

Hierarchical Automatic Modulation Classification with Conditional Cyclic-Cumulant Refinement

Ruilin Wu[✉], Xueyuan Liu[✉], and Philip H.W. Leong[✉]

Abstract—Cyclostationary features have shown robust cross-dataset generalization for single signals in flat channels, but higher-order feature extraction is costly. This paper proposes a hybrid hierarchical automatic modulation classification (AMC) algorithm that adapts computational effort to the discriminability of different modulation groups. A lightweight fast Fourier transform (FFT)–logistic regression (LR) classifier first separates the signals into BPSK, MSK, $\pi/4$ -DQPSK, 8PSK, and a QPSK-like family; for QPSK-like signals, blind symbol rate and carrier frequency offset (CFO) estimation, cyclic cumulant (CC) matrix construction, and convolutional neural network (CNN) classification are used to refine the classification into QPSK, 16QAM, 64QAM, and 256QAM. Experiments on the CSPB.ML.2018 and CSPB.ML.2022 datasets show that compared with the strip spectral correlation analyzer (SSCA)-based CC method, the proposed technique retains comparable same-dataset and cross-dataset accuracy while providing an estimated $7.20\times$ operation speedup and an approximately $9.52\times$ measured average CPU-latency speedup under the balanced eight-class setting. Ablation experiments show that direct eight-class classification using LR or a Multi-layer Perceptron (MLP) results in lower accuracy.

Index Terms—Automatic modulation classification, neural network, cyclic cumulants, cyclostationarity, fast Fourier transform.

I. INTRODUCTION

IN modern wireless systems, automatic modulation classification (AMC) identifies the modulation type of an unknown received signal before demodulation and is an important building block for cognitive radio, spectrum monitoring, and intelligent wireless receivers [1], [2]. Practical AMC is challenging because the received signal may be affected by noise, carrier frequency offset (CFO), symbol rate, pulse shaping, fading, and other channel impairments, which can differ substantially between training and deployment [3], [4].

Cyclostationary Signal Processing (CSP) exploits the cyclostationary structure introduced by modulation, sampling, pulse shaping, and other signal-generation processes [5], [6]. In particular, cycle frequency (CF) and higher-order cyclic cumulants (CCs) provide informative features that are less affected by noise and support modulation recognition under different signal conditions [7], [8].

Refs. [4], [9] combined blindly estimated CC features with a capsule network (CAP) and demonstrated strong classification and cross-dataset generalization on CSPB.ML.2018 and CSPB.ML.2022 [10]. Their front end performs band-of-interest detection [11], symbol-rate/CFO estimation, CF

search, and CC extraction for every signal. In particular, the strip spectral correlation analyzer (SSCA) [12] performs a two-dimensional spectral-correlation search for blind CF detection, introducing substantial computational cost.

Ref. [13] computes fourth-order cumulants for every input and uses them to separate modulation candidates hierarchically. Early-exit AMC skips deeper neural-network layers [14]. Our FFT–LR stage instead makes a low-cost decision first and skips blind parameter estimation, CC extraction, and convolutional neural network (CNN) inference for non-QPSK-like signals. Blind symbol-rate estimation can use cyclic-frequency peaks [15], while the fourth-power transform follows the M -th-power carrier-recovery principle [16]. Higher-order cyclic cumulants can support robust QAM identification [17]. The fast Fourier transforms (FFTs) of the squared and fourth-power signals capture these properties for low-cost coarse decisions, while CCs and the CNN are used only for QPSK-like signals. The key contributions are summarized as follows:

- We propose an efficient hybrid AMC implementation that first uses FFT–logistic regression (LR) and performs refinement using blind symbol-rate/CFO estimation and CC–CNN classification only for QPSK-like signals.
- We derive the branch-specific and expected operation costs. Compared to SSCA–CC–CAP [4], the $7.20\times$ speedup in the balanced eight-class setting combines $2.12\times$ from FFT substitution, $1.81\times$ from lightweight CNN refinement, and $1.87\times$ from conditional execution.
- Same-dataset and cross-dataset experiments show that the proposed method achieves accuracy comparable to the SSCA–CC–CAP [4]. Ablations further identify the roles of the FFT projections and separately evaluate the accuracy of blind symbol-rate and CFO estimation.

II. PROPOSED FFT-BASED CONFIGURATIONS

A. Algorithm Overview

As shown in Fig. 1, a shared front end performs band-of-interest detection [11], frequency shifting, and FFT feature extraction for two trained configurations for comparative purposes. The Direct Eight-Class FFT–LR, called the Direct Configuration, supplies the extracted features to an eight-class LR classifier. The Hierarchical FFT–LR/CC–CNN, called the Hierarchical Configuration, uses a five-class LR classifier that identifies BPSK, MSK, $\pi/4$ -DQPSK, 8PSK, and QPSK-like. For signals classified as QPSK-like, blind symbol-rate/CFO estimation, CC extraction, and a CNN are used to refine the classification to QPSK, 16QAM, 64QAM, and 256QAM.

Ruilin Wu, Xueyuan Liu, and Philip H.W. Leong are with The University of Sydney, Faculty of Engineering, School of Electrical and Computer Engineering, Australia (e-mail: ruilin.wu@sydney.edu.au; xueyuan.liu@sydney.edu.au; philip.leong@sydney.edu.au).

¹Code is available at https://github.com/ruilin-wu/Hierarchical_AMC.

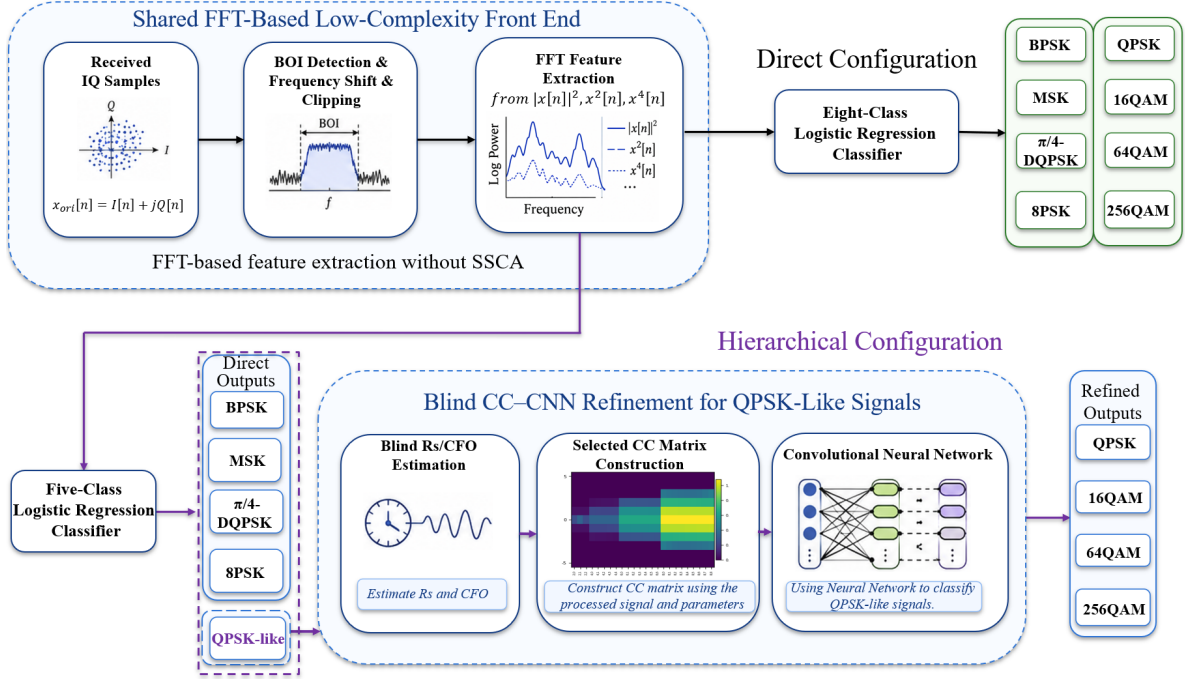


Fig. 1. Proposed direct and hierarchical AMC configurations sharing a low-complexity FFT-based front end.

B. FFT Projection Features

Let $x[n]$, $n \in \{0, 1, \dots, N-1\}$, denote a finite-length complex baseband signal preprocessed from the original signal $x_{\text{ori}}[n]$, where $N = 32,768$. Under the zero-delay configuration adopted in Ref. [4], the finite-length estimates of the non-conjugate ($\hat{Q}$) and conjugate ($\hat{R}$) second-order cyclic autocorrelation functions, evaluated on the discrete Fourier transform (DFT) grid $\alpha_k = k/N$, $k \in \{0, 1, \dots, N-1\}$, are

$$\begin{aligned}\hat{Q}_{\alpha_k}(x) &= \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2 e^{-j2\pi\alpha_k n} = \frac{1}{N} \text{FFT}\{|x[n]|^2\}[k], \\ \hat{R}_{\alpha_k}(x) &= \frac{1}{N} \sum_{n=0}^{N-1} x^2[n] e^{-j2\pi\alpha_k n} = \frac{1}{N} \text{FFT}\{x^2[n]\}[k].\end{aligned}\quad (1)$$

These features provide projections of the associated spectral correlation functions onto the CF axis while avoiding computation of the full SSCA. The zero-lag projection $R_x^\alpha(0) = \int S_x^\alpha(f) df$ integrates the spectral correlation function (SCF) over spectral frequency, so cancellation can make it differ from the full SSCA. The cyclic-moment framework follows Refs. [5], [6]. Assume one signal in a static flat channel, fixed carrier and timing, independent uniform symbols, a real bounded pulse with the required nonzero cyclic lines, circular Gaussian noise, a sufficiently long observation, and sampling without aliasing of these lines.

Thus, centered and offset second-order lines distinguish BPSK from MSK. When second-order lines vanish, centered and offset fourth-order lines distinguish QPSK from $\pi/4$ -DQPSK. The absence of both orders identifies 8PSK. Square QAM has the same peak-presence pattern as QPSK and therefore enters the QPSK-like family. This result establishes

class separation under the stated assumptions, rather than lossless equivalence to the full SSCA.

C. FFT-LR Classification Configurations

After band-of-interest (BOI) filtering and frequency shifting [11], three N -point FFTs from $\{|x[n]|^2\}$, $\{x^2[n]\}$, and $\{x^4[n]\}$ are computed. Algorithm 1 in Sec. VIII in Information for Reproducibility specifies the 32-feature construction, including FFT normalization, bin selection and spacing, feature definitions and ordering, symmetry and energy statistics, fixed thresholds, and insufficient-bin handling. After standardization, the feature vector $\mathbf{z} \in \mathbb{R}^{32}$ is supplied to a multinomial LR classifier:

$$p(c | \mathbf{z}) = \frac{\exp(\mathbf{w}_c^T \mathbf{z} + b_c)}{\sum_{d \in \mathcal{C}_m} \exp(\mathbf{w}_d^T \mathbf{z} + b_d)}, \quad \hat{c} = \arg \max_{c \in \mathcal{C}_m} p(c | \mathbf{z}). \quad (2)$$

In the Hierarchical Configuration, the LR classifier uses $m = 5$ and $\mathcal{C}_5 = \{\text{BPSK}, \text{MSK}, \pi/4\text{-DQPSK}, \text{8PSK}, \text{QPSK-like}\}$. The Direct Configuration uses $m = 8$ and $\mathcal{C}_8 = \{\text{BPSK}, \text{MSK}, \pi/4\text{-DQPSK}, \text{8PSK}, \text{QPSK}, \text{16QAM}, \text{64QAM}, \text{256QAM}\}$. In both cases, $\hat{c}$ is the modulation decision of the LR classifier, c and d denote class indices, $\mathbf{w}_c$ and b_c are the corresponding weight vector and bias, and $p(c | \mathbf{z})$ is the predicted probability.

D. CC-CNN Refinement Branch

For QPSK-like signals, symbol rate (R_s) is estimated from the spectral peaks of $\{|x[n]|^2\}$ and $\{x^4[n]\}$, while the residual CFO is estimated from $\{x^4[n]\}$. Algorithm 2 in Sec. IX in Information for Reproducibility specifies the complete estimation procedure. The estimated R_s and CFO determine the CFs for CC extraction. The CC definition, matrix construction,

TABLE I
ACCURACY–COMPLEXITY COMPARISON OF THE PROPOSED CONFIGURATIONS AND REFERENCE METHODS.

Classification Method		Computational Complexity					Classification Accuracy ²		
Method ³		Pre-processing ¹	Classifier Parameters	Operations Pre-processing/Classifier/Total	CPU Latency (ms)	Speedup Operation/CPU	Train Set	Test on 2018	Test on 2022
I/Q–BOI–USP [18]		$8N + 4N \log_2(N)$	CNN: 2,757,816	2,228,224/7,428,120,704/ 7,430,348,928	440.44	$0.013 \times /$ $0.07 \times$	2018	$88.9 \pm 0.4\%$	$79.9 \pm 0.5\%$
							2022	$86.1 \pm 0.3\%$	$90.0 \pm 0.3\%$
Full Eight-Class SSCA–CC–CAP [4]		$NN'(\log_2(N') + \log_2(N)) + 4NN' + (3 + P)N + (K + 1)N \log_2(N)$	CAP: 1,267,936	62,095,360/34,900,152/ 96,995,512	31.46	$1.00 \times /$ $1.00 \times$	2018	$91.4 \pm 0.1\%$	$92.0 \pm 0.3\%$
							2022	$90.6 \pm 0.2\%$	$93.6 \pm 0.1\%$
Full Eight-Class FFT–CC–CAP		$8N + 3N \log_2(N) + PN + KN \log_2(N)$	CAP: 1,267,936	10,813,440/34,900,152/ 45,713,592	12.91	$2.12 \times /$ $2.44 \times$	2018	$91.5 \pm 0.1\%$	$92.4 \pm 0.1\%$
							2022	$90.6 \pm 0.3\%$	$93.5 \pm 0.1\%$
Full Eight-Class SSCA–CC–CNN		$NN'(\log_2(N') + \log_2(N)) + 4NN' + (3 + P)N + (K + 1)N \log_2(N)$	CNN: 111,148	62,095,360/14,382,660/ 76,478,020	24.48	$1.27 \times /$ $1.29 \times$	2018	$91.9 \pm 0.1\%$	$93.0 \pm 0.2\%$
							2022	$91.1 \pm 0.1\%$	$94.4 \pm 0.1\%$
Full Eight-Class FFT–CC–CNN		$8N + 3N \log_2(N) + PN + KN \log_2(N)$	CNN: 111,148	10,813,440/14,382,660/ 25,196,100	5.93	$3.85 \times /$ $5.31 \times$	2018	$91.8 \pm 0.1\%$	$93.0 \pm 0.2\%$
							2022	$91.0 \pm 0.1\%$	$94.4 \pm 0.1\%$
Direct Configuration		$8N + 3N \log_2(N)$	LR Classifier: 264	1,736,704/256/ 1,736,960	0.65	$55.84 \times /$ $48.40 \times$	2018	$85.5 \pm 0.0\%$	$77.6 \pm 0.0\%$
							2022	$74.3 \pm 0.0\%$	$82.0 \pm 0.0\%$
Hierarchical Configuration	Non-QPSK- like Path	$8N + 3N \log_2(N)$	LR Classifier: 165	1,736,704/160/ 1,736,864	0.65	$55.85 \times /$ $48.40 \times$	2018	$91.9 \pm 0.1\%$	$93.5 \pm 0.2\%$
	QPSK- like Path	$8N + 3N \log_2(N) + PN + KN \log_2(N)$	LR Classifier: 165 CNN: 111,064	10,813,440/14,382,740/ 25,196,180	5.96	$3.85 \times /$ $5.28 \times$	2022	$91.1 \pm 0.1\%$	$94.5 \pm 0.0\%$

1: Following Ref. [4], we set the number of SSCA strips to $N' = 64$, with $N = 32,768$, $P = 7$, and $K = 18$, and exclude the common BOI processing from the reported counts. Each real/complex arithmetic operation, MAC, comparison, or elementary-function evaluation counts as one operation. Our implementation of the fixed-resolution BOI method [11] costs an estimated $3N \log_2(N) + 65N = 3,604,480$ operations. At $q = 0.5$, the reported speedup of $7.20 \times$ follows the counting scope of Ref. [4]; including the shared BOI cost gives $5.89 \times$. 2: Table I reports five-seed mean $\pm$ sample standard deviation (SD) from our implementations on fixed matched test splits, rounded to one decimal. The SD reflects training-seed variation; deterministic Direct LR therefore has SD 0.0. As the implementations of Refs. [4], [18] are not public, the baseline accuracies are from our reproductions using the reported model and training settings, without extra hyperparameter search; the best checkpoint was selected on the fixed validation set. For reference, published accuracies are $92.3/93.1/91.6/92.5\%$ in Ref. [4] and $90.4/86.1/88.3/92.1\%$ in Ref. [18] (train/test order: 2018/2018, 2018/2022, 2022/2018, 2022/2022). 3: For each method, accuracy, CPU latency, and operation counts refer to the same model architecture and processing specification, with common Linux hardware/software conditions. For the Hierarchical Configuration, costs are reported by execution path, while accuracy covers the complete eight-class classifier. CPU latencies also exclude BOI processing.

and BOI-power normalization follow Ref. [4]. The resulting 11×15 matrix is supplied to a CNN for QPSK, 16QAM, 64QAM, and 256QAM classification.

E. Operation Costs and Speedup

The four Full Eight-Class configurations in Table I form a 2×2 ablation separating the operation-cost effects of FFT substitution and CAP-to-CNN replacement. Let $q = \Pr(\hat{c} = \text{QPSK-like})$ be the probability that LR routes an input to refinement. Let C_S and C_F denote the total per-input operation costs of SSCA–CC–CAP and FFT–CC–CAP, respectively, and C_N and C_Q the proposed non-QPSK-like and QPSK-like path costs in the Hierarchical Configuration. The expected operation speedup $S(q)$ relative to SSCA–CC–CAP is

$$S(q) = \underbrace{\frac{C_S}{C_F}}_{\text{FFT}} \times \underbrace{\frac{C_F}{C_Q}}_{\text{classifier}} \times \underbrace{\frac{C_Q}{(1-q)C_N + qC_Q}}_{\text{conditional}}. \quad (3)$$

For the Hierarchical Configuration, assume $q = 0.5$ for the balanced eight-class setting. From Table I substituting $C_S = 96,995,512$, $C_F = 45,713,592$, $C_N = 1,736,864$, and $C_Q = 25,196,180$ operations into Eq. (3) yields the three speedup factors $2.12 \times$, $1.81 \times$, and $1.87 \times$, respectively, and an overall speedup of $7.20 \times$. In Sec. VI in Supplementary Material, Fig. 3 shows how speedup varies with q , and Table VI shows the effect of conditional execution on accuracy.

III. EXPERIMENTAL RESULTS

A. Datasets and Experimental Setup

The CSPB.ML.2018 and CSPB.ML.2022 datasets [10] each contain 112,000 signals from eight equally represented

modulation classes. Each signal contains 32,768 complex samples. CSPB.ML.2018 uses normalized CFO values in $[-0.001, 0.001]$ and in-band signal-to-noise ratio (SNR) values in $[0, 12]$ dB, whereas CSPB.ML.2022 uses CFO values in $[0.01, 0.02]$ and in-band SNR values in $[1, 18]$ dB. Short-record results with length-specific LR/CNN retraining are reported in Sec. XI in Information for Reproducibility. Overall accuracy falls to 83.6–88.1% at $N = 8192$ and 65.3–75.1% at $N = 4096$. Sec. V in Supplementary Material describes the training/validation/test split, addresses potential signal duplication due to insufficiently independent random seeds, and reports results on CSPB.ML.2018R2 [19]. All feature and estimator choices used only the CSPB.ML.2018 training split.

Both LR classifiers used standardized inputs and balanced class weights [20]. The Multi-layer Perceptron (MLP) ablation in Table I used the same 32-dimensional inputs and splits with a 32–64–8 architecture. The CNN in the Hierarchical Configuration is a 111,064-parameter ResNet-style network [21] with 20-channel, 5×5 convolutional residual stages, global-average pooling, and a linear head. It was trained with AdamW for 200 epochs using hyperparameters selected on the CSPB.ML.2018 validation set: batch size 250, learning rate 1.3×10^{-3} , weight decay 8×10^{-4} , and label smoothing 0.12. CPU experiments used one Intel Xeon Silver 4208 core at 2.10 GHz. The implementation is in C++17, uses LibTorch [22] for inference, and oneMKL [23] for the FFT.

B. Accuracy–Complexity and Robustness Analysis

Table I shows that the Hierarchical Configuration achieves 91.1–94.5% accuracy, comparable to the published base-

Supplementary Material

Hierarchical Automatic Modulation Classification with Conditional Cyclic-Cumulant Refinement
Ruilin Wu, Xueyuan Liu, and Philip H. W. Leong

V. DATASET SPLITS AND CSPB.ML.2018R2

This paper uses the original CSPB.ML.2018/2022 datasets. Identical non-class parameter groups are kept in one 70/5/25 training/validation/test split; cross-dataset tests use all target signals. One unit contains eight signals (one per class). The numbers of multi-unit/total groups are 2,169/2,257, 2,176/2,264, and 0/14,000 for 2018, 2022, and 2018R2; their maximum sizes are seven, seven, and one units, respectively.

Only 20,000 of 112,000 2022R2 signals (Batches 1–5) are public, without labels or truth metadata, so Table V covers only 2018R2. Similar R2/R2 and 18/18 accuracies ($91.8 \pm 0.1\%$ and $91.9 \pm 0.1\%$) suggest that the known repetitions did not materially increase our result, consistent with the dataset creator’s CSP finding [19]. On R2/R2, SSCA–CC–CAP and I/Q–BOI–USP achieved $89.9 \pm 0.2\%$ and $89.3 \pm 0.6\%$.

TABLE V
GENERALIZATION PERFORMANCE OF THE HIERARCHICAL CONFIGURATION IN CSPB.ML.2018R2, FIVE-SEED MEAN $\pm$ SD.

Train/test	R2/R2	18/R2	22/R2	R2/18	R2/22
Acc. (%)	91.8 ± 0.1	91.6 ± 0.1	91.0 ± 0.1	91.8 ± 0.1	93.2 ± 0.2

Note: R2,18,22 denote CSPB.ML.2018R2, CSPB.ML.2018, and CSPB.ML.2022, respectively. 18/R2 and 22/R2 reuse Table I/III weights; others use R2-trained models.

VI. CONDITIONAL EXECUTION

Fig. 3 separates overall and scheduling-only gains. Overall compares conditional CAP/CNN pipelines with SSCA–CC–CAP, including front-end, classifier and scheduling effects; Execution only compares full and conditional refinement with identical modules. For CNN, overall/execution-only speedups are 7.20/1.87 at $q = 0.5$ and 3.85/1.00 at $q = 1$. Table VI fixes FFT/CC inputs and eight-class models, comparing direct and LR-routed decisions.

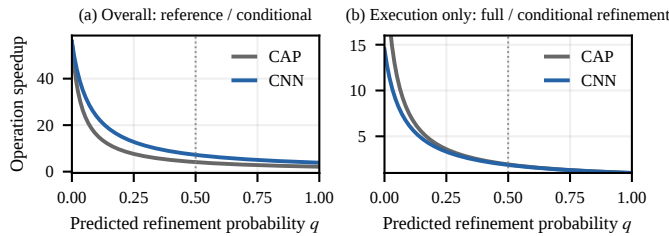


Fig. 3. Overall and execution-only speedups versus q ; dotted: $q = 0.5$.

TABLE VI
MATCHED CLASSIFIER/ROUTING ABLATION.

Model	Setting	18/18	18/22	22/18	22/22
CAP	Full	91.5 ± 0.1	92.4 ± 0.1	90.6 ± 0.3	93.5 ± 0.1
CAP	Conditional	91.4 ± 0.1	92.3 ± 0.1	90.5 ± 0.3	93.6 ± 0.1
CNN	Full	91.8 ± 0.1	93.0 ± 0.2	91.0 ± 0.1	94.4 ± 0.1
CNN	Conditional	91.8 ± 0.1	93.0 ± 0.2	91.0 ± 0.1	94.4 ± 0.1

Note: All rows use the FFT-CC inputs and eight-class models from Table I. Full and Conditional share each network and its weights to isolate routing effects. Full uses only the network; Conditional uses LR for coarse classification and the network for QPSK-like refinement, with CNN more accurate than CAP. Paired bootstrap: 95% CIs exclude zero only for 18/18 and 22/18 under both models; the maximum change is 0.081 points.

For the current Hierarchical Configuration, Table VII reports observed routing errors and their Table-I-consistent costs. Table VIII varies the true QPSK-like prior π while retaining measured TPR, FPR, and within-family accuracies. TPR is the fraction of true QPSK-like samples routed to refinement; FPR is the fraction of other-family samples routed there. Larger π therefore raises the workload.

TABLE VII
OBSERVED ROUTING AND EXPECTED COST.

Train/test	q	FR/UR	Mops	CPU N/Q/avg (ms)
18/18	0.4993	21/1	13.450	0.65/5.96/3.301
18/22	0.4990	120/5	13.442	0.65/5.96/3.300
22/18	0.4987	143/1	13.437	0.65/5.96/3.298
22/22	0.5000	0/0	13.467	0.65/5.96/3.305

Note: All rows evaluate the current Hierarchical Configuration. q is the observed fraction sent by LR to CNN refinement. FR counts true QPSK-like samples rejected before refinement and therefore unrecoverable; UR counts other-family samples refined unnecessarily. N/Q are measured; avg is their q -weighted value.

TABLE VIII
SENSITIVITY TO THE QPSK-LIKE CLASS PRIOR.

Train/test	$\pi = 0.25$	$\pi = 0.50$	$\pi = 0.75$
	Acc (%) / q / Mops	Acc (%) / q / Mops	Acc (%) / q / Mops
18/18	96.0/0.2497/7.594	91.9/0.4993/13.450	87.9/0.7489/19.305
18/22	96.7/0.2495/7.591	93.5/0.4990/13.442	90.2/0.7484/19.294
22/18	95.5/0.2494/7.587	91.1/0.4987/13.437	86.7/0.7481/19.287
22/22	97.3/0.2500/7.602	94.5/0.5000/13.467	91.8/0.7500/19.331

Note: All values reweight the current Hierarchical Configuration; π is the assumed true QPSK-like fraction and $q(\pi) = \pi \text{TPR} + (1 - \pi) \text{FPR}$. Each entry gives accuracy/ q /Mops using fixed within-family accuracies; no model is retrained or retimed, and hard-route errors remain included.

VII. R_s /CFO ESTIMATION RESULTS

Fig. 4 reports R_s /CFO success for true and LR-routed QPSK-like populations. The predicted population combines routes from the 2018 LR (18/18, 18/22) and 2022 LR (22/18, 22/22). From 3–6 dB, R_s rises to 99.0–100.0%; CFO exceeds 99.8% and nears 100.0%. The maximum True–Predicted gaps are approximately 0.8 and 0.1 percentage points for R_s and CFO, respectively. The 2018/2022 ranges end at 12/18 dB.

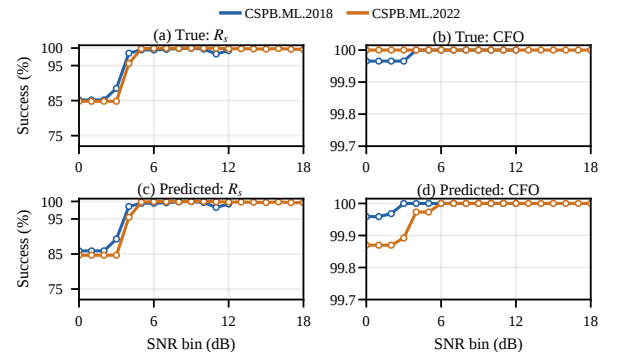


Fig. 4. R_s /CFO success versus SNR.

REFERENCES

- [1] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137–156, 2007.
- [2] Z. Zhu and A. K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications*. John Wiley & Sons, 2015.
- [3] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.
- [4] J. A. Snoop, D. C. Popescu, J. A. Latshaw, and C. M. Spooner, "Deep-learning-based classification of digitally modulated signals using capsule networks and cyclic cumulants," *Sensors*, vol. 23, no. 12, 2023. [Online]. Available: <https://www.mdpi.com/1424-8220/23/12/5735>
- [5] W. A. Gardner and C. M. Spooner, "The cumulant theory of cyclostationary time-series. I. Foundation," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3387–3408, 1994.
- [6] W. A. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a century of research," *Signal Processing*, vol. 86, no. 4, pp. 639–697, 2006.
- [7] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Cyclostationarity-based modulation classification of linear digital modulations in flat fading channels," *Wireless Personal Communications*, vol. 54, no. 4, pp. 699–717, 2010.
- [8] T. V. Câmara, A. D. Lima, B. M. Lima, A. I. Fontes, A. D. M. Martins, and L. F. Silveira, "Automatic modulation classification architectures based on cyclostationary features in impulsive environments," *IEEE Access*, vol. 7, pp. 138 512–138 527, 2019.
- [9] J. A. Snoop, J. A. Latshaw, D. C. Popescu, and C. M. Spooner, "Robust classification of digitally modulated signals using capsule networks and cyclic cumulant features," in *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, 2022, pp. 298–303.
- [10] C. M. Spooner, "Datasets," Cyclostationary Signal Processing. Accessed: Oct. 3, 2026. [Online]. Available: <https://cyclostationary.blog/data-sets/>
- [11] —, "Multi-resolution white-space detection for cognitive radio," in *MILCOM 2007 - IEEE Military Communications Conference*, 2007, pp. 1–9.
- [12] W. Brown and H. Loomis, "Digital implementations of spectral correlation analyzers," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 703–720, 1993.
- [13] A. Swami and B. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 416–429, 2000.
- [14] E. Mohammed, O. Mashaal, and H. Abou-Zeid, "Using early exits for fast inference in automatic modulation classification," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 291–296.
- [15] P. Ciblat, P. Loubaton, E. Serpedin, and G. Giannakis, "Asymptotic analysis of blind cyclic correlation-based symbol-rate estimators," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1922–1934, 2002.
- [16] A. J. Viterbi and A. M. Viterbi, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 543–551, 1983.
- [17] O. A. Dobre, M. Oner, S. Rajan, and R. Inkol, "Cyclostationarity-based robust algorithms for QAM signal identification," *IEEE Communications Letters*, vol. 16, no. 1, pp. 12–15, 2012.
- [18] J. A. Snoop, D. C. Popescu, and C. M. Spooner, "Deep-learning-based classifier with custom feature-extraction layers for digitally modulated signals," *IEEE Transactions on Broadcasting*, vol. 70, no. 3, pp. 763–773, 2024.
- [19] C. M. Spooner, "CSPB.ML.2018R2: Correcting an RNG Flaw in CSPB.ML.2018," Cyclostationary Signal Processing, Sep. 25, 2023. Accessed: Oct. 3, 2026. [Online]. Available: <https://cyclostationary.blog/2023/09/25/cspb-ml-2018r2-correcting-an-rng-flaw-in-cspb-ml-2018/>
- [20] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY: Springer, 2013.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [22] PyTorch, "PyTorch C++ API," PyTorch. Accessed: Aug. 1, 2026. [Online]. Available: <https://docs.pytorch.org/cppdocs/>
- [23] Intel Corporation, "Intel oneAPI Math Kernel Library (oneMKL)," Intel. Accessed: Aug. 11, 2026. [Online]. Available: <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html>

Information for Reproducibility

Hierarchical Automatic Modulation Classification with Conditional Cyclic-Cumulant Refinement
Ruilin Wu, Xueyuan Liu, and Philip H. W. Leong

VIII. THE 32-FEATURE CONSTRUCTION

Algorithm 1 maps three normalized FFT-bin rankings to the fixed 32-feature vector for $N = 32768$ without changing F_s . Its fixed rules were derived from the 2018 training split.

Algorithm 1: Shared 32-feature construction²

▷ Top- K finite bins, 8-bin spacing, no peak/floor test; ties follow argsort; insufficient-bin samples are counted as incorrect.

Function SelectBins(S, K):

```

 $I \leftarrow \text{argsort}_{b: \text{isfinite}(S[b])} S[b]; P \leftarrow [];$ 
foreach  $b \in I$  do
  if  $P = []$  or  $\min_{p \in P} |b - p| \geq 8$  then
     $P \leftarrow P \cup b;$ 
  if  $|P| = K$  then break;
if  $|P| < K$  then error(insufficient_bins);
 $(f_r, d_r) \leftarrow ((P_r - N/2)/N, S[P_r]), r = 1:K;$  ▷ grid bins
return ( $P, f, d$ );
```

▷ BOI filtering/centering: $x_{\text{ori}} \rightarrow x_{\text{boi}}$; P96 clipping: $x_{\text{boi}} \rightarrow x$.

Function BuildFFTFeatures($x_{\text{ori}}[0:N-1]$):

```

 $x_{\text{boi}} \leftarrow \text{BOI}(x_{\text{ori}}); \epsilon \leftarrow 10^{-12}; t_{96} \leftarrow Q_{0.96}(|x_{\text{boi}}|);$ 
 $x[n] \leftarrow \text{complex64}(x_{\text{boi}}[n] \min\{1, t_{96}/(|x_{\text{boi}}[n]| + \epsilon)\});$ 
 $w[n] \leftarrow \frac{1}{2}[1 - \cos(2\pi n/(N-1))];$ 
 $u_2 \leftarrow x^2; u_4 \leftarrow u_2^2; u_a \leftarrow |x|^2;$  ▷ each complex64
foreach  $q \in \{2, 4, a\}$  do
   $F_q \leftarrow \text{fftshift}(\text{FFT}(w u_q)); H \leftarrow \max_k |F_q[k]|;$ 
  if  $H \leq 0$  or  $\neg \text{isfinite}(H)$  then error(invalid_FFT);
   $S_q[b] \leftarrow \text{float32}(20 \log_{10}(\frac{|F_q[b]|}{H + \epsilon} + \epsilon));$ 
   $K_2 = K_4 \leftarrow 8; K_a \leftarrow 16;$ 
   $(P_q, f_q, d_q) \leftarrow \text{SelectBins}(S_q, K_q);$ 
foreach  $q \in \{2, 4\}$  do
   $j_{q,i}^* \leftarrow \arg \min_{j=1:8} |f_{q,i} + f_{q,j}|, i: |f_{q,i}| > 5/N;$ 
   $T_q \leftarrow [|d_{q,i} - d_{q,j_{q,i}^*}| + 40|f_{q,i} + f_{q,j_{q,i}^*}|]_i;$  ▷ keep repeats
   $(t_1, \dots, t_m) \leftarrow \text{sort}^\uparrow(T_q); (s_{q,1}, s_{q,2}) \leftarrow (99, 99);$ 
  if  $m > 0$  then  $(s_{q,1}, s_{q,2}) \leftarrow (t_1, t_{\min(2,m)});$ 
 $A_{q,r} \leftarrow |f_{q,r}|; \Delta_{q,ij} \leftarrow d_{q,i} - d_{q,j}; g_r \leftarrow A_{a,2r}; h_r \leftarrow d_{a,2r};$ 
 $E_{q,\Sigma} \leftarrow 10^{d_{q,r}/10};$ 
 $E_a \leftarrow \sum_{b=0}^{N-1} \max(10^{S_a[b]/10}, 10^{-12}) + 10^{-30};$ 
 $\rho_2 \leftarrow \sum_{r=1}^4 E_{2,r} / \sum_{r=1}^8 E_{2,r}; \rho_a \leftarrow \sum_{r=1}^{16} E_{a,r} / E_a^\Sigma;$ 
 $c_4 \leftarrow \sum_{r=1}^8 \mathbf{1}[A_{4,r} > 5/N \wedge d_{4,r} > -20];$ 
 $c_a \leftarrow \sum_{r=1}^{16} \mathbf{1}[A_{a,r} > 5/N \wedge d_{a,r} > -10];$ 
▷ Pack in the trained LR model's fixed feature order.
 $v_{1:4} \leftarrow (f_{2,5}, f_{2,8}, \Delta_{2,56}, \rho_2);$ 
 $v_{5:8} \leftarrow (s_{2,1}, s_{2,2}, A_{4,1}, A_{4,2});$ 
 $v_{9:12} \leftarrow (d_{4,3}, A_{4,6}, f_{4,7}, f_{4,8});$ 
 $v_{13:16} \leftarrow (A_{4,8}, \Delta_{4,12}, \Delta_{4,23}, \Delta_{4,67});$ 
 $v_{17:20} \leftarrow (c_4, s_{4,1}, s_{4,2}, d_{2,3});$ 
 $v_{21:24} \leftarrow (\Delta_{4,78}, \Delta_{2,12}, f_{2,7}, f_{4,2});$ 
 $v_{25:28} \leftarrow (\rho_a, h_1, h_6, h_3 - h_5);$ 
 $v_{29:32} \leftarrow (h_2 - h_3, g_3/(g_1 + 10^{-9}), h_5 - h_6, c_a);$ 
return ( $\text{float32}(v_{1:32}), S_4, d_a, f_a, x_{\text{boi}}$ );
```

IX. R_s /CFO AND CYCLIC CUMULANTS

Algorithm 2 estimates R_s /CFO from the FFT spectra of $|x[n]|^2$ and $x^4[n]$ and constructs the CC matrix. Its fixed thresholds and rules were derived from the 2018 training split.

X. FFT/SSCA COMPARISON

All methods use the same 32 positions and frozen LR. P32 integrates the complex SCF over f , CF32 retains its peak over f , and SSCA uses x and x^2 branches. Five-seed tests use 60 training and 30 test signals/class/condition, with $N = 32768$.

Algorithm 2: Blind R_s /CFO estimation and CC matrix construction²

▷ $p^4 \geq 0 \Rightarrow |G_4(\nu)| \leq G_4(0)$: $4f_\Delta$ is strongest; the maximum BOI residual in the 2018 training split is 0.02821, rounded to 0.03, so the x^4 search bound is $0.12 = 4 \times (0.03)$; finite data/clipping or larger residuals may fail.

Function EstimateRsCFO(S_4, d_a, f_a):

```

 $[b]_N \leftarrow b \bmod N; f_b \leftarrow (b - N/2)/N; \tau \leftarrow 23.94 \text{ dB};$ 
 $c \leftarrow \arg \max_b S_4[b];$ 
▷  $\tau$ : selected by a candidate-threshold sweep on QPSK-like signals in the 2018 training split for near-optimal one-bin  $R_s$  accuracy.
 $m \leftarrow \lfloor 25N/256 \rfloor; D[b] \leftarrow S_4[b] - \frac{1}{2m+1} \sum_{r=-m}^m S_4[b+r]_N;$ 
 $\mathcal{K} \leftarrow \{[0.03N], \dots, [0.5001N]\};$  ▷ Range: in the 2018 training split,  $R_s^{\min} = 0.0444$ . We set  $0.03 < R_s^{\min}$  and  $0.5001 > 0.5$  (Nyquist), covering all training rates and the Nyquist endpoint.
 $k^* \leftarrow \arg \max_{k \in \mathcal{K}} \{D[[c-k]_N] + D[[c+k]_N]\};$ 
 $\hat{R}_s \leftarrow k^*/N; r_a \leftarrow |f_{a,2}|;$  ▷  $R_s$ : DFT grid, no interpolation
if  $d_{a,1} - d_{a,2} \leq \tau \wedge r_a \in [0.03, 0.5001] \wedge |r_a - \hat{R}_s| > 1/N$  then
   $\hat{R}_s \leftarrow r_a;$  ▷ cross-check with  $|x|^2$ 
 $b^* \leftarrow \arg \max_{b: |f_b| \leq 0.12} S_4[b];$  ▷ argmax ties: first
 $(L, C, R) \leftarrow (S_4[b^* - 1], S_4[b^*], S_4[b^* + 1]); \eta \leftarrow L - 2C + R;$ 
 $\delta \leftarrow 0;$ 
if  $\eta \neq 0$  then  $\delta \leftarrow \text{clip}((L - R)/(2\eta), -1/2, 1/2);$ 
 $\hat{f}_\Delta \leftarrow \text{wrap}_{[-1/2, 1/2]}((f_b^* + \delta/N)/4);$  ▷ 3-bin log-dB parabola;
 $|\delta| \leq 1/2$ 
return ( $\hat{R}_s, \hat{f}_\Delta$ ); ▷ cycles/sample
```

Function CC_matrix($x_{\text{boi}}, \hat{R}_s, \hat{f}_\Delta, P_{\text{BOI}}$):

```

 $\mathcal{M} \leftarrow [(2, 0:2), (4, 0:4), (6, 0:6)]; T \leftarrow \mathbf{0}_{11 \times 15};$ 
 $\mathcal{A} \leftarrow \{(2, 1), (4, 0), (4, 2), (4, 4), (6, 1), (6, 3), (6, 5)\};$ 
 $\mathcal{Q}_{\text{mask}} \leftarrow \{(j, k) : \mathcal{M}_j \in \mathcal{A}, |k| \leq 2 - \mathbf{1}[\mathcal{M}_j = (2, 1)]\};$ 
 $\alpha_{j,k} \leftarrow (n_j - 2m_j)\hat{f}_\Delta + k\hat{R}_s, (n_j, m_j) = \mathcal{M}_j;$ 
 $C \leftarrow \text{Cumulant}(x_{\text{boi}}; \{\alpha_{j,k}\});$ 
foreach  $(j, k) \in \mathcal{Q}_{\text{mask}}$  do
   $T[k + 6, j] \leftarrow |C_{j,k}|^{2/n_j} / (P_{\text{BOI}} + 10^{-12});$ 
return  $\text{float32}(T);$  ▷ CNN input:  $1 \times 11 \times 15$ 
```

$F_s^{\text{out}} = F_s^{\text{in}}, f_\Delta^{\text{truth}} = f_{\text{CFO}}^{\text{truth}} - f_{\text{center}};$

The flat-channel baseline uses 8 samples/symbol, a 12-symbol pulse, $\beta = 0.35$, 10-dB SNR, and CFO in $[-0.02, 0.02]$. Each test changes only SNR, β , CFO, or the three-tap channel (delays 0/4/10; gains $1/0.55e^{-0.7j}/0.25e^{-1.2j}$).

TABLE IX
CONTROLLED FIVE-CLASS ACCURACY (%): MEAN $\pm$ SD.

Method	Base	CFO	Multipath	$\beta = 0.1$	$\beta = 0.8$	0 dB
FFT-32D	100.0 $\pm$ 0.0	98.0 $\pm$ 1.9	81.6 $\pm$ 2.2	98.1 $\pm$ 1.3	87.3 $\pm$ 1.6	44.8 $\pm$ 3.5
SSCA-P32	100.0 $\pm$ 0.0	98.0 $\pm$ 1.9	81.6 $\pm$ 2.2	98.1 $\pm$ 1.3	87.3 $\pm$ 1.6	44.8 $\pm$ 3.5
SSCA-CF32	100.0 $\pm$ 0.0	96.4 $\pm$ 3.5	94.0 $\pm$ 3.4	96.8 $\pm$ 1.3	91.3 $\pm$ 4.4	60.0 $\pm$ 0.0

XI. SHORT-RECORD SENSITIVITY

Signals are center-cropped from 32,768 samples; BOI and CC generation remain unchanged. For each N and year, the LR router and five CNNs are retrained with the original settings and five seeds. FFT length and bin thresholds scale with N ; R_s /CFO tolerances are $1/N$ and $1/(4N)$.

TABLE X
SHORT-RECORD SENSITIVITY.

N	R_s acc. (18/22)	CFO acc. (18/22)	Classification accuracy: routing/overall (Train/Test)				Mops
			18/18	18/22	22/18	22/22	
32768	98.0/99.4	100.0/100.0	99.9/91.9	99.9/93.5	99.9/91.1	100.0/94.5	13.467
16384	92.2/97.5	99.5/99.9	99.5/88.6	99.6/91.0	99.2/88.2	99.9/92.3	10.132
8192	83.0/89.9	94.7/98.4	98.4/84.0	97.6/85.6	97.9/83.6	99.3/88.1	8.564
4096	29.0/53.6	65.5/85.7	81.8/65.6	80.3/65.3	83.1/66.5	90.0/75.1	7.828

Accuracies are percentages. Classification entries give LR routing/five-run mean overall accuracy. Mops is the mean count at $q = 0.5$ under the counting scope of Table I. R_s /CFO use all true QPSK-like 2018/2022 signals, in 18/22 order. Lower N reduces R_s /CFO accuracy, which in turn affects overall accuracy.

²Python implementation is available at https://github.com/ruilin-wu/Hierarchical_AMC/tree/main/helper_function

TABLE II
ABLATION OF THE FFT REPRESENTATION AND CLASSIFIER.

Configuration	Task	Train on 2018		Train on 2022	
		2018	2022	2018	2022
All three FFT projections	Five-class LR	99.9%	99.9%	99.9%	100.0%
Without $\text{FFT}\{ x ^2\}$	Five-class LR	99.9%	99.9%	99.8%	100.0%
Without $\text{FFT}\{x^2\}$	Five-class LR	98.6%	93.5%	97.9%	99.1%
Without $\text{FFT}\{x^4\}$	Five-class LR	74.3%	73.7%	74.9%	74.9%
32-D FFT-Features	Eight-class LR	85.5%	77.6%	74.3%	82.0%
32-D FFT-Features	Eight-class MLP	85.8%	77.3%	75.4%	82.9%

TABLE III
BLIND PARAMETER ESTIMATION AND CLASSIFICATION ACCURACY FOR THE HIERARCHICAL CONFIGURATION.

Parameters	R_s	CFO	Train 2018		Train 2022	
	Acc. (18/22)	Acc. (18/22)	2018	2022	2018	2022
Blind R_s /CFO	98.0/99.4%	100.0/100.0%	91.9%	93.5%	91.1%	94.5%
True R_s only	—	—	92.2%	93.5%	91.5%	94.8%
True CFO only	—	—	92.0%	93.5%	91.2%	94.5%
Ground-truth R_s /CFO	—	—	92.1%	93.5%	91.6%	94.7%

Note: Tolerances: $1/N$ for R_s and $1/(4N)$ for CFO. Estimation accuracies use all true QPSK-like signals from the 2018/2022 datasets. Classification accuracies are five-seed means over all eight classes using the CNN weights and fixed LR routing in Table I. Paired bootstrap: 95% CIs exclude zero for 18/18 and 22/22 with true R_s (alone or with true CFO), and for all 22/18 cases, but not for 18/22; maximum change is 0.487 points.

lines [4], [18]. The hierarchical structure reduces computation while preserving the accuracy of the CNN classifier. For QPSK-like signals, its CPU latency is 5.96 ms, compared with 12.91 ms for FFT-CC-CAP and 31.46 ms for SSCA-CC-CAP. Tables VII and VIII in Supplementary Material report routing rates, false-family rejection, recalculated average operation counts and latency, and class-prior sensitivity.

Table III shows that the LR classifier using all three FFT projections achieves 99.9–100.0% five-class accuracy. Removing $\text{FFT}\{x^4[n]\}$ reduces accuracy to 73.7–74.9%, whereas removing $\text{FFT}\{x^2[n]\}$ mainly impairs cross-dataset generalization; $\text{FFT}\{|x[n]|^2\}$ has little effect on classification accuracy but remains necessary for R_s estimation. The eight-class MLP reaches 75.4–85.8%, leaving substantial QAM confusion.

Table IX in Information for Reproducibility shows identical results for FFT-32D and integrated SSCA-P32; complex128 integration also matches direct FFT on 4500 waveforms (maximum relative error 3.39×10^{-15}). Under three-tap multipath and at 0 dB, FFT-32D falls to 81.6/44.8%, whereas SSCA-CF32 gives 94.0/60.0%. CSPB.ML.2018 and 2022 have single signals and no multipath, so the 99.9–100.0% routing in Table III may not generalize to fading; refinement cannot correct a wrong coarse route. As shown in Table IV, errors mainly occur on QPSK-like signals. Within this family, 64QAM and 256QAM share a CF pattern and have similar higher-order CCs at low SNR.

C. Parameter Estimation and Cross-Dataset Generalization

Table III compares blind and ground-truth parameters while keeping the CNN weights and LR routing fixed. Replacing only the estimated symbol rate changes accuracy by 0.0 to +0.4 percentage points, while replacing only the estimated

TABLE IV
COMMON-SNR STANDARDIZED ACCURACY (%).

Train/test	All eight	Non-QPSK-like	QPSK-like
18/18	92.1 ± 0.1	100.0	84.3
18/22	91.7 ± 0.1	100.0	83.4
22/18	91.4 ± 0.0	100.0	82.8
22/22	93.3 ± 0.1	100.0	86.6

Note: All rows use common 1–12 dB SNR-bin weights pooled from 18/18 and 22/22.

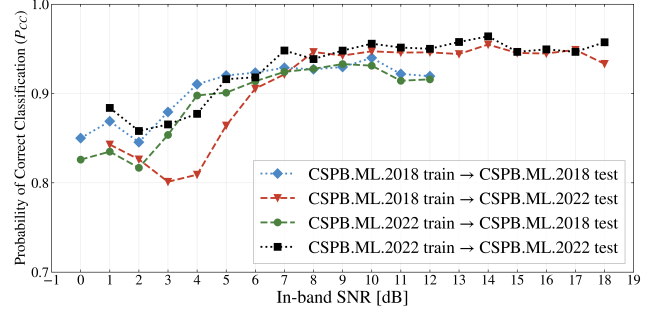


Fig. 2. SNR-dependent accuracy of the Hierarchical Configuration.

CFO changes it by at most 0.1 percentage points. Sec. VII in Supplementary Material reports estimator accuracy by dataset, SNR, and true versus predicted QPSK-like groups.

Fig. 2 further shows that classification accuracy generally improves with SNR, with the largest degradation occurring in the low-SNR region, suggesting that low-SNR QAM ambiguity is a primary source of cross-dataset error.

CSPB.ML.2018 and CSPB.ML.2022 have different SNR and CFO distributions. For a direct comparison, Table IV uses common 1–12 dB SNR-bin weights and the same model weights as Tables II and III. With the common SNR weights, the accuracy of 18/22 is only 0.4 percentage points lower than that of 18/18, showing stable cross-dataset performance. Across all settings, overall/Non-QPSK-like/QPSK-like accuracies are 91.4–93.3%/approximately 100.0%/82.8–86.6%. These results support generalization to the signal and channel conditions covered by CSPB.ML.2018 and CSPB.ML.2022.

IV. CONCLUSION

This paper presented two AMC configurations for fast classification in real-time receivers with limited resources. The Direct Configuration has the lowest expected cost, whereas the Hierarchical Configuration applies blind R_s /CFO estimation and refinement only to QPSK-like signals and does not require true R_s or CFO. It achieves accuracy comparable to SSCA-CC-CAP with an estimated $7.20\times$ operation speedup. However, the CSPB.ML datasets contain synthetic single signals without multipath. Future work will include validation using real over-the-air signals with interference, time-varying fading, and RF hardware impairments, as well as methods to improve accuracy on short records.

ACKNOWLEDGMENT

The authors thank Chad M. Spooner for his comments. OpenAI ChatGPT generated five schematic icons in Fig. I.