# A Unified Backbone–Expert Framework with Relation-Token and Residual–Classifier Interfaces for Automatic Modulation Recognition

Zhixiang Deng[a], Houbiao Li[a,*], Zongyong Cui[b]

[a]*School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, 611731, China*
[b]*School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China*

**Abstract**

Automatic modulation recognition (AMR) faces distinct representation bottlenecks under varying observation lengths, where a single model architecture often fails to excel. To address this, we propose a unified backbone-expert framework with a common convolutional state-space backbone and two specialized interfaces. For short sequences, we inject explicit lag-aware complex-plane descriptors as relation tokens before encoding to compensate for information loss. For long sequences, we design a gated multi-scale residual refinement module to correct the feature map, combined with a fixed-averaging classifier collaboration to harness complementary evidence. Our framework achieves overall average accuracies of $67.28 \pm 0.14\%$ on RML2016.10b and $87.19 \pm 0.77\%$ on HisarMod2019 (mean $\pm$ sample standard deviation over three runs), respectively. The framework's efficacy is further validated through three-seed ablations, native-length cross-configuration tests, and controlled window studies, confirming the benefit of expert-interface decoupling over one-size-fits-all architectures.

*Keywords:* Automatic modulation recognition, Backbone–expert framework, State-space model, Relation token, Classifier collaboration

*Corresponding author.
*Email addresses:* m15181178952@163.com (Zhixiang Deng), lihoubiao0189@163.com (Houbiao Li), zycui@uestc.edu.cn (Zongyong Cui)

## 1. Introduction

Automatic modulation recognition (AMR) identifies the modulation format of a received signal without transmitter-side metadata. It supports spectrum monitoring, cognitive radio, interference analysis, and non-cooperative communications. Early deep AMR systems learned representations directly from raw I/Q samples with convolutional networks [2]. Later work modeled temporal dependencies with recurrent and hybrid networks [8, 9]. Other studies explored multimodal Transformers [10], lightweight complex-valued networks [11], and mixture-of-experts classifiers [13].

Several recent models have been evaluated on both short- and long-observation benchmarks [10, 11]. These studies demonstrate broad applicability, but the available observation budget contributes to different representation bottlenecks. With limited temporal support, informative local relations can be difficult to recover under noise. Extended support provides richer evidence, but it also produces a longer feature sequence in which channel-distorted responses must be refined and complementary cues must be aggregated into one decision. Together with the signal and channel distribution of a benchmark, these differences motivate examining whether one generic expert form addresses both regimes effectively.

We therefore pair a common convolutional state-space backbone with two complementary expert designs. On RML2016.10b, lag-aware distance and complex-correlation descriptors supply explicit local complex-plane relations, and an early token interface allows the sequence encoder to contextualize them jointly with learned backbone features. On HisarMod2019, multi-scale residual refinement corrects the feature map over long receptive fields, while decision-level collaboration combines convolutional and state-space classification evidence. The resulting framework unifies the backbone and feature dimensions at the architectural level while assigning a corresponding expert inductive bias and integration interface to each evaluated benchmark condition.

The main contributions are as follows:

- We propose a unified backbone–expert framework in which a common convolutional state-space topology supports two expert inductive biases and their corresponding integration interfaces.
- We design two complementary configurations. Relation-aware token augmentation supplies explicit local complex-plane structure before

contextual encoding in the short regime. Multi-scale residual refinement and CNN–state-space decision collaboration support the long regime.

- We evaluate the framework with three-seed ablations, SNR-stratified analysis, native-length cross-configuration experiments, and controlled HisarMod2019 windows.

The remainder of this paper is organized as follows. Section 2 reviews deep AMR and expert-integration methods. Section 3 presents the proposed framework. Section 4 describes the evaluation protocol, and Section 5 reports the results. Section 6 discusses the scope and limitations, followed by the conclusion in Section 7.

## 2. Related work

### *2.1. Deep automatic modulation recognition*

Deep AMR replaces handcrafted decision statistics with representations learned from signal samples. Convolutional models established that local patterns can be learned directly from raw I/Q sequences [2], while recurrent and convolutional–recurrent architectures introduced explicit temporal modeling [8, 9]. A broader review by Zhang et al. [14] organizes these developments around neural architectures, input representations, benchmark datasets, complexity, and deployment challenges. These families remain strong baselines, but their local receptive fields or sequential recurrence can limit the efficient modeling of widely separated signal events.

Attention-based models address this limitation by modeling nonlocal dependencies. IQFormer combines convolution and staged Transformer blocks after dynamically embedding I/Q and time–frequency modalities [10]. MST instead constructs parallel resolutions and exchanges information through cross-scale token fusion [15]. Graph-based methods offer another route: STF-GCN represents spatial, temporal, and frequency information as graph nodes and uses adaptive correlation to construct their connectivity [16]. These methods demonstrate the effectiveness of contextual and multi-domain modeling in their respective evaluation settings. However, they do not directly examine whether the same specialist representation and integration interface remain equally suitable across substantially different observation budgets.

Recent studies have also introduced selective state-space models into AMR. DWMTN combines parallel Mamba and Transformer branches using

fusion weights predicted for each input [27]. ConvMamba combines convolution, Mamba2, and soft-threshold denoising for long-sequence high-order modulation recognition [28]. LM-GDMAF uses lightweight Mamba modules to extract temporal features and fuses I/Q and spectrogram modalities in a small-sample setting [29]. These studies modify or combine sequence backbones to improve contextual modeling and feature fusion. The present work does not modify the selective state-space operator itself. Instead, it uses a common state-space sequence backbone to study how signal-specific expert information and its integration interface should be organized under the two evaluated benchmark conditions.

### *2.2. Complex-valued and relation-aware signal representations*

The I/Q channels jointly describe a complex baseband signal. Processing them as unrelated real channels can therefore discard useful coupling. CP-PCNet preserves this structure with complex partial pointwise convolution while controlling computational cost [11]. DualFormer forms separate I and Q token streams, encodes them with shared Transformer parameters, and fuses them at the prediction head [17]. Both methods respect complex signal structure, although they learn the relevant relations implicitly.

Explicit relations provide a complementary inductive bias. STF-GCN derives graph connectivity from correlations among multidomain features [16]. Multimodal systems instead combine I/Q samples with time–frequency maps or constellation diagrams [10, 18]. These methods differ in what specialist information is constructed and where it enters the network. Our short path uses local lagged distance and complex correlation, then injects the resulting representation at the token interface. Unlike general multimodal fusion, it requires neither an additional signal transform nor an image modality.

### *2.3. Feature refinement and complementary classification for long observations*

Long-observation AMR has been approached through signal restoration, multi-scale feature extraction, and hybrid sequence modeling. DAE-CNN-BiLSTM first applies a denoising autoencoder. Separate CNN and BiLSTM branches then extract and concatenate spatial and temporal features [25]. MAWDN instead learns an adaptive wavelet decomposition and aggregates decisions through a residual classification flow [21]. Both approaches exploit complementary information, but the former reconstructs a cleaner waveform and the latter explicitly decomposes the signal into time–frequency scales.

Other models enlarge the effective context directly in the discriminative network. WACN combines local-window processing with depthwise and dilated convolution to balance local attention and receptive-field growth [22], while DFENet uses parallel kernels to extract features at several temporal scales [23]. MILAFormer further combines convolutional feature enhancement, bidirectional recurrence, hierarchical attention, and Mamba-inspired linear attention [24]. These studies establish the value of denoising, multi-scale context, and complementary sequence representations. Our long configuration follows a different organization: it predicts a residual correction directly in the discriminative feature space and combines independently formed CNN and state-space decisions, without waveform reconstruction or an explicit transform-domain decomposition.

### *2.4. Expert representations and integration interfaces*

Existing AMR fusion mechanisms operate at several interfaces. IQFormer performs dynamic fusion during token embedding [10]. The adaptive multimodal framework of Guo et al. uses gated attention to combine features from three signal domains [18] and DualFormer merges independently encoded I/Q streams near the prediction head [17]. MoE-AMC takes a different approach by assigning specialized experts to low- and high-SNR regimes through a learned gate [13]. These studies establish that specialist representation, expert inductive bias, and integration location can all affect AMR performance.

Multi-scale sequence models are also related. MST fuses tokens produced by several convolutional resolutions within the same observation [15], while ms-Mamba applies state-space blocks at different sampling rates to capture temporal structure efficiently [19]. Such designs enrich how a given observation is analyzed. Our framework instead starts from the representation bottleneck observed in each benchmark condition and assigns a corresponding expert and interface within one backbone template. Local relation tokens enrich evidence before contextual encoding in the short setting. Residual feature correction and classifier collaboration exploit the broader support available in the long setting. The framework therefore distinguishes relation-token augmentation from residual–classifier collaboration, rather than applying generic multi-scale processing throughout.

## 3. Proposed method

### *3.1. Problem formulation*

Consider a transmitted complex baseband sequence $s_m[\ell]$ whose modulation class is $m \in \mathcal{M} = \{1, \ldots, C\}$. A generic discrete-time received signal can be written as [14]

$$r[\ell] = e^{j(\omega\ell+\phi)} \sum_{k=0}^{K_h-1} h_k s_m[\ell - k] + n[\ell], \qquad \ell = 1, \ldots, L, \tag{1}$$

where $\{h_k\}_{k=0}^{K_h-1}$ denotes the channel response, $\omega$ and $\phi$ are frequency and phase offsets, and $n[\ell]$ is additive noise. This expression is a general abstraction, the exact channel and impairment distributions follow each benchmark protocol.

The receiver stores an observation as

$$\mathbf{x} = \begin{bmatrix} \mathrm{Re}(r[1{:}L]) \\ \mathrm{Im}(r[1{:}L]) \end{bmatrix} \in \mathbb{R}^{2\times L}. \tag{2}$$

Given a labeled dataset $\mathcal{D} = \{(\mathbf{x}_n, m_n)\}_{n=1}^{N}$, AMR learns a posterior classifier $p_{\boldsymbol{\theta}}(m \mid \mathbf{x}, L)$. Its primary objective is the empirical cross-entropy

$$\mathcal{L}_{\mathrm{cls}} = -\frac{1}{N} \sum_{n=1}^{N} \log p_{\boldsymbol{\theta}}(m_n \mid \mathbf{x}_n, L_n). \tag{3}$$

The proposed classifier separates a common representation template from configuration-specific expert processing. The short and long configurations retain the same backbone topology but use different expert inductive biases and integration interfaces.

### *3.2. Unified backbone–expert organization*

The observation budget changes the amount and temporal span of the evidence available to the classifier, while the signal and channel distribution determine how informative that evidence is. Figure 1 illustrates the temporal-support difference using two nested windows from the same synthetic QPSK signal realization. The 128-sample window provides fewer samples from which to extract discriminative structure, motivating investigation of explicit local lag-dependent relations before contextual sequence modeling.

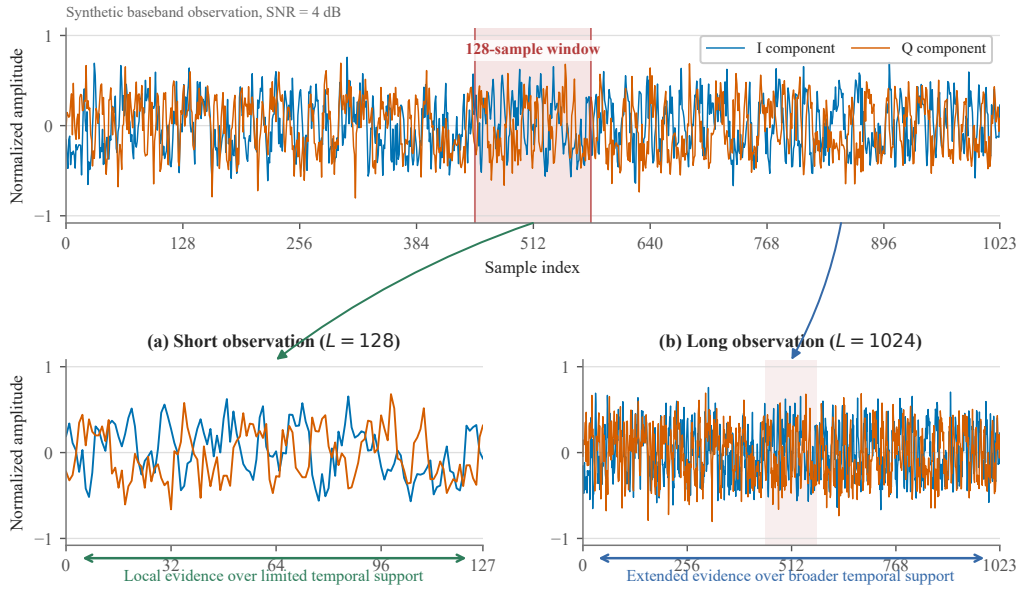


Figure 1: Illustration of short and long observation budgets using the same synthetic QPSK signal realization. The 128-sample observation is a contiguous window of the 1024-sample observation. The shorter window contains signal evidence over limited temporal support, whereas the longer window retains evidence distributed over a broader temporal range.

The 1024-sample observation covers a broader temporal range, motivating investigation of multi-scale refinement and complementary classifiers. Accordingly, the two evaluated configurations retain a common backbone topology but employ different expert inductive biases and integration interfaces for their complete benchmark conditions.

To formalize this organization, let $f_{\mathrm{s}}(\cdot;\boldsymbol{\theta}_{\mathrm{s}})$ and $f_{\mathrm{l}}(\cdot;\boldsymbol{\theta}_{\mathrm{l}})$ denote the short- and long-observation configurations:

$$\mathbf{h}_b = \mathcal{B}(\mathbf{x};\boldsymbol{\theta}_b^{\mathcal{B}}), \tag{4}$$

$$f_b(\mathbf{x};\boldsymbol{\theta}_b) = \mathcal{I}_b\big(\mathbf{h}_b, \mathcal{E}_b(\mathbf{x},\mathbf{h}_b;\boldsymbol{\theta}_b^{\mathcal{E}});\boldsymbol{\theta}_b^{\mathcal{I}}\big)\,, \qquad b \in \{\mathrm{s},\mathrm{l}\}, \tag{5}$$

where $\mathcal{B}$ denotes the common backbone topology, $\mathcal{E}_b$ is the configuration-specific expert, and $\mathcal{I}_b$ is its integration interface. The notation allows an expert to use the raw observation, an intermediate backbone representation, or both. The short interface injects position-aligned relation descriptors into the sequence tokens before contextual encoding. The long interface applies

residual feature refinement and combines independently formed convolutional and state-space decisions.

Each benchmark uses the corresponding configuration, which is trained and executed independently. The two configurations use the same backbone topology and feature dimensions but have different learned weights. In the experiments, $f_{\mathrm{s}}$ is instantiated for RML2016.10b at its native length of 128 and $f_{\mathrm{l}}$ for HisarMod2019 at its native length of 1024. The cross-configuration study reverses these assignments to assess whether the proposed expert designs contribute beyond a uniformly stronger configuration. The resulting organization is summarized in Fig. 2.

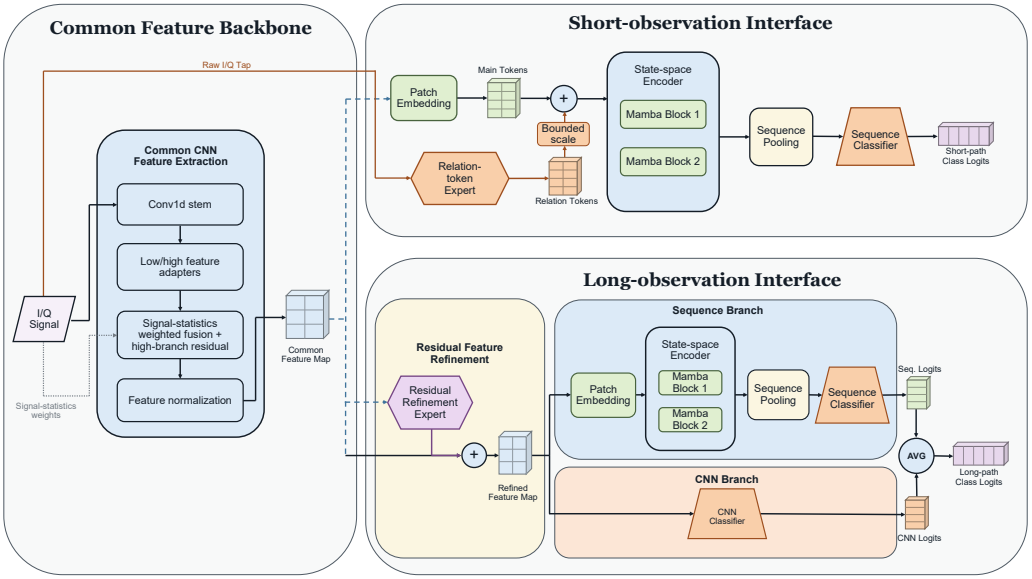


Figure 2: Overall backbone–expert framework. The short and long configurations use the same feature-extraction and state-space topology but independently trained weights. They differ in expert inductive bias and integration interface.

### *3.3. Common convolutional state-space backbone*

The two configurations use a common architectural template for initial feature extraction and sequence encoding. The experiments use a parameter-matched ordinary Conv1d stem. The stem applies a depthwise-separable convolution [5] with a kernel size of 31, followed by channel recalibration and two convolutional mixing layers. This choice keeps the stem compact and focuses the architecture on the configuration-specific experts.

The stem output $\mathbf{F}_0$ is processed by two complementary adapters. A depthwise branch with kernel size 7 emphasizes local context, whereas a standard convolutional branch with kernel size 3 provides channel mixing. A signal-statistics mixer maps eight global statistics $\boldsymbol{\xi}(\mathbf{x})$ to two normalized weights $\pi_1$ and $\pi_2$. These statistics are the means and standard deviations of I and Q, the mean and standard deviation of amplitude, mean power, and the amplitude coefficient of variation. The common feature map is

$$\mathbf{F} = \mathrm{BN}(\pi_1 A_1(\mathbf{F}_0) + \pi_2 A_2(\mathbf{F}_0) + A_2(\mathbf{F}_0)) \,. \tag{6}$$

The additional $A_2(\mathbf{F}_0)$ term forms a residual reference and prevents the mixed representation from completely suppressing the standard convolutional branch. This mixer is internal to the common backbone template; it does not select the short or long expert configuration.

A grouped patch embedding with kernel size 16 and stride 8 converts $\mathbf{F}$ into 96-dimensional tokens. Channel shuffle [6] and a pointwise projection restore interaction across convolution groups. Context is then modeled by two residual Mamba blocks [1]:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathrm{Dropout}\big(\mathrm{Mamba}_k(\mathrm{LN}(\mathbf{X}^{(k)}))\big) \,, \qquad k = 0, 1. \tag{7}$$

Each block uses model width 96 and state dimension 24. The selective state-space operator provides content-dependent sequence propagation with linear scaling in token length. Mean and maximum pooling of the encoded tokens are concatenated and passed to a linear classifier. The short configuration inserts relation tokens before the Mamba encoder, whereas the long configuration refines $\mathbf{F}$ before patch embedding and adds a second classifier at the output interface.

### *3.4. Short-sequence relation-aware token augmentation*

Complex-valued convolution provides an effective way to preserve the coupling between the I and Q components [11]. For the short configuration, we pursue a complementary design that makes selected local complex relations explicit before representation learning. This reduces the burden on a learned branch to recover these relations solely from stacked operations and provides direct control over the temporal offsets being modeled. The resulting Polar-Lag module constructs a compact set of amplitude and lag-relation descriptors, followed by a lightweight learned token projection.

To construct these descriptors, let $z_t = i_t + jq_t$ and define its amplitude and power as

$$a_t = \sqrt{i_t^2 + q_t^2 + \epsilon}, \qquad p_t = i_t^2 + q_t^2, \qquad \Delta a_t = a_t - a_{t-1}. \tag{8}$$

For each lag $\ell$ in $\mathcal{S} = \{1, 2, 4, 8, 16\}$, we compute

$$d_{t,\ell} = |z_t - z_{t-\ell}| = \sqrt{(i_t - i_{t-\ell})^2 + (q_t - q_{t-\ell})^2 + \epsilon}, \tag{9}$$

$$c_{t,\ell} = z_t z_{t-\ell}^* = r_{t,\ell} + ju_{t,\ell}, \tag{10}$$

$$r_{t,\ell} = i_t i_{t-\ell} + q_t q_{t-\ell}, \tag{11}$$

$$u_{t,\ell} = q_t i_{t-\ell} - i_t q_{t-\ell}. \tag{12}$$

Samples before the start of the observation are set to zero. The distance $d_{t,\ell}$ measures local displacement in the complex plane, while $r_{t,\ell}$ and $u_{t,\ell}$ retain the in-phase and quadrature components of lagged complex correlation. Under a common phase rotation $z'_t = z_t e^{j\varphi}$, both $d_{t,\ell}$ and $c_{t,\ell}$ remain unchanged. This invariance applies to the relation channels, not to the complete network, which also receives raw I/Q samples.

The final short path concatenates

$$\mathbf{p}_t = \text{concat}\left(i_t, q_t, a_t, p_t, \Delta a_t, \{d_{t,\ell}, r_{t,\ell}, u_{t,\ell}\}_{\ell \in \mathcal{S}}\right). \tag{13}$$

The resulting feature vector contains 20 channels and retains relative phase information through the real and imaginary correlation components.

The features in Eq. (13) are first normalized channel-wise. We denote the resulting sequence by $\mathbf{P}$. A local projection $E_{\text{rel}}$ then applies a standard convolution, a depthwise dilated convolution, and a pointwise convolution. The first operation mixes the heterogeneous relation channels over adjacent samples. The depthwise dilated operation enlarges the local temporal receptive field at limited computational cost [7], and the pointwise operation recombines the resulting channels. A patch projection with kernel size 16 and stride 8 then maps the features to $d = 96$ relation tokens:

$$\mathbf{T}_{\text{rel}} = E_{\text{rel}}(\mathbf{P}). \tag{14}$$

In parallel, the common stem and patch embedding produce the main tokens $\mathbf{T}_{\text{main}}$ with the same temporal resolution. The two streams are integrated before the sequence encoder:

$$\mathbf{T}_{\text{s}} = \mathbf{T}_{\text{main}} + \bar{\alpha}_{\text{s}} \mathbf{T}_{\text{rel}}, \qquad \bar{\alpha}_{\text{s}} = \text{clip}(\alpha_{\text{s}}, 0, \alpha_{\max}), \tag{15}$$

where $\alpha_s$ is a learned scalar and $\alpha_{\max} = 0.25$. The augmented tokens are processed by the short-path state-space sequence encoder and pooling classifier. Integrating at the token interface allows the contextual encoder to model interactions between the backbone and relation tokens, whereas logit-level integration combines only their final decisions. The complete relation-token interface is illustrated in Fig. 3.

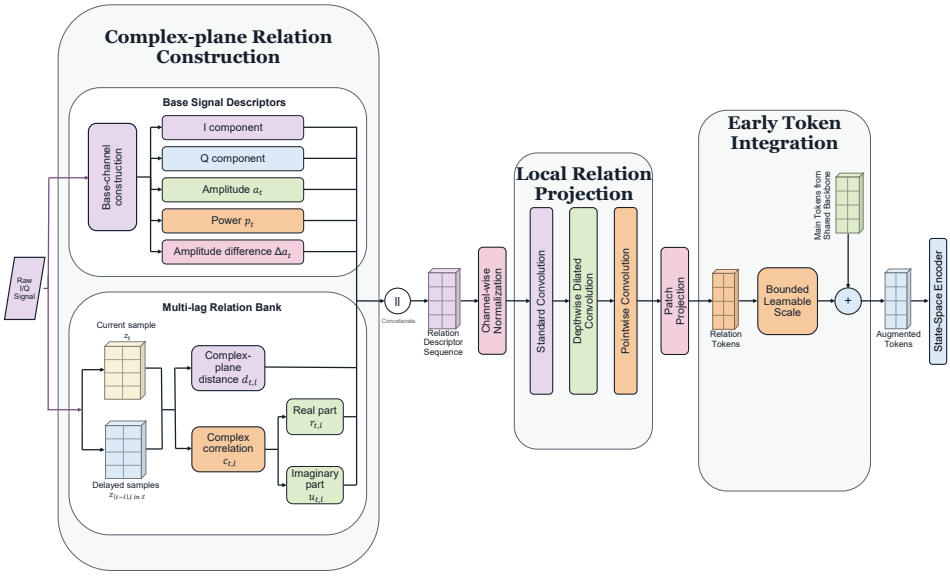


Figure 3: Short-sequence relation-aware token augmentation. Base signal descriptors and multi-lag distance and complex-correlation relations are locally projected into relation tokens, multiplied by a bounded learned scale, and added to the main tokens before state-space encoding.

Ablations with the same three seeds show that removing the relation branch or its distance channels mainly degrades low-SNR recognition, while high-SNR accuracy changes little. The early token interface also performs consistently better than the corresponding late logit interface. These observations motivate the use of the relation tokenizer and early integration in the final short path.

### *3.5. Long-sequence residual refinement and classifier collaboration*

Long observations provide more temporal evidence, but channel impairments can spread unreliable responses over a wider feature sequence. One

approach is to reconstruct a cleaner signal before classification with a denoising autoencoder [25]. Such reconstruction introduces a separate decoder and optimizes signal fidelity in addition to class discrimination. Related residual denoising work also motivates separating a corrective residual from the main task mapping [26]. Following this general principle, the proposed long configuration performs residual refinement directly in the discriminative feature space. It does not reconstruct the waveform or execute a diffusion process.

Let $\mathbf{F} \in \mathbb{R}^{C_f \times L}$ denote the common feature map in Eq. (6). The residual expert first forms a gated value representation

$$\mathbf{V} = \psi_v(\mathbf{F}) \odot \sigma(\psi_i(\mathbf{F})) \,, \tag{16}$$

where $\psi_v$ is a pointwise projection followed by normalization and activation, $\psi_i$ is a pointwise input gate, $\sigma$ denotes the sigmoid function, and $\odot$ is elementwise multiplication. Three depthwise temporal branches process $\mathbf{V}$ with kernel–dilation pairs $(15, 1)$, $(31, 2)$, and $(63, 2)$. Their effective receptive fields are 15, 61, and 125 samples, respectively. The branch outputs are concatenated and fused by a pointwise projection:

$$\mathbf{H} = \psi_f(\text{concat}\,[D_{15,1}(\mathbf{V}), D_{31,2}(\mathbf{V}), D_{63,2}(\mathbf{V})]) \,. \tag{17}$$

The depthwise branches capture complementary temporal extents without the cost of dense large-kernel convolution, while $\psi_f$ restores cross-channel interaction. An output gate and a learned residual scale produce

$$R_{\text{l}}(\mathbf{F}) = \beta_{\text{l}}\, \sigma(\psi_o(\mathbf{F})) \odot \mathbf{H}, \tag{18}$$

where $\beta_{\text{l}}$ is initialized to 0.15. The expert therefore predicts a gated correction to the shared representation rather than a replacement feature map.

The proposed expert uses a sample-dependent residual gate $g_{\text{l}}(\mathbf{F})$ obtained from global average pooling and a small multilayer perceptron. A fixed upper coefficient $\alpha_{\text{l}}$ controls the maximum contribution of the expert. The refined feature map is

$$\widetilde{\mathbf{F}} = \mathbf{F} + \alpha_{\text{l}}\, g_{\text{l}}(\mathbf{F})\, R_{\text{l}}(\mathbf{F}), \tag{19}$$

where $\alpha_{\text{l}} = 0.85$ in the reported experiments, so the sample-dependent path weight $\alpha_{\text{l}} g_{\text{l}}(\mathbf{F})$ is bounded by 0.85.

Two classifiers provide complementary views of $\widetilde{\mathbf{F}}$. The sequence path converts it into patch tokens, applies the state-space encoder, and pools the

encoded sequence to obtain logits $\mathbf{z}_{\text{seq}}$. In parallel, the convolutional classification expert adaptively pools the feature map to 16 temporal positions and applies a compact multilayer perceptron to obtain $\mathbf{z}_{\text{cnn}}$. Their decisions are combined by fixed-average logit fusion:

$$\mathbf{z} = \frac{1}{2}\left(\mathbf{z}_{\text{cnn}} + \mathbf{z}_{\text{seq}}\right). \tag{20}$$

The sequence classifier models ordered contextual dependencies, whereas the CNN expert provides a direct classification view of the refined feature map. Averaging retains their separately formed class evidence without adding a sample-dependent fusion network. This is a logit-level collaboration interface, in contrast to the short configuration's token-level relation interface. Feature concatenation and learned logit gating are evaluated as alternative interfaces in Section 5. The residual–classifier organization is shown in Fig. 4.

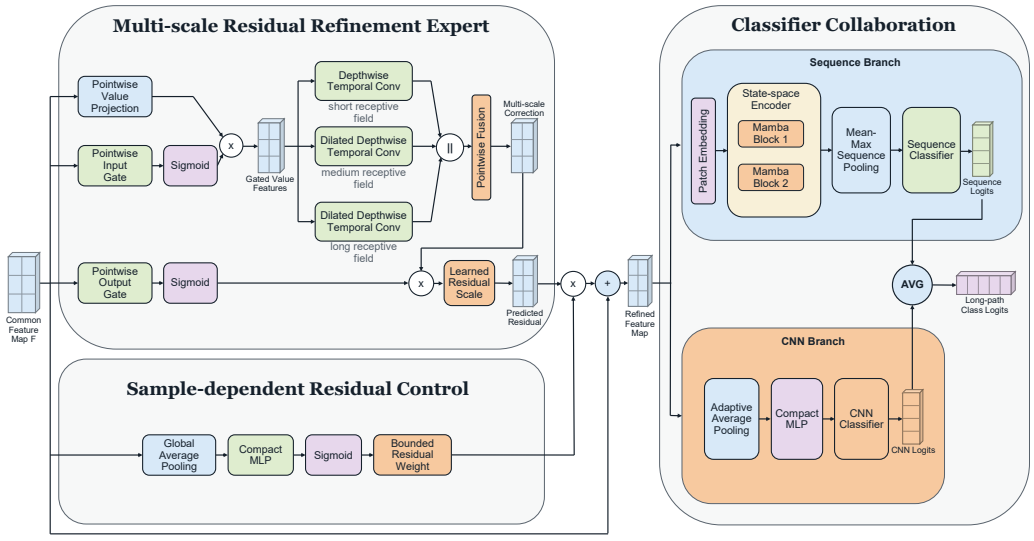


Figure 4: Long-sequence residual–classifier organization. Multi-scale gated residual refinement corrects the common feature map before patch embedding. The state-space and CNN branches then form separate logits that are combined by fixed averaging.

The residual and classification experts therefore act at different stages: the former corrects the feature sequence before contextual encoding, while the latter introduces a complementary decision after encoding. Their con-

tributions and the choice of integration interface are examined separately in the experiments.

### *3.6. Training objectives*

Both configurations are optimized only with the standard cross-entropy classification loss in Eq. (3). All ablations use the same primary supervision.

### *3.7. Computational complexity*

The two configurations are trained and executed separately. We therefore report active-configuration parameters and operations for the short and long configurations separately. The present experiments do not execute both configurations for each input. Complexity is expressed in multiply–accumulate operations (MACs). Convolutional and linear layers are counted from their realized tensor shapes. For each Mamba block, we additionally count the input, state, time-step, output projections, the causal depthwise convolution and the real-valued selective scan. For batch size $B$, token length $T$, inner width $d_{\mathrm{i}}$, and state size $d_{\mathrm{s}}$, the scan contribution is estimated as

$$\mathcal{C}_{\mathrm{scan}} = 9BTd_{\mathrm{i}}d_{\mathrm{s}} + 2BTd_{\mathrm{i}}, \tag{21}$$

where the final term accounts for the skip and output-gating operations. One MAC corresponds approximately to two scalar floating-point operations. This accounting avoids omitting the fused selective-scan kernel from the end-to-end estimate.

## 4. Experimental setup

### *4.1. Datasets and protocols*

Table 1: Dataset and evaluation protocols.

| Dataset | Length | Classes | SNR range | Split (train:validation:test) |
|---|---|---|---|---|
| RML2016.10b | 128 | 10 | −20:2:18 dB | 6:2:2 |
| HisarMod2019 | 1024 | 26 | −20:2:18 dB | 8:2:5 |

The short-observation and long-observation configurations are evaluated on RML2016.10b [3] and HisarMod2019 [4], respectively. The RML2016.10b

files used in this study were obtained from a public Kaggle mirror, while HisarMod2019 was obtained from IEEE DataPort. RML2016.10b contains simulated I/Q observations with 10 modulation classes, whereas HisarMod2019 contains 26 classes generated under multiple fading conditions. We retain the native observation length and the split protocol reported in Table 1 for each formal configuration.

### *4.2. Compared methods*

We compare against representative AMR methods using graph, complex-valued, multimodal, multi-scale, attention, denoising, and language-model representations. STF-GCN constructs adaptive spatial–temporal–frequency graphs [16]. IQFormer and MCANet use multimodal feature collaboration [10, 20], while CPPCNet uses lightweight complex-valued partial pointwise convolution [11]. Recent HisarMod2019 comparisons include MAWDN, DFENet, WACN, and MILAFormer [21, 23, 22, 24]. BioLAMR adapts a pretrained language model with dual-domain signal features [12]. DAE-CNN-BiLSTM combines denoising reconstruction with convolutional and recurrent classifiers [25].

The reported studies use different data partitions and model-selection protocols. Table 2 records the stated protocol for each dataset so that the literature results can be interpreted in context.

### *4.3. Implementation details*

All models are implemented in PyTorch with the CUDA Mamba implementation. Experiments are conducted on an NVIDIA RTX 4070 SUPER GPU with an Intel Core i5-13600KF CPU. Both configurations use the parameter-matched convolutional state-space backbone with model width 96, two Mamba blocks of state dimension 24, and grouped patch embedding with kernel size 16 and stride 8. The short configuration adds the distance and complex-correlation relation channels, whereas the long configuration uses residual refinement and averages the CNN and sequence logits.

For RML2016.10b, all configurations are trained from random initialization with AdamW, learning rate $2.5\times10^{-4}$, weight decay $10^{-4}$, batch size 800, and at most 80 epochs. All short experiments use the same augmentation protocol: circular shifts of at most four samples, multiplicative gain perturbation with standard deviation 0.05, phase rotation bounded by 0.35 radians, and same-class segment substitution with probability 0.35 and segment length 16. The checkpoint with highest validation accuracy is retained with

early-stopping patience 14. For HisarMod2019, models are trained from random initialization with Adam, learning rate $10^{-3}$, weight decay $10^{-4}$, batch size 400, and at most 200 epochs. The learning rate is halved after five validation-loss plateaus, early stopping uses patience 10, and the checkpoint is selected by validation accuracy averaged over samples with SNR $\leq 0$.

The RML2016.10b experiments use training seeds 2051, 2052, and 2053. The HisarMod2019 experiments use seeds 2028, 2029, and 2030 with data split seed 2024. For HisarMod2019, 20% of the official training partition is reserved for validation, giving the 8:2:5 ratio in Table 1; the official test partition remains untouched during model selection.

### *4.4. Evaluation metrics and statistical analysis*

We report overall average accuracy (OAA), mean accuracy for SNR $<$ 0, and mean accuracy for SNR $\geq 0$. We additionally report SNR $\leq -10$ as a descriptive very-low-SNR indicator. Results are summarized as the mean $\pm$ sample standard deviation over three independent training seeds, providing a reproducibility-oriented estimate of run-to-run variation. Same-seed OAA differences are used as descriptive stability checks rather than as formal population-level significance tests.

### *4.5. Native-length cross-configuration protocol*

To verify that the observed gains are associated with the proposed expert designs rather than one configuration being uniformly stronger, we construct a $2 \times 2$ native-length cross-configuration evaluation. On each dataset, both the short-observation and long-observation configurations are evaluated while retaining the native input length and original test protocol, so no signal samples are discarded. The configuration used for that dataset is compared with the cross-applied alternative. This comparison evaluates the suitability of each expert design under the two benchmark conditions. Because dataset identity and observation length remain coupled, it does not isolate a causal effect of length.

### *4.6. Controlled observation-window protocol*

To partially separate temporal support from dataset identity, we conduct an additional study within HisarMod2019. Each native 1024-sample observation is represented by nested center windows of length 128, 256, and 512, together with the unmodified 1024-sample observation. No padding,

resampling, or cross-record concatenation is used. Sample identity, modulation label, SNR, official train/test membership, and the train/validation split are therefore unchanged across lengths. At each length, the short and long configurations are trained independently from random initialization using seeds 2028–2030 and the HisarMod2019 optimization protocol in Section 4. The study measures how each configuration uses increasing temporal support within one data source.

## 5. Results

### *5.1. Comparison with literature-reported methods*

Table 2 places the proposed configurations alongside representative recent results. Values for competing methods are taken from their original publications, whereas the proposed results are the mean and sample standard deviation over the three seeds defined in Section 4.

Table 2: Reported OAA (%), proposed SNR-stratified results, and dataset-specific protocols. Ratios use train:validation:test order unless marked as train/validation or train/test; a dash denotes an unreported result.

| Method | RML2016.10b OAA | HisarMod2019 OAA | RML2016.10b protocol | HisarMod2019 protocol |
|---|---|---|---|---|
| STF-GCN [16] | 66.04 | – | 6:2:2 | – |
| IQFormer [10] | 65.65 | 76.32 | 6:2:2 | 8:2:5 |
| CPPCNet [11] | 66.38 | 83.50 | 8:2 train/val | 8:2 train/val |
| BioLAMR [12] | $67.43 \pm 0.27$ | – | 8:1:1 | – |
| MCANet [20] | 66.53 | 76.58 | 7:2:1 | 7:2:1 |
| MAWDN [21] | – | 74.40 | – | 8:2:5 |
| MILAFormer [24] | – | 77.45 | – | 8:2:5 |
| DFENet [23] | – | 82.76 | – | 2:1 train/test |
| WACN [22] | 64.70 | 95.54 | Not stated | Not stated |
| DAE-CNN-BiLSTM [25] | – | 86.93 | – | 8:2:5 |
| Proposed | $67.28 \pm 0.14$ | $87.19 \pm 0.77$ | 6:2:2 | 8:2:5 |
| *Proposed configurations: SNR-stratified results* | | | | |
| Configuration | OAA | SNR $\leq -10$ | SNR $< 0$ | SNR $\geq 0$ |
| RML2016.10b, short | $67.279 \pm 0.135$ | $23.725 \pm 0.602$ | $41.121 \pm 0.267$ | $93.436 \pm 0.033$ |
| HisarMod2019, long | $87.186 \pm 0.773$ | $71.126 \pm 1.710$ | $75.577 \pm 1.484$ | $98.794 \pm 0.062$ |

Under the explicitly reported 6:2:2 protocol on RML2016.10b, the proposed short configuration achieves the best OAA among the comparable entries listed in Table 2. Its result is also within 0.15 percentage points of the highest reported BioLAMR result, although BioLAMR uses a different data partition. On HisarMod2019, the proposed long configuration achieves the highest OAA among the listed methods that explicitly report the same 8:2:5 protocol, exceeding DAE-CNN-BiLSTM by 0.26 percentage

points. These comparisons indicate competitive state-of-the-art performance under protocol-matched settings, rather than a protocol-independent ranking across all published results. The lower panel of Table 2 gives the proposed SNR-stratified results.

### *5.2. Short-sequence ablation*

Table 3 evaluates the relation descriptors and their integration interface on RML2016.10b. Setting the relation gate to zero removes the complete relation-token update while preserving the backbone and the main sequence classifier. It is therefore a control for the contribution of the relation branch as a whole, rather than an ablation of distance channels alone. The late-fusion variant processes the same relation descriptors with a separate classification head and adds its gated logits after sequence classification, so it tests the integration location while retaining the relation information.

Table 3: Short-configuration ablation on RML2016.10b (%, mean ± sample standard deviation over seeds 2051–2053).

| Variant | OAA | SNR $\leq -10$ | SNR $< 0$ | SNR $\geq 0$ | $\Delta$OAA |
|---|---|---|---|---|---|
| Full short-observation configuration | $67.279 \pm 0.135$ | $23.725 \pm 0.602$ | $41.121 \pm 0.267$ | $93.436 \pm 0.033$ | – |
| w/o correlation features | $66.859 \pm 1.180$ | $21.944 \pm 3.164$ | $40.267 \pm 2.330$ | $93.450 \pm 0.032$ | $-0.420$ |
| w/o distance features | $65.224 \pm 0.245$ | $17.526 \pm 0.067$ | $36.968 \pm 0.452$ | $93.480 \pm 0.044$ | $-2.054$ |
| Relation gate $= 0$ | $65.250 \pm 0.223$ | $17.398 \pm 0.069$ | $37.054 \pm 0.388$ | $93.446 \pm 0.071$ | $-2.029$ |
| Late logit fusion | $65.749 \pm 0.163$ | $20.089 \pm 0.202$ | $38.304 \pm 0.333$ | $93.193 \pm 0.060$ | $-1.530$ |

Relative to the zero-gate variant, the full configuration improves OAA by 2.029 percentage points and SNR $< 0$ accuracy by 4.067 points, while the SNR $\geq 0$ accuracies differ by only 0.010 points. Removing distance features produces a similar OAA loss of 2.054 points. The nearly identical losses of the w/o-distance and zero-gate controls indicate that distance accounts for most of the measured relation-branch gain. Correlation features produce a modest mean difference relative to run-to-run variation; they are therefore retained as a complementary relation descriptor without attributing an independently validated accuracy gain to this component. Early token augmentation exceeds late logit fusion by 1.530 points, with most of the difference again occurring below 0 dB. This difference is consistent with the two interfaces exposing the relation information at different stages: early addition lets the state-space encoder contextualize relation and backbone tokens jointly, whereas late fusion can only combine the separately formed final decisions.

### *5.3. Long-sequence expert and interface analysis*

Table 4 evaluates the long-configuration classifier interface and expert contributions on HisarMod2019. All variants retain the convolutional state-space backbone and the same classification objective. Removing residual refinement leaves the two classifier branches and their decision interface unchanged. The learned-gate variant uses a feature-dependent coefficient instead of equal logit weights, while feature concatenation combines the sequence and convolutional features before a classifier. CNN-only retains the convolutional classifier, sequence-only retains the sequence classifier, and the both-experts-off control removes both the residual and CNN experts.

Table 4: Long-configuration expert and interface analysis on HisarMod2019 (%, mean ± sample standard deviation over seeds 2028–2030).

| Variant | OAA | SNR $\leq -10$ | SNR $< 0$ | SNR $\geq 0$ | $\Delta$OAA |
|---|---|---|---|---|---|
| Equal-weight logit averaging (formal) | $87.186 \pm 0.773$ | $71.126 \pm 1.710$ | $75.577 \pm 1.484$ | $98.794 \pm 0.062$ | – |
| w/o residual refinement | $79.893 \pm 0.365$ | $55.963 \pm 0.569$ | $61.924 \pm 0.547$ | $97.862 \pm 0.197$ | $-7.293$ |
| Learned logit gate | $86.963 \pm 0.622$ | $70.640 \pm 1.316$ | $75.131 \pm 1.213$ | $98.795 \pm 0.076$ | $-0.223$ |
| Feature concatenation | $85.186 \pm 0.258$ | $67.209 \pm 0.686$ | $71.932 \pm 0.523$ | $98.440 \pm 0.017$ | $-2.000$ |
| CNN classifier only | $86.013 \pm 0.319$ | $68.801 \pm 0.811$ | $73.410 \pm 0.670$ | $98.617 \pm 0.039$ | $-1.173$ |
| Sequence classifier only | $76.566 \pm 0.119$ | $56.938 \pm 0.040$ | $60.216 \pm 0.039$ | $92.915 \pm 0.215$ | $-10.620$ |
| Residual and CNN experts off | $72.770 \pm 0.742$ | $53.410 \pm 0.719$ | $56.103 \pm 0.660$ | $89.436 \pm 0.825$ | $-14.416$ |

Removing residual refinement while retaining equal-weight decision collaboration lowers OAA by 7.293 points and SNR $< 0$ accuracy by 13.653 points, confirming the importance of residual refinement. Equal-weight averaging and learned gating achieve comparable accuracy, while both outperform feature concatenation and the separately optimized single-branch controls. Since learned gating provides no repeatable gain while introducing an additional sample-dependent fusion module, fixed averaging is adopted as the simpler decision-level interface. Feature concatenation is 2.000 points lower, which favors collaboration between separately formed decisions over joint feature compression. The CNN classifier is stronger than the sequence classifier individually; however, adding the sequence decision to the CNN branch improves OAA by 1.173 points and SNR $< 0$ accuracy by 2.167 points. Sequence-only and both-experts-off variants produce larger losses.

Figure 5 resolves the aggregate ablation results by SNR. For the short configuration, the gains from relation-aware token augmentation and its distance descriptor occur predominantly below 0 dB, while the curves nearly converge at nonnegative SNRs. For the long configuration, residual refinement yields its largest advantage in the negative-SNR region, and collaboration with the

sequence classifier provides a smaller gain with the same concentration. The improvements are therefore associated mainly with difficult noise conditions rather than a uniform upward shift across SNRs. For clarity, Fig. 5 visualizes only mechanism-representative variants; the complete set of ablations is reported in Tables 3 and 4.

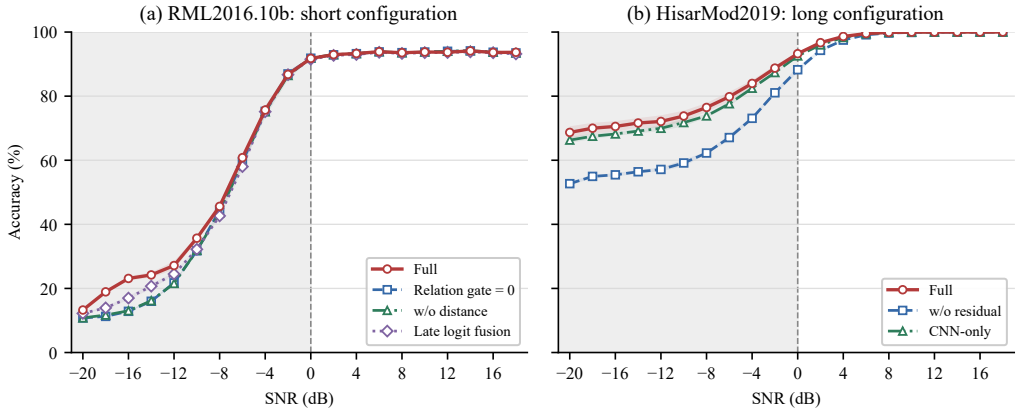


Figure 5: Per-SNR accuracy of representative short- and long-configuration ablations. Curves show the mean over three seeds, shaded bands denote one sample standard deviation, and the gray background marks negative SNRs.

### *5.4. Long-configuration mechanism diagnostics*

We further inspect the three selected formal checkpoints without retraining. For the residual interface, Table 5 reports the bounded coefficient $\alpha_1 g_1(\mathbf{F})$, its within-run variation across test samples, and the relative correction magnitude $\|\alpha_1 g_1(\mathbf{F})\mathbf{R}\|_2/\|\mathbf{F}\|_2$. Each entry is the mean $\pm$ sample standard deviation across the three seed-level statistics.

Table 5: Residual-interface diagnostics on the HisarMod2019 official test set. The coefficient is bounded above by 0.85. "Within-run SD" measures sample-to-sample coefficient variation within each checkpoint.

| Region | Residual coefficient | Within-run SD | Relative correction norm |
|---|---|---|---|
| All SNRs | $0.642 \pm 0.033$ | $0.157 \pm 0.017$ | $1.194 \pm 0.096$ |
| SNR $< 0$ | $0.678 \pm 0.030$ | $0.132 \pm 0.020$ | $1.283 \pm 0.094$ |
| SNR $\geq 0$ | $0.606 \pm 0.035$ | $0.171 \pm 0.017$ | $1.106 \pm 0.097$ |

The residual coefficient is larger on average below 0 dB, and the relative correction norm changes in the same direction. The nonzero within-run deviations show that the gate is not a fixed global multiplier. These statistics support sample-dependent residual adjustment and a stronger average intervention in the difficult SNR region. They do not imply that the gate is an explicit SNR estimator, because its input is the learned feature map and the coefficient distributions overlap across the two regions.

Table 6 examines the two logit branches inside the same jointly trained formal checkpoint. "Fusion-only correct" denotes samples for which the averaged logits predict the correct class although neither branch is individually top-1 correct. The margin correlation is the Pearson correlation between the two branches' true-class margins, where a margin is the true-class logit minus the largest competing logit.

Table 6: Score-level complementarity of the jointly trained sequence and CNN classifiers on HisarMod2019 (%, except correlation; mean ± sample standard deviation over seeds 2028–2030).

| Region | Sequence top-1 | CNN top-1 | Fused top-1 | Disagreement | Fusion-only correct | Margin correlation |
|---|---|---|---|---|---|---|
| All SNRs | $60.537 \pm 0.718$ | $38.590 \pm 1.370$ | $87.186 \pm 0.773$ | $84.238 \pm 0.537$ | $6.524 \pm 0.287$ | $-0.439 \pm 0.015$ |
| SNR $< 0$ | $55.175 \pm 0.330$ | $24.382 \pm 0.886$ | $75.577 \pm 1.484$ | $91.468 \pm 0.103$ | $9.564 \pm 0.427$ | $-0.324 \pm 0.021$ |
| SNR $\geq 0$ | $65.900 \pm 1.360$ | $52.798 \pm 1.858$ | $98.794 \pm 0.062$ | $77.007 \pm 0.973$ | $3.485 \pm 0.189$ | $-0.676 \pm 0.004$ |

The high prediction disagreement and negative margin correlation show that the two branches form different score residuals. Averaging can therefore recover decisions even when both separate argmax outputs are wrong, with the largest fusion-only recovery rate occurring below 0 dB. The extracted branch accuracies in Table 6 should not be equated with the separately optimized CNN-only and sequence-only controls in Table 4: the formal model applies supervision only to the averaged logits, so its two branch scores are free to co-adapt. Together, the retrained controls and the within-checkpoint diagnostic support decision-level collaboration while avoiding a claim that either jointly trained branch is a calibrated standalone classifier.

### 5.5. Native-length cross-configuration comparison

Table 7: Native and cross-applied configurations at the original observation length of each dataset (OAA in %, mean ± sample standard deviation). Intervals are computed from three same-seed OAA differences.

| Dataset | Native configuration | Cross-applied configuration | Native advantage | Exploratory 95% interval |
|---|---|---|---|---|
| RML2016.10b ($L = 128$) | Short: $67.279 \pm 0.135$ | Long: $65.326 \pm 0.067$ | +1.952 | [1.726, 2.179] |
| HisarMod2019 ($L = 1024$) | Long: $87.186 \pm 0.773$ | Short: $67.023 \pm 1.393$ | +20.163 | [16.074, 24.252] |

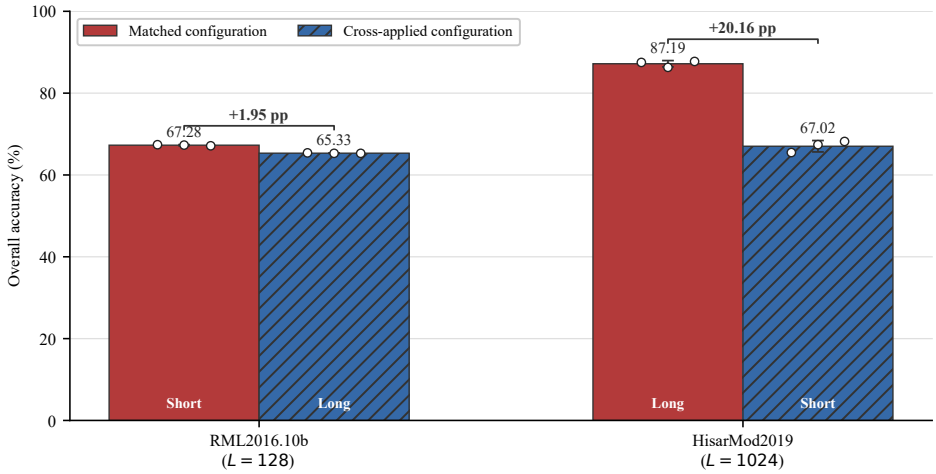


Figure 6: Native and cross-applied configuration performance at the original observation length of RML2016.10b and HisarMod2019. Bars and error bars denote mean OAA and one sample standard deviation over three seeds, respectively; markers indicate individual seeds. Numbers above the brackets report the native-configuration advantage in percentage points.

As summarized in Table 7 and visualized in Fig. 6, the native configuration wins in all six same-seed comparisons. On RML2016.10b, the short-observation configuration has a mean advantage of 1.952 points over the cross-applied long configuration. On HisarMod2019, the long-observation configuration has a larger mean advantage of 20.163 points over the cross-applied short configuration. These results show that neither configuration is uniformly stronger and support the use of different expert designs under the two evaluated benchmark conditions. Dataset identity and observation length nevertheless remain coupled in this comparison.

### *5.6. Controlled observation-window comparison*

Table 8 compares the two configurations using nested windows from the same HisarMod2019 observations. Long–Short differences are computed seed by seed before aggregation.

Table 8: Controlled observation-window comparison on HisarMod2019 (OAA in %, mean ± sample standard deviation over seeds 2028–2030).

| Length | Short | Long | Long–Short |
|---|---|---|---|
| 128 | $61.783 \pm 0.952$ | $71.527 \pm 0.233$ | $9.744 \pm 0.719$ |
| 256 | $65.090 \pm 0.646$ | $76.970 \pm 0.228$ | $11.881 \pm 0.482$ |
| 512 | $67.433 \pm 0.310$ | $81.679 \pm 0.527$ | $14.247 \pm 0.225$ |
| 1024 | $67.023 \pm 1.393$ | $87.186 \pm 0.773$ | $20.163 \pm 1.646$ |

The long configuration improves by 15.659 points from 128 to 1024 samples, whereas the short configuration improves by 5.240 points and saturates after 512 samples. More importantly, the Long–Short advantage increases monotonically with window length for every seed. This same-source result shows that the residual–classifier configuration uses additional temporal support more effectively on HisarMod2019. At the same nominal length of 128 samples, the preferred configuration differs between HisarMod2019 and RML2016.10b, indicating that temporal support interacts with the modulation and channel distributions of the data source when determining the effective expert design.

### *5.7. Paired stability analysis*

Table 9 reports the same-seed OAA differences between each formal configuration and its ablated counterpart. Positive values indicate an advantage for the formal configuration. The short-path columns correspond to seeds 2051–2053, and the long-path columns correspond to seeds 2028–2030. This table exposes the direction and magnitude of run-to-run differences directly; it is used as a reproducibility-oriented stability check rather than as a formal significance test.

Table 9: Same-seed paired OAA differences (percentage points) for the short and long configurations. Positive values favor the corresponding formal configuration.

| Comparison | Seed 1 | Seed 2 | Seed 3 | Mean ΔOAA |
|---|---|---|---|---|
| *Short configuration (seeds 2051, 2052, 2053)* | | | | |
| Full − w/o correlation | −0.135 | −0.242 | +1.637 | +0.420 |
| Full − w/o distance | +2.300 | +1.795 | +2.067 | +2.054 |
| Full − relation gate = 0 | +2.292 | +1.795 | +1.999 | +2.029 |
| Full − late fusion | +1.487 | +1.557 | +1.546 | +1.530 |
| *Long configuration (seeds 2028, 2029, 2030)* | | | | |
| Full − w/o residual refinement | +8.035 | +6.209 | +7.635 | +7.293 |
| Full − learned gate | −0.143 | −0.494 | +1.307 | +0.223 |
| Full − feature concatenation | +2.366 | +1.350 | +2.283 | +2.000 |
| Full − CNN-only | +1.173 | +0.610 | +1.735 | +1.173 |
| Full − sequence-only | +10.909 | +9.638 | +11.312 | +10.620 |
| Full − both experts off | +15.135 | +13.992 | +14.121 | +14.416 |

The paired analysis is used as a descriptive stability check rather than as a claim of definitive significance. Across the three matched seeds, the short full configuration consistently outperforms the zero-gate, w/o-distance, and late-fusion variants. The long full configuration likewise consistently outperforms the w/o-residual, feature-concatenation, sequence-only, and both-experts-off variants. The two native-length cross-configuration comparisons also retain the same direction for every seed. The correlation-removal and learned-gating controls instead quantify smaller changes relative to the main component and interface effects. The CNN-only control serves a different purpose: its larger gap from the complete long configuration supports complementary evidence from the state-space classifier, especially at low SNR.

### *5.8. Accuracy–complexity trade-off*

Only the active configuration is instantiated for an experiment. Table 10 reports the complete learned-operator estimate defined in Eq. (21), rather than a tracer result that omits the fused scan. Latency is the median of 300 float32 batch-1 forward passes after 100 stabilization passes, measured with CUDA events in PyTorch 2.4.1. Peak allocated memory includes model parameters, the input, and inference activations. Both runtime quantities are measured on the RTX 4070 SUPER specified in Section 4.

Table 10: Active-configuration complexity and batch-1 inference cost. FLOPs use the approximation 1 MAC $\approx$ 2 scalar FLOPs.

| Configuration | Length | Parameters (M) | MACs (M) | Approx. FLOPs (M) | Latency (ms) | Peak memory (MiB) | OAA (%) |
|---|---|---|---|---|---|---|---|
| Short-observation configuration | 128 | 0.330 | 11.13 | 22.25 | 2.85 | 10.51 | $67.279 \pm 0.135$ |
| Long-observation configuration | 1024 | 0.530 | 103.08 | 206.16 | 2.37 | 12.44 | $87.186 \pm 0.773$ |

The long configuration requires approximately 9.3$\times$ more MACs because of its longer feature sequence and residual branches. Its measured batch-1 latency is nevertheless slightly lower in this GPU setting. The short relation tokenizer contains several fine-grained descriptor and convolution kernels, whereas the longer operators expose more device parallelism; consequently, hardware latency is not proportional to the analytical MAC count. The latency and memory values are implementation- and hardware-specific, while parameter count and MACs provide the more portable comparison.

## 6. Discussion

### *6.1. Representation bottlenecks and expert design*

The ablations distinguish two design axes: what information an expert contributes and where that information enters the classifier. On the short RML2016.10b benchmark, explicit lagged distance contributes most of the relation branch's gain, and placing relation features before contextual encoding is more effective than combining a separate relation decision at the output. Correlation is retained as a complementary descriptor, while the distance channels provide the dominant measured gain. The near-constant nonnegative-SNR accuracy shows that the main relation-branch benefit is not a uniform increase in classifier capacity; it is concentrated where the available evidence is noisy.

On the long benchmark, multi-scale residual refinement acts on the feature map before tokenization, whereas the CNN expert collaborates with the sequence classifier at the decision level. Removing residual refinement produces a clear loss, concentrated below 0 dB, which supports feature correction as a distinct component of the long configuration. Fixed averaging performs comparably to learned gating and clearly exceeds feature concatenation. Because the learned gate adds a sample-dependent fusion module without a repeatable gain, the complete long configuration uses fixed averaging. When separately optimized as single-branch controls, the CNN-only model is stronger than the sequence-only model, and the complete model

exceeds both. Within the jointly trained formal checkpoints, the larger residual coefficient at negative SNRs and the negatively correlated classifier margins further support adaptive feature correction and score-level collaboration. The long design thus separates feature correction from decision collaboration without requiring a learned fusion gate.

Cross-configuration results complement the component ablations. Across all seeds, the short configuration wins on RML2016.10b, while the long configuration wins on HisarMod2019. This consistency supports relation-token augmentation under limited observations and residual–classifier collaboration with extended support. The larger HisarMod2019 margin suggests that the latter uses richer temporal evidence more effectively. Within HisarMod2019, controlled windows further show that the Long–Short gap grows monotonically with temporal support for every seed. The long configuration thus scales more effectively with observation span on this data source. Its advantage at all four lengths, including 128 samples, also shows why nominal length alone is insufficient for assigning an expert configuration.

### *6.2. Scope and limitations*

The two datasets differ in more than observation length. Their modulation sets, channel models, data distributions, and evaluation protocols differ simultaneously. The controlled HisarMod2019 study keeps sample identity and labels fixed while varying the retained temporal support, reducing this confounding and demonstrating a stable within-source trend. However, its nested windows are derived from fixed 1024-sample recordings rather than from independent acquisitions made at several durations. The evidence therefore shows more effective use of extended support by the long configuration on this data source; it does not isolate a causal rule based on sequence length alone.

The literature comparison in Table 2 uses reported results rather than same-codebase reimplementations; protocol and model-selection differences therefore limit strict cross-paper ranking. At the descriptor level, distance accounts for the dominant measured gain, whereas correlation is retained as a complementary descriptor.

## 7. Conclusion

This work introduced a unified backbone–expert framework for representation bottlenecks encountered under different AMR benchmark condi-

tions. A common convolutional state-space backbone topology is paired with two complementary expert designs and integration interfaces. The short configuration adds lag-aware distance and complex-correlation tokens before sequence encoding, whereas the long configuration combines long-receptive-field residual refinement with sequence–CNN decision collaboration. The configurations achieved $67.28 \pm 0.14\%$ OAA on RML2016.10b and $87.19 \pm 0.77\%$ on HisarMod2019.

The short ablations show that the relation branch and its token interface mainly improve negative-SNR recognition, with distance providing the dominant measured contribution. On the long benchmark, residual refinement produces a clear gain, decision-level collaboration exceeds feature concatenation, and the CNN and sequence classifiers provide complementary evidence. The native-length cross-configuration comparison favors the configured design on both datasets, while the controlled HisarMod2019 study shows that the long configuration's advantage grows consistently with temporal support. Taken together, the results support a topology-unified design in which relation-token augmentation and residual–classifier collaboration address different representation bottlenecks.

## CRediT authorship contribution statement

Zhixiang Deng: Methodology, Software, Investigation, Validation, Formal analysis, Visualization, Writing–original draft. Houbiao Li: Conceptualization, Supervision, Writing–review and editing. Zongyong Cui: Validation, Writing–review and editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The RML2016.10b files used in this study are available from a public mirror on the Kaggle dataset page. HisarMod2019 is available from the IEEE DataPort dataset page [3, 4]. The datasets are not redistributed by the authors. Code and experiment configurations will be made publicly available upon publication.

## References


[1] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, in: First Conference on Language Modeling (COLM), 2024.

[2] T. J. O'Shea, J. Corgan, T. C. Clancy, Convolutional radio modulation recognition networks, in: Engineering Applications of Neural Networks, Springer, 2016, pp. 213–226.

[3] T. J. O'Shea, N. West, Radio machine learning dataset generation with GNU Radio, in: Proceedings of the GNU Radio Conference, vol. 1, 2016, pp. 1–6.

[4] K. Tekbiyik, A. R. Ekti, A. Görçin, G. K. Kurt, C. Keçeci, Robust and fast automatic modulation classification with CNN under multipath fading channels, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), IEEE, 2020, pp. 1–6. doi:10.1109/VTC2020-Spring48590.2020.9128408. doi:10.1007/978-3-319-44188-7_16.

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, B. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[6] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6848–6856. doi:10.1109/CVPR.2018.00716.

[7] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: International Conference on Learning Representations (ICLR), 2016. arXiv:1511.07122.

[8] D. Hong, Z. Zhang, X. Xu, Automatic modulation classification using recurrent neural networks, in: 2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE, 2017.

[9] N. E. West, T. J. O'Shea, Deep architectures for modulation recognition, arXiv preprint arXiv:1703.09197 (2017).

[10] M. Shao, D. Li, S. Hong, J. Qi, H. Sun, IQFormer: A novel Transformer-based model with multi-modality fusion for automatic modulation recognition, IEEE Transactions on Cognitive Communications and Networking 11 (3) (2025) 1623–1634. doi:10.1109/TCCN.2024.3485118.

[11] G. Xin, Z. Cai, Y. Lou, C. Wang, CPPCNet: High-performance and low-complexity automatic modulation classification for resource-limited IoT communication, IEEE Internet of Things Journal 12 (20) (2025) 43842–43854. doi:10.1109/JIOT.2025.3598976.

[12] Y. Mao, W. Xu, J. Sang, H. Liu, BioLAMR: A biomimetically inspired large language model adaptation framework for automatic modulation recognition, Biomimetics 11 (4) (2026) 288. doi:10.3390/biomimetics11040288.

[13] J. Gao, Q. Cao, Y. Chen, MoE-AMC: Enhancing automatic modulation classification using mixture-of-experts, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2026, pp. 2186–2190. doi:10.1109/ICASSP55912.2026.11461635.

[14] F. Zhang, C. Luo, J. Xu, Y. Luo, F.-C. Zheng, Deep learning based automatic modulation recognition: Models, datasets, and challenges, Digital Signal Processing 129 (2022) 103650. doi:10.1016/j.dsp.2022.103650.

[15] J. Zhang, S. An, F. Meng, Q. Liu, MST: A multi-scale Transformer framework with cross-scale token fusion for automatic modulation recognition, IEEE Wireless Communications Letters 14 (12) (2025) 4112–4116. doi:10.1109/LWC.2025.3614235.

[16] M. Shao, Z. Fu, D. Li, F. Zhang, Y. Cai, S. Hong, L. Cao, Y. Peng, J. Qi, STF-GCN: A multi-domain graph convolution network method for automatic modulation recognition via adaptive correlation, arXiv preprint arXiv:2504.08504 (2025).

[17] Y. Zhao, S. Cao, X. Wang, M. Cheng, Z. Liu, Complex-value automatic modulation recognition via a dual-channel Transformer, in: 2025 International Joint Conference on Neural Networks (IJCNN), IEEE, 2025. doi:10.1109/IJCNN64981.2025.11228074.

[18] L. Guo, D. Wu, W. Yang, J. Liu, K. Cheng, J. Tu, Adaptive multimodal modulation recognition with feature enhancement under low SNR conditions, IEEE Transactions on Cognitive Communications and Networking 12 (2026) 3235–3249. doi:10.1109/TCCN.2025.3626423.

[19] Y. M. Karadag, I. Talaz, I. G. Dino, S. Kalkan, ms-Mamba: Multi-scale Mamba for time-series forecasting, Neurocomputing 680 (2026) 133226. doi:10.1016/j.neucom.2026.133226.

[20] W. Jiang, H. Yang, X. Lu, M. Wang, H. Sun, J. Zhang, MCANet: A coherent multimodal collaborative attention network for advanced modulation recognition in adverse noisy environments, arXiv preprint arXiv:2510.18336 (2025).

[21] X. Qin, W. Jiang, G. Gui, D. Li, D. Niyato, J. Lu, Multilevel adaptive wavelet decomposition network-based automatic modulation recognition: Exploiting time-frequency multiscale correlations, IEEE Transactions on Cognitive Communications and Networking 11 (5) (2025) 3218–3231. doi:10.1109/TCCN.2025.3535738.

[22] Y. Feng, K. Peng, J. Wei, Z. Tang, Window attention convolution network (WACN): A local self-attention automatic modulation recognition method, IEEE Transactions on Cognitive Communications and Networking 11 (3) (2025) 1597–1608. doi:10.1109/TCCN.2024.3462905.

[23] H.-K. Le, V.-P. Hoang, V.-S. Doan, DFENet: A diverse feature extraction neural network for improving automatic modulation classification accuracy in wireless communication systems, PLOS ONE 21 (1) (2026) e0341020. doi:10.1371/journal.pone.0341020.

[24] J. Zhao, Y. Sun, Y. Chen, X. Dong, G. Song, N. Jin, D. Quan, MILAFormer: A multi-stage hybrid deep learning architecture for robust radio signal modulation recognition, IEEE Transactions on Cognitive Communications and Networking 12 (2026) 7574–7588. doi:10.1109/TCCN.2026.3683207.

[25] F. Long, J. Zhao, N. Zhou, X. Ni, S. Chen, Y. Zhao, Deep learning-based modulation recognition for communication signals, in: Proc. IEEE ICEMI, 2025, pp. 498–502. doi:10.1109/ICEMI66537.2025.11306508.

[26] Z. Lin, J. Hou, H. Xia, X. Xie, F. Wang, Y. Zhou, W. Wang, J. Liu, L. Qu, Decoupled residual denoising diffusion models for unified and data efficient image-to-image translation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2026, pp. 35967–35977.

[27] W. Ma, C. Wang, E. Zhang, DWMTN: A dynamic Mamba–Transformer network with adaptive feature fusion for automatic modulation recognition, IEEE Signal Processing Letters 32 (2025) 4069–4073. doi:10.1109/LSP.2025.3622526.

[28] E. Zhu, R. Li, Y. Ren, J. Lu, L. Tang, T. Huang, Modulation recognition algorithm for long-sequence, high-order modulated signals based on Mamba architecture, Applied Sciences 15 (17) (2025) 9805. doi:10.3390/app15179805.

[29] Z. Zhang, Z. Wei, S. Han, Y. Yang, J. Zhan, W. Wu, H. You, C. Li, LM-GDMAF: A lightweight Mamba multimodal fusion algorithm for small-sample modulation recognition, Algorithms 19 (7) (2026) 532. doi:10.3390/a19070532.