

Research on Intelligent Wireless Communication Algorithms Combining Reinforcement Learning and Error Coding

Shiyi Wang

*Department of Communication Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China
3389470800@qq.com*

Abstract. Wireless communication systems are afflicted by bit errors resulting from signal fading and environmental interference. Meanwhile, conventional error coding depends on fixed parameters that are unable to adapt to time - varying channel conditions. Existing RL-based adaptive coding schemes, though promising, either require modifications to the underlying hardware or assume computational resources beyond those available on typical IoT terminals—constraints that limit their practical deployment. This paper investigates the integration of reinforcement learning with error coding as a solution to this adaptability problem. Through literature review and theoretical analysis, we propose a layered fusion architecture that adds a lightweight Q-learning scheduling agent atop unmodified coding hardware. This design preserves hardware compatibility and enables firmware-based upgrades without replacement of existing equipment. The analysis validates the theoretical feasibility of this approach in achieving a balance between transmission speed and reliability. Meanwhile, it also pinpoints practical limitations, such as the absence of empirical validation and single-link restriction, which require further investigation prior to deployment in dense IoT scenarios.

Keywords: wireless communication, error coding, reinforcement learning, Q-learning, lightweight adaptive architecture

1. Introduction

Wireless communication has become ubiquitous in both daily life and industrial applications. However, wireless signals are easily affected by attenuation, multi-path fading, and electromagnetic noise, which corrupt transmitted bits and make error control necessary.

Error coding addresses this problem by adding redundant check bits. Traditional schemes like block codes and convolutional codes fix their parameters at the design stage and never change them afterward. Reinforcement learning, particularly Q-learning, offers a possible solution for wireless devices with limited resources since it mainly involves table look ups and simple calculations.

Several studies have explored RL-based adaptive coding. Wu and Liu [1] used Q-learning to adjust coding rates but assumed ideal feedback. Li [2] applied RL to forward error correction, but his method required hardware changes. Huang [3] showed Q-learning can adjust redundancy, and Liu [4] focused on lightweight designs for industrial use. Ferreira et al. [5] took into account deep-space scenarios with constraints similar to ours. However, most existing solutions either modify the

hardware or demand more computing power than typical IoT devices can provide. A framework that works with existing hardware and can be updated through firmware alone has not been fully explored.

To fill this gap, we propose a two-layer structure where the lower layer handles error coding and the upper layer uses Q-learning to choose redundancy settings. Our main contributions are as follows: (1) Recognizing the issue of fixed coding parameters in changing channels; (2) Designing a layered framework with a closed-loop mechanism; and (3) Assessing its anticipated benefits and current limitations. This is a theoretical study and does not include simulations or experiments.

Section 2 discusses drawbacks of conventional coding. Section 3 presents our proposed fusion framework. Section 4 covers limitations and corresponding optimization directions. Section 5 concludes the paper.

2. Limitations of conventional error coding

2.1. Strengths of traditional coding

Error coding has evolved over decades into a well-established technical system. Three practical advantages account for its continued prevalence in commercial wireless devices.

First, standardization ensures compatibility. Communication protocols specify fixed coding parameters for block and convolutional codes, allowing manufacturers to implement proven hardware modules without the need for custom design. Devices from different vendors can interoperate seamlessly, which reduces production costs and simplifying quality assurance.

Second, computational simplicity leads to a significant reduction in energy consumption. Traditional coding relies on elementary bit-wise operations, such as XOR, shifts, and table lookups, which can be executed efficiently on the least powerful processors. Battery-powered sensor nodes are capable of sustaining error correction for extended periods without requiring processor upgrades.

Third, performance is predictable in stable environments. When channel noise statistics remain approximately constant, fixed redundancy maintains the bit error rate within acceptable bounds. Network operators rarely need to adjust parameters after deployment, thereby reducing operational overhead.

2.2. Failure modes in time-varying channels

While these advantages make traditional coding attractive for many applications, they come with significant drawbacks when the channel conditions are no longer stable. Modern wireless applications operate in environments with rapidly fluctuating channel quality, such as vehicular networks, underground mining, outdoor mobile sensing, and dense urban deployments. In such scenarios, fixed coding exhibits three interrelated failure modes.

There is a mismatch between redundancy and channel state. When the interference is weak, high-redundancy coding leads to unnecessary waste of bandwidth. When sudden interference occurs, low-redundancy coding provides insufficient protection, resulting in a sharp increase in bit error rates. Subsequently, the system has to rely on upper-layer re-transmissions, which incurring latency penalties that can exceed 100 milliseconds per retry. This is unacceptable for real-time control or streaming applications [2].

Inefficient spectrum utilization is evident. The constant redundancy overhead, regardless of channel quality, implies that a significant portion of the transmitted bits do not convey any new information. In vehicular and outdoor IoT scenarios, where channel conditions change within

seconds, the spectral efficiency, which is defined as the number of information bits per second per Hertz, drops far below the theoretical capacity.

Limited resilience to burst interference exists. Fixed codes possess pre-determined error-correction bounds. When short-term electromagnetic interference produces error bursts that exceed the code's capability, a large number of packet losses occur. Multiple re-transmissions can lead to channel congestion, which in turn triggers cascading failures. Although some systems support manual mode switching, most commercial equipment lacks automatic real-time adjustment [6].

These limitations motivate our proposed fusion framework. The key requirement is not to abandon traditional coding, which still holds value due to its compatibility and efficiency, but rather to enhance it with adaptive decision-making capability.

3. Proposed fusion framework

This section presents our proposed fusion framework in three parts. We begin with the overall architecture and design philosophy, explaining why we chose a layered structure and how it balances compatibility with adaptivity. We describe the closed-loop operational logic, thoroughly walking through the four phases that facilitate real-time coding decisions without human intervention. Finally, we analyze the theoretical advantages of this framework over conventional fixed coding, covering adaptive channel matching, reduced re-transmission latency, and minimal deployment barriers. Collectively, these three parts demonstrate that the framework is not only theoretically sound but also practical enough to be deployed on existing hardware through firmware updates.

3.1. Overall architecture and design philosophy

We propose a layered architecture that separates low-level error correction from high-level adaptive scheduling. The core insight is to keep the existing coding hardware intact, thereby preserving its energy efficiency, compatibility, and predictability. Meanwhile, a software-based RL agent is added to select coding parameters on a per-transmission or per-slot basis.

The architecture comprises two distinct layers. The bottom layer consists of conventional error coding hardware or firmware that implements block or convolutional encoding and decoding. This layer takes redundancy parameters, such as code rate or the number of parity bits, as configuration inputs and conducts the actual bit manipulation. Critically, this layer remains unchanged from standard implementations, ensuring that the fusion scheme does not disrupt existing communication protocols or require chip-level redesign. The top layer is a Q-learning agent that observes channel state indicators and outputs optimal redundancy settings. This agent operates at a slower timescale compared to the physical transmission rate. For instance, it updates decisions every 100 milliseconds or every N packets to avoid excessive computation.

This separation yields a practical upgrade path: existing devices need only a firmware update to incorporate the RL scheduler, requiring no hardware replacement. We consider this backward compatibility a critical design constraint, as large installed bases of 4G, Bluetooth, LoRa, and ZigBee devices could potentially gain intelligent adaptation without capital expenditure on new hardware [7].

Several design considerations informed this architecture. First, the decision frequency must match the channel variation rate. If the decision frequency is too slow, it will miss transient interference; if it is too fast, it will waste computation during stable periods. We recommend configuring the scheduling interval based on the expected coherence time of the deployment environment. Second, the agent's output should be restricted to a small set of feasible redundancy

levels (e.g., three options: high, medium, and low). This is done both to simplify the learning process and to ensure compatibility with the limited configuration interfaces of existing coding hardware. Third, the learning process must tolerate delayed or imperfect feedback, since channel measurements and acknowledgment packets may arrive with latency.

This scheme differs from existing approaches in two important respects. In contrast to Li's work [2], which required modifications to the coding hardware itself, our framework operates at the software level. In contrast to Wu and Liu's design [1], which presupposed an idealized feedback channel, we explicitly take into account realistic constraints such as delayed acknowledgments and quantized measurements.

3.2. Closed-loop operational logic

The fusion system operates as a continuous closed loop comprising four phases: environment perception, intelligent decision-making, coding execution, and feedback optimization. No human intervention is required after initialization.

Phase 1: Channel perception. The communication system continuously gathers channel state information, including the signal-to-noise ratio, noise intensity, packet error rate, or signal attenuation, and quantifies these measurements into a compact representation that is suitable for the RL agent. This perception window must be long enough to obtain reliable estimates but short enough to track channel variations. In practice, a sliding window of the last 10 to 20 packets provides a reasonable trade-off.

Phase 2: Adaptive decision-making. The Q-learning agent utilizes its current Q-table to map the perceived channel state to an action. This action is associated with a specific redundancy level: low redundancy is adopted when interference is weak (prioritizing throughput), and high redundancy is employed when interference is strong (prioritizing reliability). The agent balances exploration (trying sub-optimal actions to discover better policies) against exploitation (choosing the currently best-known action), typically using an ϵ -greedy strategy where the exploration probability decays over time.

Phase 3: Coding execution. The agent transmits the selected redundancy parameter to the bottom coding layer. The hardware then applies the corresponding error correction scheme. This is done by selecting among precomputed generator matrices, adjusting puncturing patterns, or choosing appropriate interleaving depths. After that, it transmits the encoded data over the wireless channel.

Phase 4: Feedback and learning. The receiving end reports the actual bit error rate and whether re-transmission occurred. The system converts these communication outcomes into a reward signal that reflects the dual objectives of throughput maximization and error minimization. The agent updates its Q-table using this reward, reinforcing successful decisions and discouraging unsuccessful ones. Over time, the agent accumulates experience and converges toward a policy that matches redundancy levels to channel conditions.

These four phases execute iteratively, forming a self-adaptive cycle. The closed-loop nature is of utmost importance: in the absence of feedback, the agent would be unable to learn; without perception, it would be incapable of contextualizing its decisions; without execution, the loop would merely remain theoretical. Each phase depends on the others, and the overall system's performance hinges on the coherence of the design across all four phases.

It is worth noting that the current framework supports only point-to-point single-device communication. Multi-terminal scenarios, in which co-channel interference leads to strategic interactions among multiple agents, necessitate an extension to the basic scheme, and we will discuss this in Section 4.

3.3. Theoretical advantages

The layered fusion architecture offers three principal theoretical advantages over conventional fixed coding.

Adaptive channel matching. Through the dynamic adjustment coding redundancy based on real-time channel observations, the system breaks the fixed trade-off that afflicts traditional schemes. In the case of weak interference, it automatically switches to higher code rates to maximize throughput; In the case of strong interference, it switches to lower rates to prevent errors. This adaptivity enables the system to track channel variations across time and to respond to both gradual drifts (e.g. diurnal interference patterns) and sudden changes (e.g. momentary obstructions). The theoretical foundation of this advantage lies in the RL agent's ability to approximate the optimal mapping from channel states to redundancy levels through iterative learning [1].

Reduced re-transmission latency. By applying stronger error protection precisely when it is needed—rather than uniformly or re-actively—the system reduces packet error rates during interference bursts. Fewer re-transmissions translate directly to lower average latency. This is particularly significant for ultra-reliable low-latency communication (URLLC) scenarios, such as industrial closed-loop control and real-time video streaming, where deadlines are measured in tens of milliseconds and re-transmissions may cause mission failure [6]. The reduction in re-transmissions also alleviates channel congestion, creating positive secondary effects on overall network throughput.

Minimal deployment barriers. The layered design, which solely adds software logic to the existing hardware, presents a practical advantage compared to proposals that necessitate chip - level modifications. Large installed bases of commercial wireless devices (4G modems, Bluetooth peripherals, LoRa sensors) could potentially gain intelligent adaptation through firmware updates alone, avoiding the capital expense and logistical complexity of hardware replacement [7]. For small and medium-sized enterprises with limited upgrade budgets, this low-cost path to intelligent coding is particularly attractive.

These theoretical advantages, however, should be interpreted with caution. They stem from the architecture's structural properties and the known convergence guarantees of Q-learning, but they have not been empirically validated. The magnitude of performance improvement, whether it is 5% or 50%, remains unknown without simulation or hardware testing.

4. Limitations and optimization directions

4.1. Absence of empirical validation and simulation-based evaluation

The current study is deliberately limited to theoretical investigation and does not include simulation or experimental validation. Consequently, the proposed framework cannot yet provide quantitative evidence regarding improvements in bit error rate, throughput, or latency. Although this design choice allows the paper to focus on architectural feasibility, it also limits the practical persuasiveness of the proposed approach.

Future research should therefore prioritize simulation-based evaluation. A discrete-event simulator with Rayleigh fading channels and configurable interference scenarios could be developed to compare the proposed method with conventional fixed coding schemes. Quantitative evaluation would provide stronger evidence for the framework's effectiveness and support subsequent hardware implementation.

4.2. Single-link restriction and multi-agent coordination

The proposed framework currently addresses only point-to-point communication. In practical IoT networks, multiple devices often compete for shared wireless resources, resulting in co-channel interference and interactions among different communication links. An agent optimizing independently for each link may inadvertently degrade others' performance, as coding decisions affect the shared interference landscape. Extending to multi-agent coordination—where agents share information or learn joint policies—is theoretically nontrivial and introduces challenges such as non-stationary, communication overhead, and policy conflicts [8].

For dense networks, we envision a hierarchical architecture where a central coordinator collects channel state information from all terminals, computes joint coding policies, and distributes redundancy assignments to each node. This would mitigate co-channel interference while preserving the local lightweight nature of per-device scheduling. Alternatively, independent agents could share Q-tables or exchange information about congestion levels to achieve coordination without a central entity.

4.3. Increased computational and energy overhead and lightweight optimization

The RL scheduler consumes processor cycles that would otherwise remain idle. Each decision cycle requires SNR quantization, Q-table lookup, reward computation, and a Q-value update. For button-cell-powered sensors with micro-joule energy budgets, this additional burden may shorten battery life. The severity of this impact depends on the scheduling frequency and the specific hardware capabilities [4].

We aim to further simplify Q-learning's state representation (e.g., from three states to two) and action space (e.g., from three redundancy levels to two) to reduce computational and energy overhead. We also plan to explore event-triggered updates, where the agent recomputes decisions only when channel conditions change beyond a threshold, rather than at fixed intervals. These modifications require careful validation to ensure that performance is not compromised.

5. Conclusion

This paper has presented a theoretical investigation into integrating reinforcement learning with error coding for wireless communication. Following the structural requirements of our research program, we incorporated the necessary background on wireless channels and Q-learning into a compact introduction, reserving the bulk of the discussion for our core contributions.

We analyzed why traditional fixed-parameter coding fails in time-varying channels, identifying three key failure modes: inefficiency induced by mismatch, wasted spectrum, and burst intolerance. Subsequently, we put forward a layered fusion architecture. This architecture adds an RL-based scheduling agent on top of unmodified coding hardware. We elaborated on the closed-loop operational logic and carried out a comprehensive theoretical evaluation. Our analysis confirms that the scheme can, in principle, adapt to changing interference conditions while preserving hardware compatibility, a combination that has rarely been addressed in prior work.

Nevertheless, our proposal remains an architectural design, not an engineered system. The absence of simulation data, the restriction to single-link scenarios, and the energy overhead constitute significant gaps that we have candidly acknowledged. We have delineated specific steps, namely simulation, multi-agent extension, and lightweight optimization, to direct subsequent research.

Overall, the pairing of lightweight reinforcement learning with error coding holds genuine theoretical promise for intelligent wireless communication. The challenge ahead lies not in proving the concept, but in bridging the gap between architectural vision and field-ready implementation.

More broadly, this work aligns with the growing trend of embedding intelligence into physical-layer communication, which is seen as a key direction for 6G and next-generation IoT. While high-end systems may adopt deep learning approaches, lightweight algorithms such as the one presented here will still be essential for low-power devices, which make up the majority of wireless terminals. We hope our framework contributes a useful reference for future development of adaptive coding technologies.

References

- [1] Wu, S., & Liu, H. (2023). Q-learning based adaptive coding strategy for time-varying wireless channels. *Applied Sciences*, 13(19), 10872. <https://doi.org/10.3390/app131910872>
- [2] Li, L. (2024). Reinforcement learning optimized adaptive forward error correction for wireless IoT transmission. *Electronics*, 13(15), 3052. <https://doi.org/10.3390/electronics13153052>
- [3] Huang, W. (2024). Q-learning based adaptive redundancy adjustment for wireless error correction coding. *arXiv preprint arXiv:2407.12583*. <https://arxiv.org/pdf/2407.12583>
- [4] Liu, Z. (2024). Lightweight reinforcement learning for dynamic error control coding in industrial wireless networks. *Information*, 15(8), 462. <https://doi.org/10.3390/info15080462>
- [5] Ferreira, L., Ferreira, M., & Oliveira, R. (2025). Reliable transmission of LTP using reinforcement learning-based adaptive FEC. *arXiv preprint arXiv:2506.22470*. <https://arxiv.org/pdf/2506.22470>
- [6] Yu, H., & Yang, H.-C. (2025). Adaptive cooperative transmission design for ultra-reliable low-latency communications via deep reinforcement learning. *arXiv preprint arXiv:2511.02216*. <https://arxiv.org/pdf/2511.02216>
- [7] Zhang, Y., & Wang, X. (2024). RL-ANC: Reinforcement learning-based adaptive network coding in the ocean mobile internet of things. *Journal of Marine Science and Engineering*, 12(6), 998. <https://doi.org/10.3390/jmse12060998>
- [8] Chen, Z. (2024). Dynamic FEC allocation via multi-agent reinforcement learning for dense IoT networks. *arXiv preprint arXiv:2409.08761*. <https://arxiv.org/pdf/2409.08761>