

Blind Recognition of Puncturing Patterns in Convolutional Codes with Experimental Validation

Pranshu Singh and Swaminathan R.

Department of Electrical Engineering, Indian Institute of Technology (IIT) Indore, India

Email: phd2401102002@iiti.ac.in, swamiramabadrans@iiti.ac.in

Abstract—Blind estimation of puncturing patterns plays a vital role in modern communication systems, including adaptive modulation and coding, non-cooperative and cognitive radio systems, reconfigurable wireless transceivers, and signal intelligence applications. In this work, we propose a blind recognition framework for puncturing pattern identification in convolutional codes. The proposed approach exploits statistical features extracted from the received noisy bit streams over Rayleigh fading channel and employs deep-learning based classification models to accurately recognize the underlying puncturing patterns without any prior knowledge of encoder parameters. Extensive simulation results are presented under different noisy channel conditions, and the accuracy versus signal-to-noise ratio (SNR) is evaluated for various test cases of puncturing pattern. The obtained simulation results are subsequently verified through real-time experiments implemented on a USRP-based software-defined radio (SDR) testbed. The performance of the proposed blind puncturing recognition framework is finally compared with existing methods reported in the literature, demonstrating its robustness and generalization capability.

Index Terms—Adaptive modulation and coding, blind recognition, CNN-attention, convolutional codes, non-cooperative systems, puncturing patterns, USRP B210.

I. INTRODUCTION

Forward error correction (FEC) codes improve communication reliability by adding redundancy that allows the receiver to detect and correct channel errors. Puncturing increases the code rate and transmission efficiency by selectively removing encoded bits without modifying the original encoder. In conventional systems, the encoder parameters are known at the receiver; however, in non-cooperative application scenarios such as military communications, spectrum monitoring, and signals intelligence, this information may be unavailable. In such cases, blind identification of the FEC encoder becomes necessary for reliable signal recovery [1]. Although adaptive modulation and coding (AMC) systems transmit coding parameters through control channels, this approach may reduce spectral efficiency. Automatic recognition of the coding scheme at the receiver can therefore improve spectral utilization and enable more flexible communication systems [2], [3].

In [4] and [5], parameter estimation algorithms were developed for turbo and convolutional codes operating in noisy conditions. For Reed-Solomon (RS) codes, [6] proposed methods based on the number of non-zero columns and the count of non-zero elements in the column echelon form of the received data. Blind recognition of puncturing patterns in punctured

convolutional codes has been explored in [7] and [8]. The work in [9] investigates techniques to identify and reconstruct convolutional codes when some encoded bits are periodically omitted due to puncturing. In [10], the proposed algorithms for identifying interleaver parameters are tested using both punctured and unpunctured convolutionally encoded sequences. However, puncturing pattern of the convolutional codes have not been identified.

Classical approaches proposed in [7]–[10] rely on algebraic techniques to determine punctured positions. These methods require predefined code structures and often involve computationally intensive searches, with performance degrading under high noise, synchronization errors, and fading channels. More recently, deep learning approaches have been applied to channel encoder recognition [11]. Further, long short-term memory (LSTM)-based decoder has been proposed in [12] for decoding punctured convolutional and turbo codes. However, blind recognition of puncturing pattern from the incoming convolutional coded signals using deep learning techniques has not been explored in the literature to the best of our knowledge. This paper adopts a convolutional neural network (CNN) - attention framework, as it is capable of capturing both local feature representations and broader contextual dependencies within the received signal. In contrast to conventional CNN or recurrent architectures, the attention mechanism enables the model to emphasize the most informative bit positions, leading to improved reliability in identifying puncturing patterns.

A. Motivation

The motivations for this study are summarized as follows:

- When a convolutional code is punctured and if the puncturing pattern is not known at the receiver, especially in a non-cooperative military communication scenario, identifying the structure of the transmitted code becomes a difficult blind detection problem.
- Many conventional reconstruction and signal processing techniques rely on predefined structural properties of the code. Their effectiveness decreases when the received data is strongly affected by noise, synchronization uncertainty, or channel impairments such as fading.
- Learning-based approaches provide an alternative by extracting useful patterns directly from the received signal.

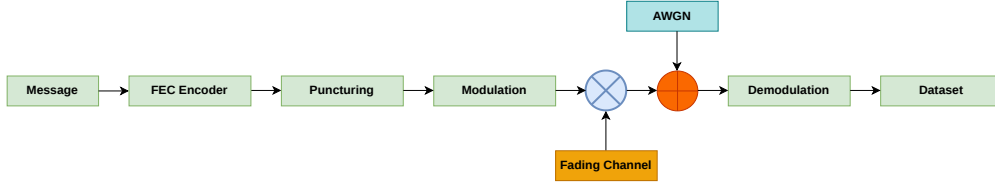


Fig. 1. Generic block diagram for the generation of dataset.

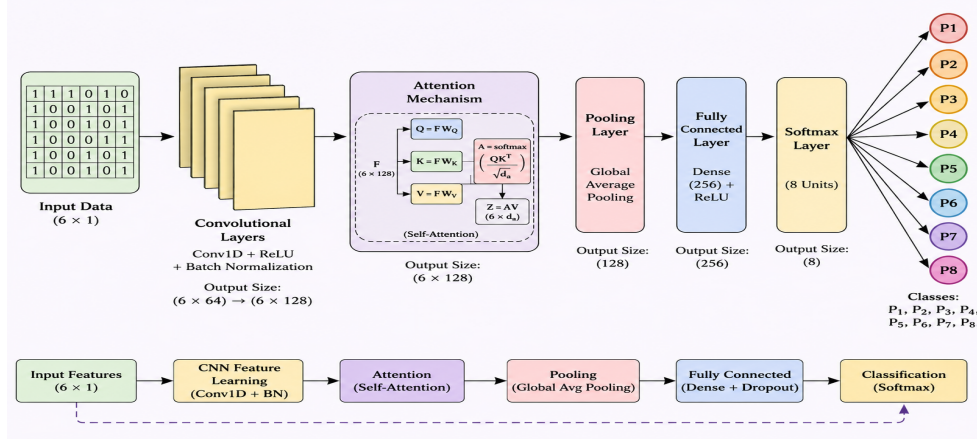


Fig. 2. Architecture of CNN-attention model.

B. Contributions

The main contributions of this work are summarized as follows:

- A dataset of convolutionally encoded signals with puncturing is generated using eight 2×2 puncturing patterns over an SNR range from -10 dB to 30 dB under Rayleigh fading conditions, representing diverse channel scenarios.
- A CNN-attention based framework is developed to analyze punctured encoded sequences, where convolutional layers extract local signal features and the attention module emphasizes the most relevant information.
- The proposed approach is evaluated through extensive experiments, including accuracy versus SNR analysis and normalized confusion matrices, demonstrating reliable identification of puncturing patterns under different noise levels.
- The method is further validated using real-time measurements obtained from a GNU-Radio-based USRP B210 platform, and its performance is compared with existing techniques reported in the literature.

II. SYSTEM MODEL FOR BLIND RECOGNITION OF PUNCTURING PATTERNS

Convolutional codes are widely employed for error control in digital communication systems due to their ability to improve transmission reliability over noisy channels. To obtain different coding rates without modifying the encoder structure, puncturing is applied by periodically removing selected coded bits according to a predefined pattern, providing a flexible trade-off between bandwidth efficiency and error protection.

The considered communication model employs a rate-1/2 convolutional encoder followed by a puncturing block to generate multiple effective coding rates. The punctured bit stream is modulated using binary phase shift keying (BPSK) and transmitted over a channel affected by additive white Gaussian noise (AWGN) and Rayleigh fading under different SNR conditions, as illustrated in Fig. 1.

The received signal can be expressed as

$$y(t) = h(t)x(t) + n(t) \quad (1)$$

where $x(t)$ denotes the transmitted signal, $h(t)$ represents the fading channel coefficient, and $n(t)$ is the AWGN component. For a Rayleigh fading channel, the probability density function (PDF) of the amplitude r of the channel coefficient is given by

$$p(r) = \frac{r}{\sigma^2} e^{-r^2/(2\sigma^2)}, \quad r \geq 0 \quad (2)$$

where σ^2 is the variance of the Gaussian distribution. Since the receiver does not know the puncturing pattern used during transmission, a CNN with an attention mechanism analyzes the received samples and predicts the underlying puncturing structure directly from the observed data.

A. Dataset Generation for Punctured Convolutional Codes

A dataset was generated for blind identification of puncturing patterns in convolutionally encoded signals transmitted over a noisy channel. A rate-1/2 convolutional encoder with constraint length 3 and generator polynomials (7, 5) (octal) was employed. For each realization, $K = 1024$ random

bits were encoded. Eight 2×2 puncturing matrices (P_1 – P_8), listed in Table I, were considered, where a puncturing pattern is generally represented by an $n_M \times L$ matrix, with n_M denoting the number of encoder outputs and L the puncturing period. The puncturing matrices P_1 – P_8 were adopted from the benchmark set in [9] for punctured convolutional codes. Since the focus of this work is puncturing pattern identification rather than rate optimization, a diverse set of puncturing matrices, including rate-one cases, is considered to generate distinct classification classes.

TABLE I
SET OF 2×2 PUNCTURING MATRICES USED IN THE STUDY

Matrix	Pattern	Matrix	Pattern
P_1	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	P_5	$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$
P_2	$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	P_6	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$
P_3	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	P_7	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
P_4	$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	P_8	$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$

In each realization, a puncturing matrix was randomly selected and periodically applied to the encoder output. The punctured sequence was transmitted through a Rayleigh fading channel for SNR values ranging from -10 dB to 30 dB. From the received sequence, six statistical features namely mean, variance, standard deviation, skewness, kurtosis, and signal energy were extracted. The resulting six-dimensional feature vector, together with the corresponding SNR value and puncturing pattern label, was stored to form the dataset used for training and testing the proposed CNN-attention classifier. By repeating this process for 5000 realizations across all SNR values, a large labeled dataset was generated and saved in comma-separated format. Each row corresponds to a received signal instance, while the class label indicates the applied puncturing pattern. The overall procedure is illustrated in Fig. 1.

To experimentally validate the puncturing patterns, a GNU Radio flowgraph was implemented using two Intel Core i9-based PCs and a pair of USRP B210 SDRs placed approximately 1 m apart. The convolutionally encoded bitstream was punctured using predefined 2×2 matrices prior to modulation and transmission. At the receiver, real-time demodulation and bit recovery were performed, and the received data were stored for experimental validation.

B. Proposed CNN-Attention model

A CNN-attention model is employed to classify puncturing patterns from the extracted feature sequence. Let the input feature vector be

$$\mathbf{x} = [x_1, x_2, \dots, x_L] \in \mathbb{R}^L, \quad (3)$$

where L denotes the sequence length. The input vector is reshaped into a matrix $\mathbf{X} \in \mathbb{R}^{L \times d}$, where d represents the feature dimension. A one-dimensional convolution layer extracts local features

TABLE II
ARCHITECTURE SPECIFICATIONS OF THE IMPLEMENTED CNN-ATTENTION MODEL FOR PUNCTURING PATTERNS

Layer Type	Output Shape	Kernel / Units	Activation
Input Layer	(6, 1)	–	–
Conv1D	(6, 64)	3	ReLU
Batch Normalization	(6, 64)	–	–
Conv1D	(6, 128)	3	ReLU
Batch Normalization	(6, 128)	–	–
Attention	(6, 128)	$d_a = 128$	–
Global Average Pooling	(128)	–	–
Dense (Feature Layer)	(256)	–	ReLU
Dropout	(256)	0.4	–
Dense (Softmax Classifier)	(8)	–	Softmax

TABLE III
ABLATION STUDY OF THE PROPOSED CNN-ATTENTION MODEL FOR PUNCTURING PATTERN CLASSIFICATION

Model ID	Configuration	Accuracy (%)
M1	Full CNN + BatchNorm + Attention + Dropout	94.8
M2	CNN + BatchNorm (No Attention)	87.4
M3	CNN + Attention (No BatchNorm)	85.9
M4	CNN only (No BatchNorm, No Attention)	81.2
M5	CNN + BatchNorm + Attention (No Dropout)	89.3

$$\mathbf{F}_k(t) = \sigma \left(\sum_{i=0}^{h-1} \mathbf{W}_k(i) \mathbf{X}(t+i) + b_k \right), \quad (4)$$

where h is the kernel size, $\mathbf{W}_k$ denotes the convolution weights, b_k is the bias, and $\sigma(\cdot)$ represents the ReLU activation function. The resulting feature matrix $\mathbf{F}$ is then processed by an attention mechanism to capture dependencies within the sequence.

The query, key, and value matrices are, respectively, computed as

$$\mathbf{Q} = \mathbf{F}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{F}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{F}\mathbf{W}_V, \quad (5)$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learnable projection matrices. The attention weights are obtained using

$$\mathbf{A} = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_a}} \right), \quad (6)$$

where d_a denotes the attention dimension. The attention-weighted representation $\mathbf{Z} = \mathbf{A}\mathbf{V}$ is globally pooled, i.e.,

TABLE IV
COMPUTATIONAL COMPLEXITY COMPARISON OF DEEP LEARNING MODELS

Metric	1D CNN	CNN-SVM	CNN-Attn.
Accuracy (%)	89.5	92.5	94.8
FLOPs/sample	1.6×10^6	2.3×10^6	4.5×10^6
Training time (ms/epoch)	6.5	9.2	13.8
Testing time (ms)	0.6	1.0	1.9
Classification time (ms/sample)	0.3	0.5	0.9
Parameters (M)	1.2	2.0	2.8

$\mathbf{z} = \text{GAP}(\mathbf{Z})$, and passed to a fully connected layer with softmax output given by

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_o \mathbf{z} + \mathbf{b}_o), \quad (7)$$

where $\hat{\mathbf{y}}$ is the predicted class probability vector, and $\mathbf{W}_o$ and $\mathbf{b}_o$ are the output-layer weight matrix and bias vector, respectively. The model is trained using the categorical cross-entropy loss, which is given by

$$\mathcal{L} = - \sum_{c=1}^C y_c \log(\hat{y}_c), \quad (8)$$

where C denotes the number of classes, y_c is the true label for class C , and $\hat{y}_c$ denotes the predicted probability of class C . The overall architecture is illustrated in Fig. 2.

Table II summarizes the CNN-attention architecture for an input vector of size 6×1 . Two convolutional layers followed by batch normalization extract feature maps, the attention layer captures dependencies among feature positions, and global average pooling, dense, and softmax layers perform final classification into eight puncturing pattern classes.

Performance evaluation of the proposed CNN-attention model is carried out in terms of classification accuracy, which is computed using true positives (TP) and false positives (FP) for each puncturing pattern, and is defined as

$$\text{Accuracy (in \%)} = \frac{TP}{TP + FP} \times 100. \quad (9)$$

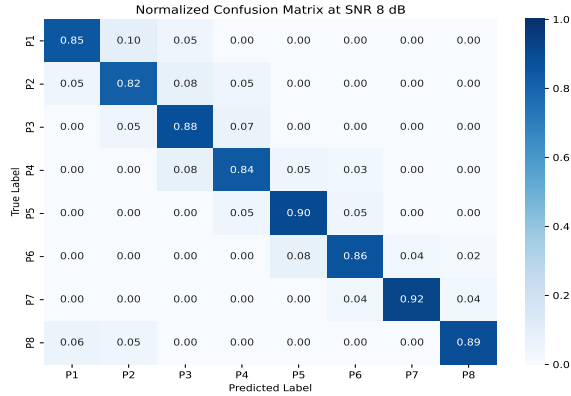


Fig. 3. Confusion matrix for CNN-attention model at 8 dB SNR.

III. SIMULATION RESULTS AND DISCUSSIONS

Table III presents the ablation study of the CNN-attention model. The complete model (M1) achieves the highest accuracy. Removing the attention module (M2) or batch normalization (M3) reduces performance, while the basic CNN model (M4) yields the lowest accuracy. A slight decrease in accuracy for M5 further indicates the contribution of dropout to improved generalization.

Fig. 3 shows the normalized confusion matrix of the proposed CNN-attention model at 8 dB SNR for eight puncturing

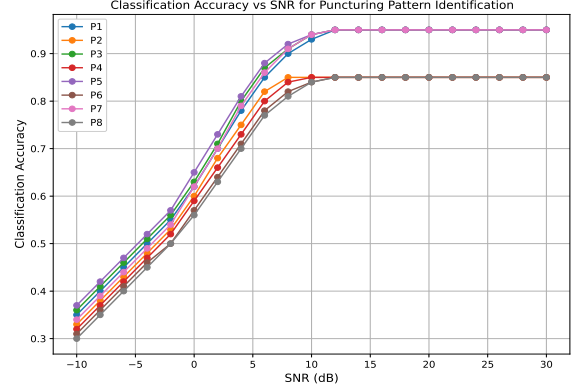


Fig. 4. Classification accuracy for different puncturing patterns over a Rayleigh fading channel.

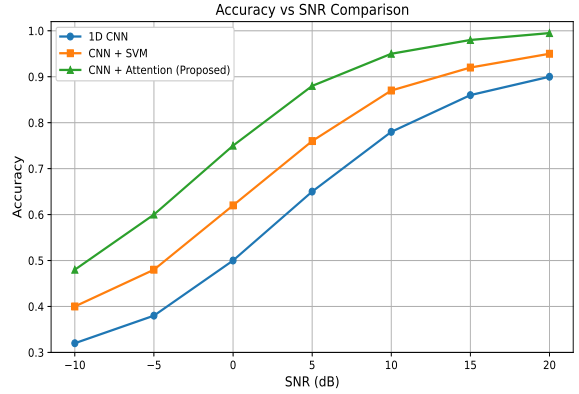


Fig. 5. Classification accuracy comparison for different deep learning models over a Rayleigh fading channel.

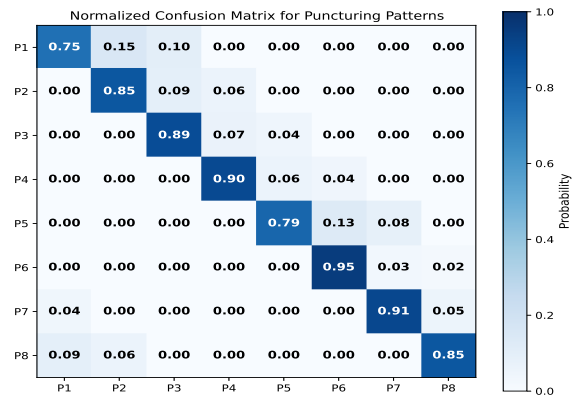


Fig. 6. Confusion matrix for real-time classification of puncturing patterns using a hardware dataset.

patterns (P_1 - P_8) under a Rayleigh fading channel. Each row represents the true class, while each column represents the predicted class. The diagonal entries range from 0.82 to 0.92, indicating reliable recognition of the puncturing patterns at a relatively low SNR. Among the considered patterns, P_7 achieves the highest classification accuracy (92%), followed by P_5 (90%), whereas P_2 exhibits the lowest accuracy (82%). The non-zero off-diagonal entries correspond to occasional confusion between structurally similar puncturing patterns. The dominant diagonal entries confirm effective puncturing pattern classification.

Fig. 4 shows the accuracy versus SNR curves for puncturing patterns P_1 to P_8 over a Rayleigh fading channel using the proposed CNN-attention model. At low SNR (-10 to 0 dB), the accuracy remains near zero due to severe noise, while a sharp rise is observed in the intermediate region, forming a waterfall behavior as the signal becomes more distinguishable. Further, we have also generated a dataset over a binary symmetric channel (BSC), which shows an accuracy of around 95% at a bit error probability (BEP) of 10^{-3} , around 89% at 10^{-2} , and about 85% at 2×10^{-2} . Compared with the method in Fig. 3 of [9], the proposed approach achieves consistently better performance in identifying puncturing patterns under moderate and high noise conditions, while maintaining lower computational complexity than conventional search-based methods.

Fig. 5 compares the performance of the proposed CNN-attention, 1D-CNN, and CNN-SVM models under Rayleigh fading. The corresponding computational complexity is given in Table IV. Despite higher computational cost, the proposed model achieves improved classification performance, demonstrating an effective trade-off. In general, classification accuracy increases with SNR due to reduced noise effects. At -10 dB, the 1D CNN achieves about 0.32, while CNN-SVM and CNN-attention reach approximately 0.40 and 0.48, respectively. As SNR increases, the performance gap becomes more pronounced. At 0 dB, accuracies are around 0.50, 0.62, and 0.75, respectively. At 10 dB, the proposed model achieves nearly 0.95, outperforming CNN-SVM (0.87) and 1D CNN (0.78). At 20 dB, all models approach saturation, with CNN-attention attaining about 0.99, remaining superior to CNN-SVM (0.95) and 1D CNN (0.90).

Fig. 6 shows the normalized confusion matrix of the CNN-attention model for eight puncturing patterns (P_1 - P_8) considering real-time hardware-generated dataset using USRP B210. Each row represents the true class and each column the predicted class. The strong diagonal entries (0.75–0.90) indicate reliable recognition of the puncturing patterns. In particular, P_1 , P_3 , P_5 , and P_7 achieve nearly 90% accuracy, while P_2 , P_4 , P_6 , and P_8 obtain around 80%. The small off-diagonal values indicate limited misclassification, mainly due to real-time noise and hardware effects. The recognition accuracy varies across puncturing patterns due to their structural characteristics. Patterns such as P_1 , P_3 , P_5 , and P_7 exhibit more distinctive transmission arrangements and are therefore easier to classify. In contrast, P_2 , P_4 , P_6 , and P_8 share similar transmission positions, leading to a higher

probability of misclassification.

IV. CONCLUSION

This work presented a learning-based method for blind identification of puncturing patterns in convolutionally coded systems. A dataset was generated using a rate-1/2 encoder with multiple puncturing matrices over a Rayleigh fading channel under varying SNR conditions, and a CNN-attention model was used for classification. The network effectively extracts relevant features and focuses on informative components, achieving higher accuracy than other models, particularly at moderate and high SNR levels, albeit with a higher classification time. Unlike conventional blind reconstruction methods reported in the literature, which exhibit exponential computational complexity, the proposed approach enables efficient direct classification with significantly lower complexity and improved accuracy. Finally, the proposed model was also further validated using real-time data acquired through a USRP B210 and GNU Radio setup, demonstrating its practical applicability.

ACKNOWLEDGMENT

This work was supported by the EMR-II scheme of CSIR-HRDG (22/0881/23/EMR-II), the TTDF scheme of DoT under 6G project (ID: 209), and by the 5G Use Case Lab at IIT Indore, established with funding from DoT.

REFERENCES

- [1] A. D. Yardi, S. Vijayakumaran and A. Kumar, "Blind reconstruction of binary cyclic codes from unsynchronized bitstream," *IEEE Trans. on Commun.*, vol. 64, no. 7, pp. 2693-2706, July 2016.
- [2] C. Choi and D. Yoon, "Enhanced blind interleaver parameters estimation for noisy environment," *IEEE Access*, vol. 6, pp. 5910-5915, Sept. 2017.
- [3] T. Xia and H.-C. Wu, "Novel blind identification of LDPC codes using average LLR of syndrome a posteriori probability," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 632-640, Feb. 2014.
- [4] M. Marazin, R. Gautier, and G. Burel, "Dual code method for blind identification of convolutional encoder for cognitive radio receiver design," in *Proc. IEEE GLOBECOM*, Honolulu, USA, 2009, pp. 1-6.
- [5] Y. Ding, Z. Huang, and J. Zhou, "An improved blind recognition method for synchronization position and coding parameters of k/n rate convolutional codes in a noisy environment," *IEEE Access*, vol. 8, pp. 171305-171315, 2020.
- [6] R. Swaminathan and A. S. Madhukumar, "Blind parameter estimation of turbo convolutional codes: noisy and non-synchronized scenario," *Digital Signal Process.*, vol. 95, Article 102577, Dec. 2019.
- [7] M. Cluzeau and M. Finiasz, "Reconstruction of punctured convolutional codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Oct. 2009, pp. 75-79.
- [8] M. Marazin, R. Gautier, and G. Burel, "Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream," *IET Signal Processing*, vol. 6, no. 2, pp. 122-131, Apr. 2012.
- [9] M. Teimouri, "Blind reconstruction of punctured convolutional codes," *Phy. Comm.*, vol. 47, pp. 101297, 2021.
- [10] R. Swaminathan, A. S. Madhukumar, N. W. Teck, and S. C. M. Samson, "Parameter estimation of convolutional and helical interleavers in a noisy environment," *IEEE Access*, vol. 5, pp. 6151-6167, 2017.
- [11] S. Dehdashtian, M. Hashemi, and S. Salehkaleybar, "Deep-learning based blind recognition of channel code parameters over candidate sets under AWGN and multi-path fading conditions," *IEEE Access*, vol. 8, pp. 173666-173681, 2020.
- [12] Y. Yan and L. Dai, "Decoding for punctured convolutional and Turbo codes: A deep learning solution for protocols compliance," *IEEE Trans. Commun.*, vol. 74, no. 1, pp. 200-216, Jan. 2026.