Journal of Intelligent Computing and Networking
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
ISSN 3107-2798 (online)
E-mail: jicn.office@ffspub.com

Article

RGF-Net: A Dual-Stream Spatiotemporal Fusion Network for Robust Automatic Modulation Recognition in Cognitive Radio

Fangning Shi[1], Yiheng Zhang[2], Yongkang Gong[3,†]
[1]*China Telecom Research Institute, Beijing 100191, China*
[2]*School of Electronics and Information Engineering, Beihang University, Beijing 100191, China*
[3]*School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore*
[†]*E-mail: eee-ykgong@ntu.edu.sg*

Received: April 1, 2026 / Revised: April 19, 2026 / Accepted: April 19, 2026 / Published online: May 6, 2026

Abstract: Automatic Modulation Recognition (AMR) is an enabling component of cognitive radio because it helps secondary users infer incumbent transmission formats under limited prior information. Existing single-stream deep networks often emphasize either spatial constellation structure or temporal evolution, which can reduce robustness under noisy channel conditions. This paper presents RGF-Net, a dual-stream framework that processes the same raw I/Q sequence with a ResNet branch and a GRU branch, then combines their class-posterior outputs through a learnable decision-fusion weight. The methodological contribution therefore lies in the parallel decomposition and end-to-end calibration of complementary spatial and temporal classifiers, rather than in proposing a new backbone block. Experiments on the synthetic RadioML2018.10a benchmark show that, among the compared models, RGF-Net achieves the highest accuracy across the evaluated SNR range and improves the 0 dB accuracy by 12.2 percentage points over the strongest single-branch baseline. These results indicate that the proposed fusion design is effective on this benchmark, while evaluation on additional real-world datasets remains necessary.

Keywords: Automatic Modulation Recognition; Deep Learning; Model Fusion; Residual Networks; Decision Fusion
https://doi.org/10.64509/jicn.21.95

1 Introduction

The rapid growth of wireless traffic in 5G, Internet-of-Things, and emerging 6G scenarios has intensified pressure on spectrum utilization [1, 2]. Cognitive radio addresses this pressure by allowing secondary users to sense and opportunistically access underutilized bands [3, 4]. Within this sensing chain, Automatic Modulation Recognition (AMR) helps characterize incumbent signals without requiring full prior knowledge of the channel or waveform parameters [5–7]. Reliable AMR is therefore useful for coexistence management, waveform adaptation, and non-cooperative signal analysis in dynamic radio environments [8].

Classical AMR methods are commonly divided into likelihood-based and feature-based families. Likelihood-based methods can be statistically powerful but are computationally demanding and rely on signal and channel assumptions that are difficult to satisfy in practical non-cooperative settings [9]. Feature-based methods reduce that burden by extracting handcrafted descriptors such as cumulants, cyclostationary statistics, or wavelet features, yet their performance depends heavily on the quality of the predefined feature set and often degrades at low Signal-to-Noise Ratio (SNR) [10]. Deep learning mitigates this handcrafting burden by learning task-specific representations directly from raw In-phase/Quadrature (I/Q) samples. In particular, Convolutional Neural Network (CNN)-style models are effective at capturing local spatial structure in the I/Q plane [11, 12], whereas recurrent models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are better suited to temporal evolution and sequential dependencies [13, 14].

Prior hybrid AMR architectures usually follow one of two routes. The first is a sequential CNN-RNN pipeline, exemplified by Convolutional Long Short-Term Memory (CLDNN)-style processing, in which temporal modeling is applied to features already transformed by a convolutional frontend [15].

† Corresponding author: Yongkang Gong
*Academic Editor: Chunxiao Jiang

The second is a generic fusion framework that concatenates features or averages classifier outputs [16–18]. These strategies are useful references, but they do not fully answer whether spatial and temporal evidence should be learned in parallel from the same raw signal and then calibrated with a lightweight trainable fusion rule. Our work is positioned in this gap: it uses standard backbones, but organizes them as two parallel specialists and fuses their posterior probabilities through a jointly learned scalar weight.

This design choice is intentionally conservative. Rather than introducing a high-capacity fusion subnetwork, we investigate whether a low-parameter decision-level calibration is sufficient to capture the overall reliability difference between the spatial and temporal branches on a standard AMR benchmark. The resulting method is easy to implement, transparent to analyze, and directly comparable with sequential and feature-concatenation baselines. The principal contributions of this work are summarized as follows:

- We formulate AMR as a parallel dual-stream problem in which a ResNet branch and a GRU branch process the same raw I/Q observation independently, preserving their respective spatial and temporal inductive biases instead of forcing one branch to inherit the representation bottleneck of the other.
- We adopt a learnable decision-level fusion rule that combines the branch posteriors with a single trainable scalar. Because the two branches already output class-conditioned probabilities, this design offers a low-variance and interpretable way to calibrate their relative influence without introducing an additional large fusion network.
- On the RadioML2018.10a benchmark, RGF-Net outperforms the compared single-branch and feature-concatenation baselines, with the clearest gain appearing at 0 dB. We report these results as evidence on a synthetic benchmark and explicitly discuss the remaining limits regarding real-world generalization and finer-grained adaptive fusion.

The remainder of this paper is organized as follows. Section 2 reviews related literature. Section 3 presents the architecture and training objective of RGF-Net. Section 4 describes the experimental setup and discusses the results. Section 5 concludes the paper.

2 Related Work

This section reviews AMR methods that are most relevant to the proposed framework.

2.1 Traditional Feature-Based AMR

Before deep learning became dominant, practical AMR systems mainly relied on handcrafted statistics and a separate classifier. Higher-order cumulants, cyclostationary descriptors, and wavelet-domain features are representative examples [10, 19]. These approaches can be effective when their assumptions match the signal model, but they remain sensitive to low SNR and channel mismatch.

2.2 Single-Model Deep Learning Approaches

The deep-learning transition in AMR was accelerated by O'Shea et al. [11], who showed that CNNs can learn discriminative representations directly from raw I/Q samples. Subsequent work explored VGG-style convolutional models [12] and deeper residual architectures [20, 21]. Recurrent models were also adopted because modulation patterns have clear temporal structure. LSTM-based AMR was studied in [13, 14], while GRU offers a lighter recurrent alternative with similar gating behavior [22]. These single-model approaches are effective but specialized: CNNs emphasize spatial structure, whereas recurrent networks emphasize temporal dynamics.

2.3 Hybrid and Fusion Models for AMR

Recognizing the limitations of single-model designs, researchers explored hybrid architectures. A representative example is CLDNN, which first uses convolutional layers to extract local patterns and then applies recurrent modeling to the transformed features [15]. A second line of work uses explicit fusion or ensemble learning, either by concatenating heterogeneous features before classification [17] or by combining classifier outputs through averaging and related rules [16, 18].

RGF-Net is closest to the second category, but differs from common fusion frameworks in two respects. First, the ResNet and GRU branches analyze the same raw input in parallel rather than sequentially, which preserves branch-specific specialization. Second, the final fusion is trainable but intentionally lightweight: a single scalar weight calibrates the relative contribution of the two posterior distributions. The contribution of this work therefore lies in a specific parallel hybrid formulation and an end-to-end decision-level calibration strategy, rather than in proposing a new convolutional or recurrent block.

3 The Proposed RGF-Net Framework

3.1 Problem Formulation

The goal of AMR is to map a received signal sequence to one of K modulation classes, denoted by $\mathcal{M} = \{m_1, m_2, \ldots, m_K\}$. In a non-cooperative setting, the continuous-time baseband signal can be written as

$$r(t) = s(t)e^{j(2\pi\Delta f t+\phi)} * h(t) + n(t), \tag{1}$$

where $s(t)$ is the transmitted complex baseband signal, $h(t)$ is the channel impulse response, Δf is the carrier-frequency offset, ϕ is the carrier-phase offset, and $n(t)$ denotes additive noise.

After down-conversion and sampling, a sequence of N complex samples is represented by the real-valued matrix

$$\mathbf{X} = \begin{bmatrix} I[1], I[2], \ldots, I[N] \\ Q[1], Q[2], \ldots, Q[N] \end{bmatrix} \in \mathbb{R}^{2\times N}, \tag{2}$$

where $I[n] = \Re\{r[n]\}$ and $Q[n] = \Im\{r[n]\}$. The training objective is to learn the parameters θ of a classifier f_θ by

minimizing a multi-class loss over the training set:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(\mathbf{X},y)\in\mathscr{D}} \left[\mathscr{L}(f_\theta(\mathbf{X}), y)\right]. \tag{3}$$

3.2 Architectural Rationale and Overview

The design goal of RGF-Net is to preserve complementary feature domains while keeping the fusion stage simple enough to analyze. Communication signals contain both spatial information, such as constellation geometry in the I/Q plane, and temporal information, such as the evolution of phase, frequency, and amplitude across time. We therefore use two standard but complementary branches: ResNet emphasizes spatial structure, whereas GRU emphasizes temporal dynamics.

A key modeling choice is that these two branches operate in parallel on the same raw observation, as shown in Figure 1, rather than in a sequential CNN-to-RNN pipeline. This means the GRU branch is not restricted to features already filtered by the convolutional frontend, and the ResNet branch is not forced to encode temporal cues that are more naturally modeled recurrently. The methodological contribution of RGF-Net lies in this parallel specialization together with the subsequent trainable decision fusion, not in introducing a new branch architecture.

3.3 Spatial Feature Extraction Branch: ResNet

The spatial branch uses a one-dimensional ResNet to extract robust spatial features from the input matrix $\mathbf{X}$. The network starts with a convolutional stem followed by six residual blocks. Each residual block contains two convolutional layers with Batch Normalization and ReLU activation, together with a shortcut connection that preserves gradient flow. A global average pooling layer then converts the final feature map into a fixed-length representation, which is mapped to class-posterior probabilities:

$$\mathbf{p}_{\text{ResNet}} = \text{softmax}\left(\mathbf{W}_{\text{ResNet}} \cdot \text{GAP}(f_{\theta'_{\text{ResNet}}}(\mathbf{X})) + \mathbf{b}_{\text{ResNet}}\right), \tag{4}$$

where $f_{\theta'_{\text{ResNet}}}$ denotes the convolutional part of the spatial branch.

3.4 Temporal Feature Extraction Branch: GRU

The temporal branch treats the same signal as a sequence of length N with two features per time step. We use two stacked GRU layers with 128 hidden units per layer. The first layer outputs a hidden sequence, and the second layer returns the final hidden state $\mathbf{h}_N$ as a summary representation of the signal. This representation is then mapped to class-posterior probabilities. The GRU dynamics are

$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z), \tag{5}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r), \tag{6}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \tag{7}$$

$$\mathbf{h}_t = (1-\mathbf{z}_t)\odot\mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t, \tag{8}$$

where $\odot$ denotes element-wise multiplication.

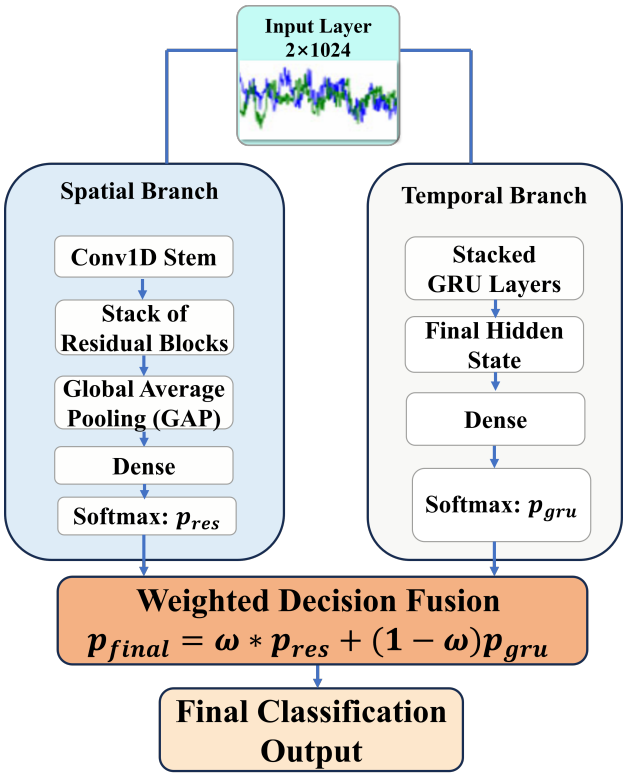


Figure 1: Architecture of the proposed RGF-Net. The same raw I/Q signal is processed in parallel by a ResNet spatial branch and a GRU temporal branch, and their posterior probabilities are fused through a learnable scalar weight.

3.5 Weighted Decision Fusion and End-to-End Training

Rather than concatenating high-dimensional feature vectors, which increases dimensionality and can obscure the contribution of each branch, we opt for decision-level fusion. In this setting, each branch already produces a class-posterior distribution, so the fusion module acts as a calibration layer over two informative classifiers.

The final probability distribution is computed as a learnable convex combination:

$$\mathbf{p}_{\text{final}} = w \cdot \mathbf{p}_{\text{ResNet}} + (1-w)\cdot \mathbf{p}_{\text{GRU}}, \tag{9}$$

where $w \in [0,1]$ is optimized jointly with all other network parameters. We chose a global scalar on purpose. Since both branches already output class-conditioned probabilities, a single weight is sufficient to estimate the dataset-level reliability balance between spatial and temporal evidence while adding only one extra parameter. This parsimonious design reduces the risk of overfitting relative to a larger gating network and makes the fusion behavior easy to interpret. At the same time, it should be viewed as a deliberately lightweight mechanism rather than a universal solution, because it does not adapt separately to each class or SNR condition.

The entire RGF-Net is trained end to end with categorical cross-entropy:

$$\mathscr{L}(\theta) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K} y_{i,k}\log(p_{\text{final},i,k}), \tag{10}$$

where B is the batch size, $y_{i,k}$ is the one-hot ground-truth label, and $p_{\text{final},i,k}$ is the predicted class probability. Joint optimization ensures that both branches and the fusion parameter are tuned toward the same classification objective.

4 Experimental Evaluation

4.1 Dataset and Preprocessing

We evaluate RGF-Net on the publicly available RadioML2018.10a dataset [23]. This benchmark is widely used in AMR studies and contains synthetic signals generated with GNU Radio under multiple modulation formats and SNR conditions. Because the dataset is synthetic, the experiments in this paper should be interpreted as controlled benchmark evidence rather than direct proof of over-the-air deployment performance.

Each sample contains 1024 complex baseband observations. For every record, we stack the in-phase and quadrature components into the matrix $\mathbf{X} \in \mathbb{R}^{2\times 1024}$ defined in Equation 2. The ResNet branch receives this sample as a two-channel one-dimensional input of shape 2×1024. The GRU branch uses the same signal after transposition, namely as a sequence of length 1024 with two features per time step. The dataset label is mapped to one of 24 integer class indices and then converted to a one-hot target vector for training. No handcrafted signal descriptors are introduced; all compared models use the same raw I/Q samples and the same 60%/20%/20% train/validation/test partition.

The main training settings are summarized in Table 1. These settings are kept fixed across all models so that the comparison reflects differences in model design rather than differences in optimization schedules.

4.2 Evaluation Metrics

We report overall accuracy together with macro-averaged precision, recall, and F1-score. Accuracy measures the proportion of correctly classified samples over the test set. Precision, recall, and F1-score are first computed for each class and then macro-averaged so that confusable classes are not overshadowed by easier categories. These metrics complement the accuracy curves by providing a more balanced view of class-wise behavior.

4.3 Baselines and Implementation Details

To evaluate RGF-Net, we compare it with representative single-branch models and with a common feature-fusion baseline. This selection allows us to isolate the effect of parallel spatiotemporal modeling and the effect of the proposed decision-level fusion rule.

4.3.1 Baseline Model Architectures

We compare RGF-Net against the following baselines:

- **VGG-CNN**: a conventional CNN baseline inspired by the architecture in [11].
- **ResNet-Only**: the standalone spatial branch used in RGF-Net.
- **GRU-Only**: the standalone temporal branch described in Section 3.4.
- **Feature-Concat**: a dual-branch baseline that concatenates the penultimate ResNet and GRU features and feeds the result to an additional dense classifier.

4.3.2 Implementation and Training Hyperparameters

All models were implemented in PyTorch and trained under identical settings. We used Adam with an initial learning rate of 1×10^{-3} for all models. The learning rate was adjusted with

Table 1: Summary of experimental settings and hyperparameters

Parameter	Value / Description
Dataset	RadioML2018.10a
Signal Length	1024 complex samples
Number of Classes	24
Data Split	60% training, 20% validation, 20% test
Input to ResNet	Raw I/Q tensor of shape 2×1024
Input to GRU	Sequence of shape 1024×2
Label Handling	Integer class index with one-hot target
Handcrafted Features	None; raw I/Q samples only
Optimizer	Adam
Initial Learning Rate	1×10^{-3}
Learning Rate Schedule	ReduceLROnPlateau (factor $= 0.1$, patience $= 5$)
Loss Function	Categorical cross-entropy
Batch Size	256
Number of Epochs	50 with early stopping
Dropout Rate	0.5 before final dense layers
Early Stopping	Patience $= 10$ on validation loss
GPU	NVIDIA GeForce RTX 3090 (24 GB)
Software	CUDA 11.2, cuDNN 8.1, PyTorch

ReduceLROnPlateau using a patience of 5 and a decay factor of 0.1, with the minimum learning rate capped at 1×10^{-6}. Dropout with rate 0.5 was applied before the final dense layers, Batch Normalization was used in the CNN and ResNet branches, and early stopping with patience 10 was applied on the validation loss. All models were trained for at most 50 epochs with batch size 256, and the final reported numbers correspond to the checkpoint with the lowest validation loss evaluated on the held-out test set. For RGF-Net, the learnable fusion weight w was initialized to 0.5 before training.

4.4 Results and Analysis

4.4.1 Overall Performance

Figure 2 summarizes classification accuracy versus SNR for all compared models. On RadioML2018.10a, RGF-Net yields the highest accuracy across the evaluated SNR range. The gap is most visible in the low-to-moderate SNR region, where the benchmark is more challenging. As shown in Table 2, at 0 dB RGF-Net achieves 78.3% accuracy, compared with 66.1% for the strongest single-branch baseline (ResNet-Only) and 72.8% for Feature-Concat. The corresponding gains are 12.2 and 5.5 percentage points, respectively. At 18 dB, the proposed model reaches 98.3% accuracy. These numbers indicate that the parallel dual-stream design is beneficial on this synthetic benchmark, but they should not be interpreted as establishing general superiority beyond the evaluated dataset.

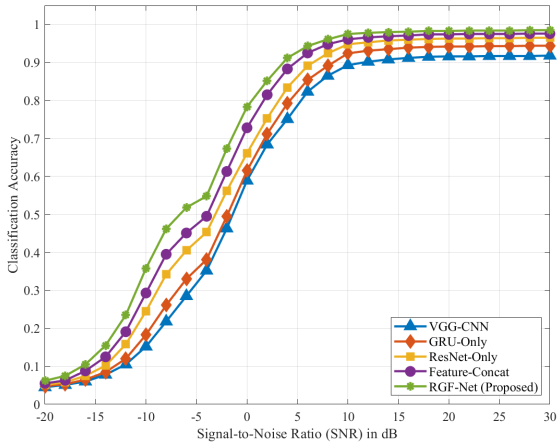


Figure 2: Classification accuracy versus SNR for RGF-Net and the compared baselines.

Table 2: Classification accuracy (%) at representative SNR points

Model	-4 dB	0 dB	4 dB	10 dB	18 dB
VGG-CNN	35.2	58.9	75.1	89.3	91.5
GRU-Only	38.1	61.5	79.2	92.4	94.1
ResNet-Only	45.3	66.1	83.4	94.8	96.2
Feature-Concat	49.5	72.8	88.3	96.1	97.4
RGF-Net	**54.8**	**78.3**	**91.2**	**97.5**	**98.3**

4.4.2 Confusion Matrix Analysis

To further inspect model behavior, Figure 3 shows the confusion matrix of RGF-Net at 10 dB. Most classes remain concentrated on the main diagonal, which is consistent with the high accuracy reported at this SNR. The remaining errors are mainly concentrated among spectrally similar or hierarchically related classes, such as higher-order QAM variants and AM sideband categories. This pattern suggests that the fused model alleviates several difficult confusions, although the confusion matrix should be interpreted as descriptive evidence rather than as a complete explanation of the fusion mechanism.

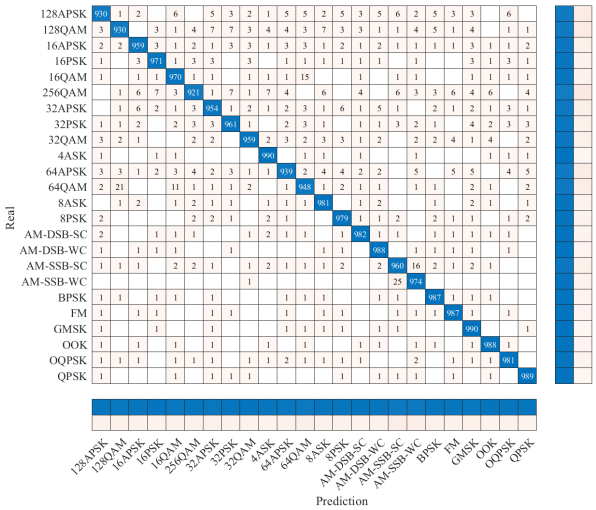


Figure 3: Confusion matrix of RGF-Net at 10 dB SNR.

4.4.3 Ablation Study

Table 3 compares the full RGF-Net with its constituent parts and simpler fusion rules. The dual-branch average-fusion model already outperforms the two single-branch baselines, indicating that spatial and temporal information are complementary on this dataset. Replacing the fixed average with a learnable scalar further improves the macro F1-score from 0.745 to 0.779. The learned weight converges to approximately 0.62, which suggests a modest dataset-level preference for the spatial branch while still preserving a substantial temporal contribution. Because w is global, this value should be interpreted as an overall calibration factor rather than as a class-specific or SNR-specific optimum.

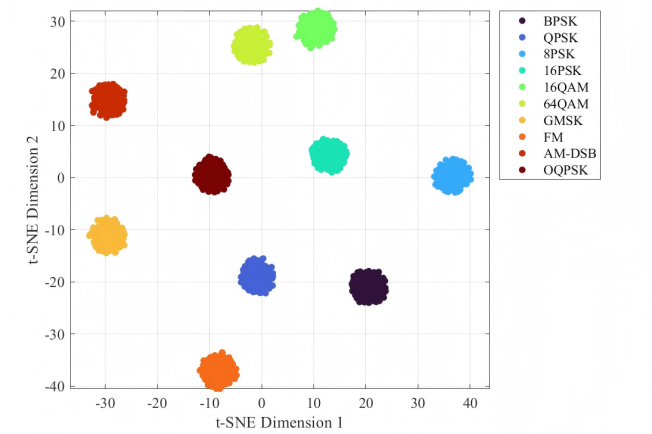
Table 3: Ablation study results (macro F1-score at 0 dB)

Model Configuration	Macro F1-Score
ResNet-Only (Spatial Branch)	0.658
GRU-Only (Temporal Branch)	0.611
Avg-Fusion (Fixed $w = 0.5$)	0.745
Feature-Concat	0.721
RGF-Net (Learnable w)	**0.779**

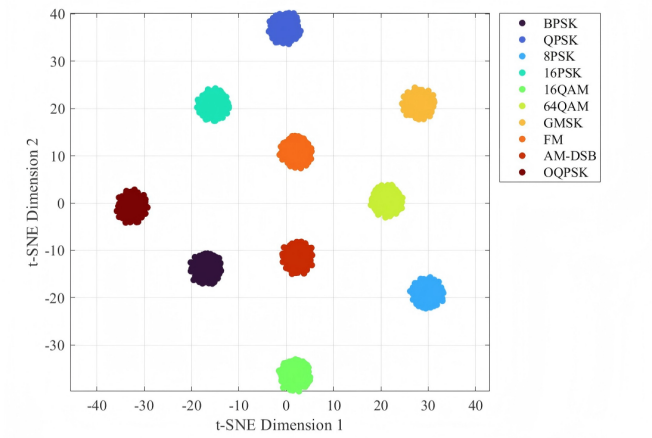
4.4.4 Feature Visualization with t-SNE

We also use t-SNE as a qualitative diagnostic to visualize the feature representations learned by the models. The purpose of this analysis is illustrative: it provides an intuitive view of the penultimate-layer embeddings, but it does not by itself constitute rigorous statistical proof of class separability. For this comparison, we extract the penultimate features of the ResNet-Only baseline and the full RGF-Net on the test set at 10 dB, then project them into two dimensions.

Relative to the ResNet-Only baseline in Figure 4(a), the RGF-Net embedding in Figure 4(b) shows more compact clusters and less visible overlap among several challenging modulation groups. This qualitative trend is consistent with the accuracy and confusion-matrix results. At the same time, the t-SNE plots should be interpreted cautiously because the visualization depends on a nonlinear projection and on hyperparameter choices. We therefore treat the figure as supporting evidence that the fused model learns a more organized representation on this benchmark, not as standalone proof of the underlying mechanism.



(a) ResNet-Only baseline



(b) RGF-Net

Figure 4: t-SNE visualization of penultimate-layer features on the test set at 10 dB.

5 Conclusion

This paper studied automatic modulation recognition in cognitive radio through RGF-Net, a parallel dual-stream model that combines a ResNet spatial branch, a GRU temporal branch, and a learnable decision-level fusion weight. On the RadioML2018.10a benchmark, this design improves classification performance relative to the compared single-branch and feature-concatenation baselines, with the largest gain appearing at 0 dB. The ablation study indicates that both branches contribute useful information and that a lightweight trainable fusion rule is preferable to fixed averaging on this dataset.

The results should nevertheless be interpreted within the scope of the present study. Our evaluation is limited to one synthetic dataset, so robustness to real over-the-air signals still needs to be verified. In addition, the current scalar fusion weight provides only a global calibration and does not adapt explicitly to different modulation categories or SNR conditions. Future work will therefore focus on evaluation on real-world datasets, finer-grained adaptive fusion mechanisms, and model compression for deployment on resource-constrained radio platforms.

Funding

This research received no external funding.

Author Contributions

Conceptualization, Fangning Shi and Yongkang Gong; methodology, Fangning Shi and Yiheng Zhang; software, Fangning Shi; validation, Fangning Shi and Yiheng Zhang; formal analysis, Fangning Shi and Yongkang Gong; investigation, Fangning Shi; resources, Yongkang Gong; data curation, Fangning Shi and Yiheng Zhang; writing—original draft preparation, Fangning Shi; writing—review and editing, Yiheng Zhang and Yongkang Gong; visualization, Fangning Shi; supervision, Yongkang Gong; project administration, Yongkang Gong; funding acquisition, Yongkang Gong. All authors have read and agreed to the published version of the manuscript.

Conflict of Interest

All the authors declare that they have no conflict of interest.

Data Available

The data and materials used in this study are available upon request from the corresponding author.

References

[1] Akyildiz, I.F., Lee, W.-Y., Vuran, M.C., Mohanty, S.: Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. Computer Networks **50**(13), 2127–2159 (2006). https://doi.org/10.1016/j.comnet.2006.05.001

[2] Zhang, P., Wang, G., Chen, S., Yu, Y., Chen, L.: The application of mobile edge computing in the space-air-ground integrated network. Journal of Intelligent Computing and Networking **1**(1), 71–97 (2025). https://doi.org/10.64509/jicn.11.13

[3] Mitola III, J.: Cognitive radio: An integrated agent architecture for software defined radio. PhD Thesis, KTH Royal Institute of Technology (2000).

[4] Luo, M., Mao, J., Xiong, G., Luo, M., Mao, J., Xiong, G., Anjum, M.R.: A semantic communication-assisted federated learning framework for Internet of Vehicles.

Journal of Intelligent Computing and Networking **2**(1), 55–69 (2026). https://doi.org/10.64509/jicn.21.74

[5] Yucek, T., Arslan, H.: A survey of spectrum sensing algorithms for cognitive radio applications. IEEE Communications Surveys & Tutorials **11**(1), 116–130 (2009). https://doi.org/10.1109/SURV.2009.090109

[6] Dobre, O.A., Abdi, A., Bar-Ness, Y., Su, W.: Survey of automatic modulation classification techniques: Classical approaches and new trends. IET Communications **1**(2), 137–156 (2007). https://doi.org/10.1049/iet-com:20050176

[7] Augustine, Z., Alabi, C., Njoku, F.C., Imoize, A.L., Raymond, J.: Throughput optimization using spectrum sensing vertical hypotheses integrated with F-OFDM technique. Journal of Intelligent Computing and Networking **2**(1), 1–12 (2026). https://doi.org/10.64509/jicn.21.50

[8] Huang, S., Jiang, Y., Gao, Y., Feng, Z., Zhang, P.: Automatic modulation classification using contrastive fully convolutional network. IEEE Wireless Communications Letters **8**(4), 1044–1047 (2019). https://doi.org/10.1109/LWC.2019.2904956

[9] Wei, W., Mendel, J.M.: Maximum-likelihood classification for digital amplitude-phase modulations. IEEE Transactions on Communications **48**(2), 189–193 (2000). https://doi.org/10.1109/26.823550

[10] Gardner, W.A.: Signal interception: A unifying theoretical framework for feature detection. IEEE Transactions on Communications **36**(8), 897–906 (1988). https://doi.org/10.1109/26.3769

[11] O'Shea, T., Corgan, J.: Convolutional radio modulation recognition networks. In International Conference on Engineering Applications of Neural Networks (EANN), pp. 213–226 (2016). https://doi.org/10.1007/978-3-319-44188-7_16

[12] Wang, Y., Liu, M., Yang, J., Gui, G.: Data-driven deep learning for automatic modulation recognition in cognitive radios. IEEE Transactions on Vehicular Technology **68**(4), 4074–4077 (2019). https://doi.org/10.1109/TVT.2019.2900460

[13] Rajendran, S., Meert, W., Giustiniano, D., Lenders, V., Pollin, S.: Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. IEEE Transactions on Cognitive Communications and Networking **4**(3), 433–445 (2018). https://doi.org/10.1109/TCCN.2018.2835460

[14] Zhou, Q., Jing, X., He, Y., Cui, Y., Kadoch, M., Cheriet, M.: LSTM-based automatic modulation classification. In 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1–4 (2020). https://doi.org/10.1109/BMSB49480.2020.9379677

[15] Sainath, T.N., Vinyals, O., Senior, A., Sak, H.: Convolutional, long short-term memory, fully connected deep neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4580–4584 (2015). https://doi.org/10.1109/ICASSP.2015.7178838

[16] Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine **6**(3), 21–45 (2006). https://doi.org/10.1109/MCAS.2006.1688199

[17] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1725–1732 (2014). https://doi.org/10.1109/CVPR.2014.223

[18] Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(3), 226–239 (1998). https://doi.org/10.1109/34.667881

[19] Swami, A., Sadler, B.M.: Hierarchical digital modulation classification using cumulants. IEEE Transactions on Communications **48**(3), 416–429 (2000). https://doi.org/10.1109/26.837045

[20] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

[21] West, N.E., O'Shea, T.: Deep architectures for modulation recognition. In: IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), pp. 1–6 (2017). https://doi.org/10.1109/DySPAN.2017.7920754

[22] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734 (2014). https://doi.org/10.3115/v1/D14-1179

[23] O'Shea, T., Roy, T., Clancy, T.C.: Over-the-air deep learning based radio signal classification. IEEE Journal of Selected Topics in Signal Processing **12**(1), 168–179 (2018). https://doi.org/10.1109/JSTSP.2018.2797022