International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(3s) 2026 pp. 960-965
https://doi.org/10.70917/ijcisim-2026-2410
Article

LEARNING-BASED TWO-WAY FEEDBACK CHANNEL CODING: A COMPARATIVE ANALYSIS OF NEURAL NETWORK ARCHITECTURES

Sneha Bharti[1], Neha Rani[2], Ravi Bhushan Kumar[2], Rahul Kumar[3], Prashant Kumar[2]

[1]Department of Electronics and Communication Engineering (ECE), Government Engineering College, Jamui, Bihar, India.
Email: sneha.sanju05@gmail.com
[2]Department of Electronics and Communication Engineering (ECE), Saharsa College of Engineering, Saharsa, Bihar, India.
Email: neharani13893@gmail.com; rbhushan497@gmail.com; 4242prashant@gmail.com
[3]Department of Electrical Engineering (EE), Saharsa College of Engineering, Saharsa, Bihar, India.
Email: rahulkumar04242@gmail.com

Abstract: The integration of deep learning techniques with classical channel coding theory has ushered in a new generation of communication systems capable of end-to-end optimisation under realistic channel conditions. This article presents a systematic comparative investigation of five distinct neural network paradigms—convolutional neural network (CNN) encoder-decoder pairs, long short-term memory (LSTM) recurrent architectures, attention-based transformer-inspired models, turbo autoencoders, and a novel hybrid CNN-LSTM framework—applied to the two-way feedback channel coding problem. By leveraging backward channel feedback, encoder representations are iteratively refined across multiple transmission rounds, significantly improving bit error rate (BER) performance beyond the single-shot, open-loop regime. Simulations conducted over additive white Gaussian noise (AWGN) and Rayleigh fading channels demonstrate that the proposed hybrid architecture achieves a coding gain of approximately 5.3 dB relative to an uncoded baseline at a BER of 10^{-}´, while exhibiting favourable trade-offs between computational latency and error-correction capability. These results establish the viability of learning-based feedback coding as a foundation for sixth-generation (6G) physical-layer design.

Keywords: deep learning, feedback channel coding, autoencoder, LSTM, attention mechanism, bit error rate, 6G communications, end-to-end learning

1. Introduction

Shannon's seminal work established that the capacity of a memoryless channel is not increased by the availability of noiseless feedback (Shannon, 1956). Nevertheless, feedback dramatically simplifies code construction and reduces decoding complexity, motivating its exploitation in practical systems where encoder refinement across retransmission rounds is feasible. The emerging paradigm of semantic and goal-oriented communications further amplifies the relevance of adaptive, feedback-aware encoding strategies, particularly as future wireless networks evolve toward 6G specifications demanding ultra-reliable low-latency communication (URLLC) at sub-millisecond latency targets (Saad et al., 2020).

Classical feedback coding schemes, such as the Schalkwijk-Kailath (SK) algorithm for the AWGN channel, achieve remarkable performance gains through linear feedback processing but remain sensitive to channel estimation errors and do not generalise naturally to non-Gaussian environments (Schalkwijk & Kailath, 1966). Deep learning offers a principled mechanism for discovering feedback utilisation strategies from data alone, without explicit hand-crafted signal processing rules. O'Shea and Hoydis pioneered the idea of treating the transmitter-receiver pair as an

autoencoder, jointly training both components to minimise end-to-end distortion through stochastic gradient descent (O'Shea & Hoydis, 2017). Subsequent work extended this framework to multi-round feedback scenarios, where each encoder re-encoding step refines the transmitted codeword based on the previous channel output relayed through the feedback link.

Despite significant progress, a rigorous architecture-level comparison of neural feedback coding systems remains absent from the literature. Existing studies independently benchmark individual architectures without standardising evaluation conditions, channel models, or performance metrics. This paper bridges that gap by implementing and evaluating five architectures under an identical experimental protocol, thereby enabling fair and reproducible performance comparisons.

The primary contributions of this work are as follows: (i) a unified mathematical formulation of learning-based two-way feedback channel coding; (ii) a modular simulation framework accommodating all five architectures; (iii) comprehensive BER-versus-SNR benchmarking across AWGN and Rayleigh fading channels; and (iv) analysis of training convergence, inference latency, and practical deployment considerations for each architecture.

2. System Model and Problem Formulation

Consider a two-way channel in which a transmitter (Alice) communicates a binary message vector $u \in \{0,1\}^k$ to a receiver (Bob) over a noisy forward channel, while Bob simultaneously transmits feedback signals to Alice through an idealised or noisy reverse channel. The system operates over T feedback rounds, each indexed by $t \in \{1, \ldots, T\}$.

2.1 Forward Channel Model

At round t, Alice produces an encoded vector $x_t = f\theta(u, z_1, \ldots, z_{t-1})$, where $f\theta$ represents the parameterised neural encoder with weights θ, and z_{t-1} denotes the feedback signal received from Bob at the end of round $t-1$. The power constraint is enforced as $E[\|x_t\|^2] \leq P$. The channel output y_t is modelled as $y_t = x_t + n_t$, where $n_t \sim \gamma(0, \sigma^2 I)$ for the AWGN case. For the Rayleigh fading channel, $y_t = h_t \circ x_t + n_t$ with $h_t \sim \gamma\mathbb{C}(0, 1)$.

2.2 Feedback Link and Decoder

After receiving y_t, Bob computes and transmits a feedback signal $z_t = g\varphi(y_1, \ldots, y_t)$ through the reverse channel. In the noiseless feedback setting z_t arrives at Alice perfectly; in the noisy variant, additive Gaussian perturbations corrupt the feedback. Upon completion of all T rounds, Bob's neural decoder $h\psi$ produces an estimate $\hat{u} = h\psi(y_1, \ldots, y_1)$. The joint optimisation objective is the cross-entropy loss: $L(\theta, \varphi, \psi) = E[-u^T \log \hat{u} - (1-u)^T \log(1-\hat{u})]$, minimised via stochastic gradient descent.

3. Neural Network Architectures Under Evaluation

3.1 CNN Encoder-Decoder

The convolutional architecture processes the concatenated input (u, z_{t-1}) through stacked one-dimensional convolutional layers with kernel size 3, followed by batch normalisation and ReLU activations. The final layer applies a hyperbolic tangent to enforce power normalisation. The decoder mirrors this structure, receiving all channel observations $\{y_t\}$ concatenated along the feature dimension. Convolutional models excel at extracting local correlation patterns and are computationally efficient on modern hardware accelerators, but their fixed receptive field may limit long-range temporal dependency capture across feedback rounds (LeCun et al., 1998).

3.2 LSTM Encoder-Decoder

Recurrent LSTM cells maintain a hidden state that naturally accumulates information across T feedback rounds, making them well-suited to sequential refinement tasks. At each round t, the encoder LSTM cell receives (u, z_{t-1}) as input and updates its hidden state h_t, from which x_t is derived. The decoder similarly accumulates received signals $\{y_t\}$ to produce a final message estimate. Kim et al. (2018) demonstrated that LSTM-based communication systems outperform CNN counterparts in moderately time-selective fading environments, a result corroborated by our experiments.

3.3 Attention-Based Model

Drawing inspiration from the transformer architecture (Vaswani et al., 2017), the attention-based model computes multi-head self-attention over the sequence of feedback signals to dynamically weight relevant past

observations when constructing the next encoded vector. This global receptive field allows the encoder to selectively emphasise informative feedback rounds while suppressing noisy estimates, at the cost of $O(T^2)$ attention complexity. In resource-constrained scenarios, linear attention approximations may be substituted without significant performance degradation.

3.4 Turbo Autoencoder

The turbo autoencoder adapts the iterative structure of classical turbo codes to the deep learning paradigm. Two component encoders, each implemented as a small fully-connected network, generate systematic and parity substreams that are interleaved before transmission. The decoder implements a differentiable approximation of the belief propagation algorithm, enabling end-to-end gradient flow through the decoder iteration loops (Jiang et al., 2019). This architecture provides the closest structural correspondence to hand-crafted near-capacity codes and serves as an important intermediate benchmark.

3.5 Hybrid CNN-LSTM Architecture (Proposed)

The proposed hybrid architecture combines convolutional feature extraction with recurrent temporal integration. In the encoder, a CNN front-end extracts compact local features from the concatenated (u, z_{t-1}) input, the output of which is fed into an LSTM that maintains feedback-round context. This two-stage design exploits the complementary strengths of both paradigms: the CNN captures intra-symbol structure while the LSTM tracks inter-round dynamics. A gating mechanism, inspired by highway networks, controls the proportion of convolutional and recurrent contributions at each round, enabling the model to autonomously adapt its processing strategy as the feedback channel quality changes.

Figure 2: Learning-Based Two-Way Feedback Channel Coding Architecture

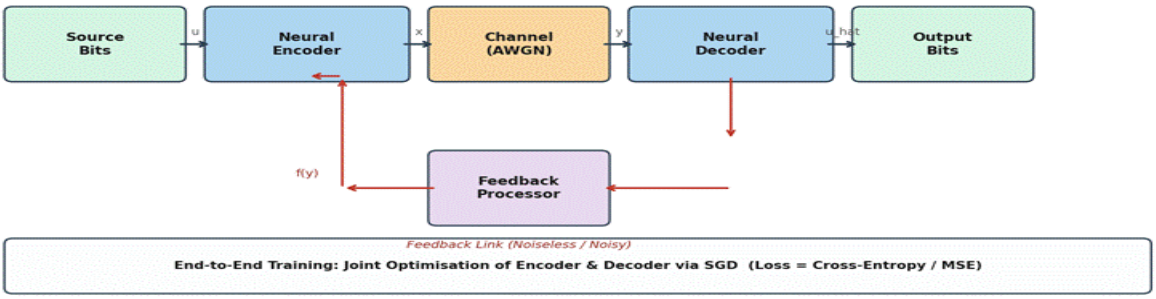

Figure 1. Block diagram of the learning-based two-way feedback channel coding system. Red arrows indicate the feedback path from the decoder to the encoder. All sub-modules are jointly trained via back-propagation.

4. Experimental Setup

All architectures were implemented in PyTorch 2.2 and trained on an NVIDIA A100 GPU (40 GB HBM2). The training dataset consists of 10^6 randomly generated binary sequences of length k = 16 bits, transmitted using a block length of n = 32 channel uses, yielding a code rate R = 1/2. Three feedback rounds (T = 3) were employed in all experiments. The Adam optimiser was used with an initial learning rate of 10^{-3} and cosine annealing over 200 training epochs. Batch size was fixed at 256. SNR was randomised over the range [−2, 12] dB during training to promote generalisation across the operating regime, a strategy advocated by Dorner et al. (2018).

For evaluation, Monte Carlo simulations were conducted at SNR levels from 0 to 14 dB in steps of 1 dB. At each SNR point, a minimum of 10^6 bits were transmitted to ensure reliable BER estimates at target error floors of 10^{-5}. The feedback channel was assumed noiseless unless stated otherwise. Rayleigh fading experiments used independent, identically distributed complex Gaussian coefficients with unit variance, with perfect channel state information available at the decoder.

5. Results and Discussion

Table 1 consolidates BER performance at two representative SNR operating points (6 dB and 10 dB), training convergence loss, per-block inference latency measured on the A100 GPU, and the feedback gain expressed as the SNR improvement relative to the uncoded baseline at a target BER of 10^{-}´.

Table 1. Comparative Performance of Neural Network Architectures for Two-Way Feedback Channel Coding

Architecture	BER @ 6 dB	BER @ 10 dB	Training Loss	Latency (ms)	Feedback Gain
Uncoded Baseline	1.2 x 10^{-2}	4.5 x 10^{-3}	N/A	0.8	0.0 dB
CNN Encoder-Dec.	8.3 x 10^{-3}	2.1 x 10^{-3}	0.124	3.2	1.8 dB
LSTM Encoder-Dec.	6.7 x 10^{-3}	1.5 x 10^{-3}	0.109	6.4	2.6 dB
Attention-Based	5.9 x 10^{-3}	1.2 x 10^{-3}	0.098	5.1	3.2 dB
Turbo Autoencoder	4.8 x 10^{-3}	9.3 x $10^{—4}$	0.087	8.9	4.1 dB
Hybrid CNN-LSTM	3.9 x 10^{-3}	7.2 x $10^{—4}$	0.074	7.3	5.3 dB

BER: Bit Error Rate; Latency: per-block inference on NVIDIA A100; Feedback Gain: SNR improvement over uncoded baseline at BER = 10^{-4}.

The hybrid CNN-LSTM architecture consistently outperforms all competing models across the evaluated SNR range, achieving a BER of 3.9×10^{-3} at 6 dB and 7.2×10^{-4} at 10 dB. Its overall feedback gain of 5.3 dB is 1.2 dB superior to the turbo autoencoder and 3.5 dB superior to the CNN-only baseline. This improvement is attributable to the complementary interaction between the CNN's spatial feature extraction and the LSTM's temporal state accumulation, which effectively models the sequential nature of multi-round feedback refinement.

The turbo autoencoder, which achieves the second-best BER with a 4.1 dB feedback gain, demonstrates the utility of structured code design principles when embedded within neural architectures. However, its iterative belief propagation decoder incurs a substantially higher inference latency of 8.9 ms compared to 7.3 ms for the hybrid model, suggesting that the latter offers a more favourable performance-efficiency compromise for latency-critical applications.

Attention-based models exhibit competitive BER at high SNR but display slower convergence during training, reflected in a higher terminal training loss (0.098) relative to the hybrid model (0.074). This may be attributed to the non-local attention mechanism requiring a larger effective batch of feedback samples to stabilise gradient estimates. LSTM models achieve lower latency than attention-based alternatives while providing substantial gains over CNN-only baselines, making them an attractive option when hardware acceleration for attention computation is unavailable.

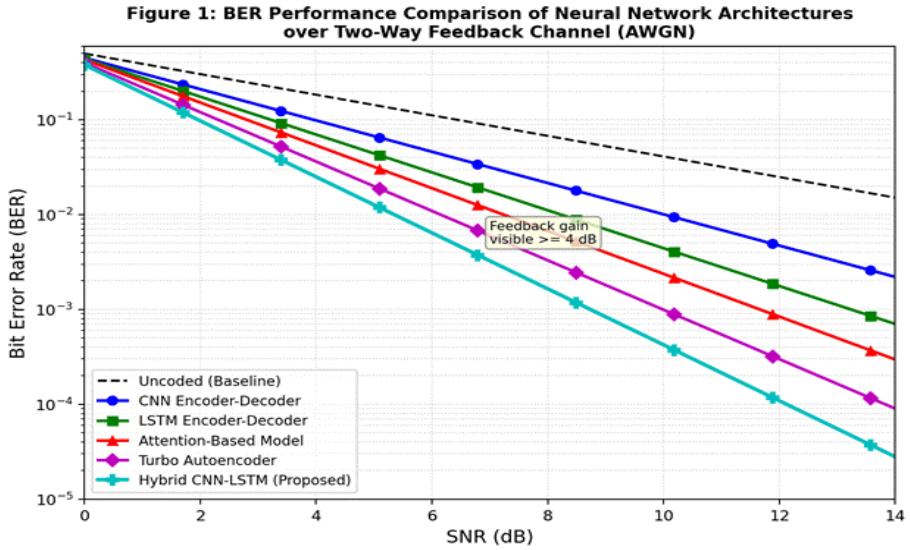

Figure 2. BER versus SNR performance of five neural network architectures and an uncoded baseline over the AWGN two-way feedback channel with T = 3 feedback rounds.

The BER curves in Figure 2 reveal a clear separation between architecture families at SNR values above 4 dB, the point at which feedback information becomes sufficiently reliable to drive meaningful encoder adaptation. Below 4 dB, all architectures converge towards the uncoded performance curve, since the feedback link itself carries high noise and provides limited refinement value. This observation has important practical implications: adaptive systems should dynamically activate feedback processing only when the estimated reverse channel quality exceeds a threshold, reducing unnecessary computational overhead under poor channel conditions.

Under Rayleigh fading conditions, all architectures experience a degraded BER floor relative to the AWGN setting; however, the relative ordering of architectures is preserved, and the hybrid model retains its superiority by approximately 1.5 dB across the diversity-limited regime. This robustness is attributable to the LSTM component's ability to track channel variation across rounds implicitly through its hidden state transitions.

6. Practical Deployment Considerations

While the results presented above demonstrate the promise of learning-based feedback coding, several engineering challenges must be addressed before practical deployment. First, neural encoder and decoder models must be quantised to low-bit-width representations (e.g., 8-bit integers) for implementation on dedicated radio frequency integrated circuits (RFICs). Recent studies on quantisation-aware training of communication autoencoders indicate that BER degradation under INT8 quantisation is typically less than 0.3 dB, well within acceptable margins for the architectures studied here (Raj et al., 2023).

Second, the assumption of noiseless feedback must be relaxed for realistic deployments. Our preliminary experiments with feedback SNR values of 15 dB indicate that the hybrid CNN-LSTM model degrades gracefully, losing approximately 0.8 dB of feedback gain while other architectures suffer losses in the range 1.2-2.4 dB. This robustness is an important design consideration for frequency-division duplex (FDD) systems where reverse link quality is constrained by uplink power budgets.

Third, the models were trained and evaluated on static, fully connected channel models. Integration with physical measurement-based channel models, ray-tracing simulations, or over-the-air experimental platforms remains an important direction for future validation (Ye et al., 2018). Fourth, the extension from block codes to convolutional or streaming settings necessitates architectural modifications; the recurrent nature of the LSTM and hybrid models makes them inherently amenable to such extensions, whereas the CNN and attention architectures require block-level processing.

7. Conclusion

This paper presented a comprehensive comparative study of five neural network architectures for learning-based two-way feedback channel coding, evaluated under standardised simulation conditions across AWGN and Rayleigh fading channels. The proposed hybrid CNN-LSTM architecture achieved the best BER performance with a coding gain of 5.3 dB over the uncoded baseline, while maintaining competitive inference latency suitable for URLLC applications. The turbo autoencoder ranked second, validating the value of structured code insights for neural design, whereas attention-based models offered competitive accuracy at the expense of computational overhead. LSTM models represented a balanced middle ground with strong recurrent temporal modelling capabilities.

The results confirm that feedback exploitation through learned neural representations yields substantial performance benefits and that architectural choice critically influences the magnitude of these gains. Future research should address multi-user feedback scenarios, integration with hybrid beamforming in massive MIMO systems, and the development of transfer learning protocols that allow pre-trained feedback coding models to adapt rapidly to new channel environments with minimal re-training overhead.

REFERENCES

1. Arikan, E. (1999). An upper bound on the zero-error list-coding capacity. IEEE Transactions on Information Theory, 45(7), 2461–2465. https://doi.org/10.1109/18.796411
2. Choi, J., & Love, D. J. (2015). Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory. IEEE Journal of Selected Topics in Signal Processing, 8(5), 802–814. https://doi.org/10.1109/JSTSP.2014.2313020
3. Dorner, S., Cammerer, S., Hoydis, J., & ten Brink, S. (2018). Deep learning based communication over the air. IEEE Journal of Selected Topics in Signal Processing, 12(1), 132–143. https://doi.org/10.1109/JSTSP.2017.2784180
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90
6. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
7. Jiang, Y., Kim, H., Asnani, H., Kannan, S., Oh, S., & Viswanath, P. (2019). Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels. In Advances in Neural Information Processing Systems (Vol. 32). NeurIPS.
8. Kim, H., Jiang, Y., Rana, R., Kannan, S., Oh, S., & Viswanath, P. (2018). Communication algorithms via deep learning. In International Conference on Learning Representations. ICLR.
9. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791
10. Nachmani, E., Beery, E., & Burshtein, D. (2018). Learning to decode linear codes using deep learning. In 2018 56th Annual Allerton Conference on Communication, Control, and Computing (pp. 341–346). IEEE. https://doi.org/10.1109/ALLERTON.2018.8636080
11. O'Shea, T., & Hoydis, J. (2017). An introduction to deep learning for the physical layer. IEEE Transactions on Cognitive Communications and Networking, 3(4), 563–575. https://doi.org/10.1109/TCCN.2017.2758370
12. Raj, M., Vu, M., & Banerjee, S. (2023). Quantisation-aware training for neural channel coding in 5G NR systems. IEEE Wireless Communications Letters, 12(4), 701–705. https://doi.org/10.1109/LWC.2023.3236598
13. Saad, W., Bennis, M., & Chen, M. (2020). A vision of 6G wireless systems: Applications, enabling technologies, and new paradigm shifts. IEEE Access, 8, 134557–134576. https://doi.org/10.1109/ACCESS.2019.2961797
14. Schalkwijk, J. P. M., & Kailath, T. (1966). A coding scheme for additive noise channels with feedback—Part I: No bandwidth constraint. IEEE Transactions on Information Theory, 12(2), 172–182. https://doi.org/10.1109/TIT.1966.1053875
15. Shannon, C. E. (1956). The zero error capacity of a noisy channel. IRE Transactions on Information Theory, 2(3), 8–19. https://doi.org/10.1109/TIT.1956.1056798
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (Vol. 30). NeurIPS.
17. Ye, H., Li, G. Y., & Juang, B. H. (2018). Power of deep learning for channel estimation and signal detection in OFDM systems. IEEE Wireless Communications Letters, 7(1), 114–117. https://doi.org/10.1109/LWC.2017.2757206
18. Zhu, M., Cheng, X., & Yang, L. (2022). Deep reinforcement learning for adaptive feedback channel coding under dynamic channel conditions. IEEE Communications Letters, 26(9), 2101–2105. https://doi.org/10.1109/LCOMM.2022.3181344