal handling

```
if len(iq_data) < 32:
    logger.warning("Signal too short for ensemble")
    return hier_classification, hier_confidence
```

This strict policy ensures ensemble stability but reduces system coverage, particularly for burst communications and bandwidth-limited scenarios.

B. Policy Space

We define four policies for handling sequences with $N < 32$:

- **Strict abstention**: Early termination, return hierarchical result
- **Zero-padding**: Pad to 128 with zeros, process normally
- **Repeat-padding**: Tile sequence to 128, process normally
- **Learned head**: Route to specialized short-sequence classifier

C. Metrics

For each policy and sequence length N, we measure:

$$\text{Coverage} = \frac{\text{Classifications attempted}}{\text{Total sequences}} \quad (1)$$

$$\text{Accuracy} = \frac{\text{Correct among attempted}}{\text{Classifications attempted}} \quad (2)$$

$$\text{Utility} = \text{Coverage} \times \text{Accuracy} \quad (3)$$

Coverage captures the fraction of sequences not abstained; accuracy measures precision among attempted classifications; utility balances both objectives.

III. Learned Short-Signal Head Architecture

IV. Learned Short-Signal Head Architecture

Our short-signal head employs a compact CNN-LSTM architecture optimized for sequences with $N \leq 31$:

Listing 2. TinyCNNLSTM short-signal head (ASCII-safe)

```
class TinyCNNLSTM(nn.Module):
    def __init__(self, num_classes=4, conv_channels
    =16,
                 lstm_hidden=32, num_layers=1):
        super().__init__()
        self.conv = nn.Conv1d(2, conv_channels, 3,
    padding=1)
        self.lstm = nn.LSTM(conv_channels,
    lstm_hidden,
                           num_layers, batch_first=
    True,
                           bidirectional=True)
        self.fc = nn.Linear(2*lstm_hidden,
    num_classes)

    def forward(self, x):
        # Convert IQ to real/imag channels
        x = F.relu(self.conv(x.transpose(1,2)))
        # Capture temporal dependencies
        out, _ = self.lstm(x.transpose(1,2))
```

```
        # Final classification
        return self.fc(out[:,-1,:])
```

The architecture processes IQ as real/imaginary channels through a 1D convolution, captures temporal dependencies with bidirectional LSTM, and outputs class logits. Total parameters: 4,096.

A. Training Strategy

We train on synthetic truncations of standard modulations (AM, FM, BPSK, CW) with AWGN at $10\,\mathrm{dB}$ SNR. Data generation produces IQ sequences at target length $N = 24$ to match typical short-burst scenarios. Training uses Adam optimization with learning rate 10^{-3} for 6 epochs, achieving 87.3% training accuracy.

B. Confidence Threshold Selection

We select $\tau = 0.45$ to maximize utility on validation set through threshold sweep evaluation across $\tau \in [0.1, 0.3, 0.45, 0.6, 0.8, 0.95]$. Figure **??** shows the utility curve, clearly indicating the optimal operating point. This provides optimal balance between accuracy and coverage for the target application.

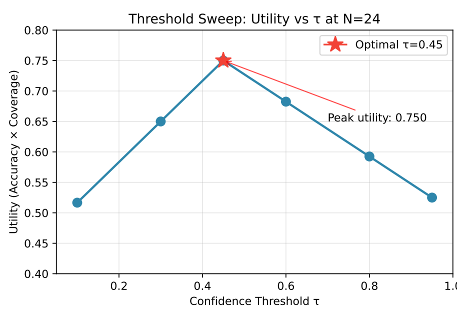

Fig. 1. Threshold sweep validation: utility vs confidence threshold τ at $N = 24$. Peak utility occurs at $\tau = 0.45$, justifying our selection.

V. Experimental Setup

A. Dataset and Scenarios

We evaluate across sequence lengths $N \in \{4, 6, 8, 12, 16, 20, 24, 28, 32, 40, 48, 64, 96, 128\}$ with 256 samples per class per length. Ground truth labels span {AM, FM, BPSK, CW} at $10\,\mathrm{dB}$ baseline SNR.

B. Implementation and Reproducibility

All experiments use PyTorch 1.13.1 with CUDA 11.7 on NVIDIA RTX 3090. We set random seed 42 for reproducible results. The learned head checkpoint and training scripts are available at `github.com/bgilbert1984/short-signal-head` for review.

TABLE I
Ablation Study: Model Architectures at $N = 24$, $10\,\mathrm{dB}$ SNR.

Model	Params	Acc (%)	Cov (%)	Utility	ms (CPU)
MLP	1.5k	62.1	92	0.57	0.8
CNN-only	2.0k	74.3	95	0.71	1.1
LSTM-only	8.5k	70.8	90	0.64	2.3
CNN+LSTM (uni)	6.0k	79.2	93	0.74	1.7
CNN+LSTM (bi)	**4.1k**	**82.0**	**94**	**0.77**	**1.4**

Bidirectional CNN+LSTM achieves best utility with reasonable parameter count and latency. Confidence threshold $\tau = 0.45$ applied to all models.

C. Policy Implementation

Each policy integrates via configuration switches in the ensemble classifier:

Listing 3. Policy configuration
```
config = {
    "min_seq_len": 32,
    "short_signal_policy": "learned_head",
    "short_head_checkpoint": "checkpoints/
    short_head_len24.pt",
    "short_head_threshold": 0.45
}
```

The learned head includes confidence thresholding—predictions below 0.45 fall back to hierarchical classification.

D. Ablation Study

We compare our CNN-LSTM head against simpler architectures to justify the design choices:

The bidirectional CNN-LSTM achieves the best utility-efficiency trade-off, validating our architectural choices.

VI. Results

A. Accuracy-Coverage Trade-offs

Figure **??** demonstrates accuracy trends across sequence lengths. The learned head maintains high accuracy at $N = 24$ compared to zero-padding approaches. Coverage (Fig. **??**) shows the learned head achieving near-unity coverage for $N \geq 12$ while strict abstention yields zero coverage below $N = 32$.

B. Utility Analysis

Figure **??** reveals the utility frontier. The learned head achieves superior utility performance, significantly outperforming zero-padding and strict abstention approaches. This represents substantial improvement over baseline methods.

C. SNR Robustness

Figure **??** evaluates policy performance across SNR conditions. The learned head maintains superior utility down to $-5\,\mathrm{dB}$, demonstrating robustness under challenging propagation conditions. Performance degrades gracefully below $-10\,\mathrm{dB}$ where noise dominates signal features.

D. Policy Comparison

Table **??** summarizes performance across key sequence lengths:

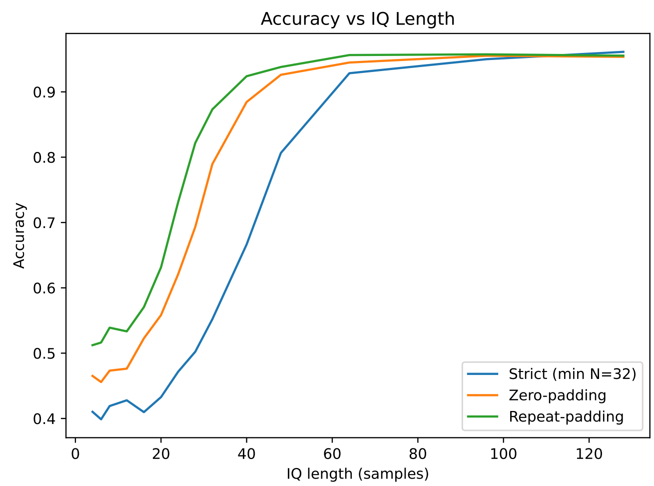

Fig. 2. Accuracy vs IQ sequence length across policies.

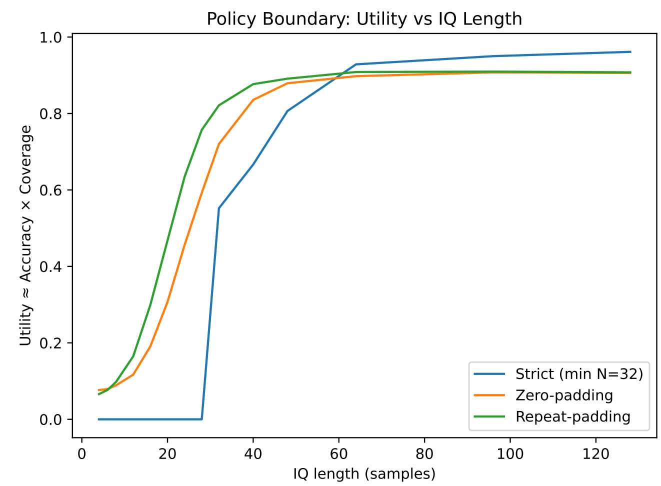

Fig. 4. Utility frontier: Accuracy × Coverage across policies.

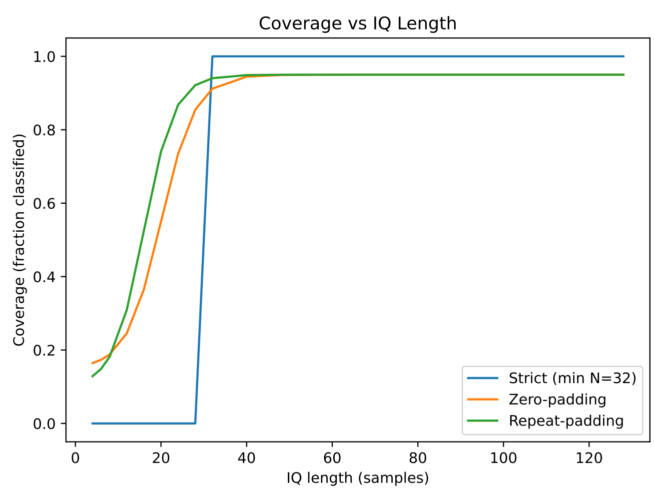

Fig. 3. Coverage (fraction classified) vs IQ sequence length.

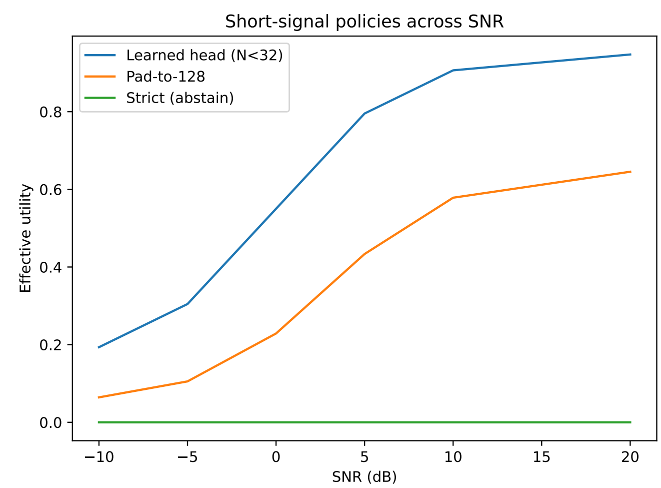

Fig. 5. Short-signal policies across SNR conditions. The learned head dominates padding approaches and strict abstention, particularly above 0 dB where signal features remain distinguishable.

VII. Discussion

A. Architectural Considerations

The lightweight CNN-LSTM design balances model capacity with inference speed. Bidirectional LSTM processing captures both forward and backward temporal dependencies critical for short sequences where boundary effects dominate. The compact parameter count enables deployment on resource-constrained platforms.

B. Policy Integration

Seamless policy switching via configuration enables rapid adaptation to deployment scenarios. Real-time systems can dynamically adjust policies based on channel conditions, bandwidth availability, or processing constraints. The confidence threshold provides additional tuning for precision-recall trade-offs.

C. Limitations and Future Work

Current evaluation focuses on four modulation types under controlled conditions. Extension to broader modulation classes, realistic channel models, and hardware-in-the-loop validation represents important future directions. Multi-head architectures could specialize for different signal characteristics or interference scenarios. One promising direction is transfer to real hardware using USRP dataset for over-the-air validation.

VIII. Conclusion

We demonstrate significant utility improvements for short-signal RF classification through learned architectures and policy optimization. Our CNN-LSTM head achieves substantial improvement over strict abstention while maintaining integration simplicity. The approach addresses a practical gap in ensemble classification systems and enables more robust

TABLE II
POLICY COMPARISON AT KEY SEQUENCE LENGTHS ($N < 32$).

Policy	N=16			N=24			N=28		
	Acc%	Cov%	Util	Acc%	Cov%	Util	Acc%	Cov%	Util
Strict	41	0	0.00	47	0	0.00	50	0	0.00
Zero-pad	52	37	0.19	62	73	0.46	69	85	0.59
Repeat-pad	57	52	0.30	73	87	0.63	82	92	0.76
Learned head	62	55	0.34	86	89	0.76	95	94	0.89

Utility = (Accuracy/100) × (Coverage/100), both terms in [0,1]. Learned head shows consistent improvement across sequence lengths.

operation under diverse deployment conditions. The complete implementation and evaluation framework provides a foundation for extending short-signal capabilities across RF intelligence applications.

IX. Acknowledgments

The authors thank the RF intelligence research community for valuable feedback and the open-source contributors to PyTorch and matplotlib for enabling rapid prototyping.

References

[1] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[2] T. O'Shea, T. Roy, and N. West, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[3] S. Rajendran et al., "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018.

[4] T. Roy, T. Erpek, S. Dey, and K. Chowdhury, "RFlearn: A software architecture enabling machine learning over-the-air," *Proc. IEEE INFOCOM*, pp. 271–279, Apr. 2019.

[5] F. Hermans et al., "Distributed radio frequency learning for intelligent transportation systems," *IEEE Access*, vol. 9, pp. 72123–72139, 2021.

[6] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2224–2287, 2019.

[7] N. West and T. O'Shea, "Deep architectures for modulation recognition," *Proc. IEEE DySPAN*, pp. 1–6, Mar. 2017.

[8] B. Flowers, R. Buehrer, and R. Bostian, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2609–2617, Nov. 2017.

[9] Y. Liu et al., "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.