



HAL
open science

Improved Non-Uniform Constellations for Non-Binary Codes Through Deep Reinforcement Learning

Rami Klaimi, Stefan Weithoffer, Charbel Abdel Nour

► **To cite this version:**

Rami Klaimi, Stefan Weithoffer, Charbel Abdel Nour. Improved Non-Uniform Constellations for Non-Binary Codes Through Deep Reinforcement Learning. SPAWC 2022: IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication, Jul 2022, Oulu, Finland. 10.1109/SPAWC51304.2022.9834022 . hal-03668962

HAL Id: hal-03668962

<https://hal.science/hal-03668962>

Submitted on 16 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improved Non-Uniform Constellations for Non-Binary Codes Through Deep Reinforcement Learning

Rami Klaimi, Stefan Weithoffer and Charbel Abdel Nour
 IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
 e-mail: firstname.surname@imt-atlantique.fr

Abstract—Non-binary forward error correction (FEC) codes have been getting more attention lately in the coding society thanks mainly to their improved error correcting capabilities. Indeed, they reveal their full potential in the case of a one-to-one mapping between the code symbols over Galois fields (GF) and constellation points of the same order. Previously, we proposed non-binary FEC code designs targeting a given classical constellation through the optimization of the minimum Euclidean distance between candidate codewords. To go a step further, a better Euclidean distance spectrum can be achieved through the joint optimization of code parameters and positions of constellation symbols. However, this joint optimization for high order GFs reveals to be intractable in number of cases to evaluate. Therefore in this work, we propose a solution based on the multi-agent Deep Q-Network (DQN) algorithm. Applied to non-binary turbo codes (NB-TCs) over GF(64), the proposal largely improves performance by significantly lowering the error floor region of the resulting coded modulation scheme.

Keywords—Non-binary, turbo codes, coded-modulation, high order constellation, reinforcement learning, deep learning

I. INTRODUCTION

Supporting ultra-high throughput and highly reliable communications is becoming essential to enable emerging beyond-5G scenarios such as the ultra-reliable low latency communications (URLLC) and the internet of things (IoT). Recently, the European research project EPIC (*Enabling Practical Wireless Tb/s Communications with Next Generation Channel Coding*) [1] has tackled the development of close to 1 Tb/s channel decoder implementations for a wide range of forward error correction (FEC) codes [2]–[5]. Meanwhile, improved code design is a continuous process where non-binary (NB) codes were found to achieve improved coded-modulation capacities [6] for wireless communications and to represent an appealing solution when targeting better error correction and higher spectral efficiencies for short frame sizes.

NB low-density parity-check (NB-LDPC) and polar codes were first introduced in [7] and [8] respectively, improving error correction in comparison with their binary counterparts. Recently in [9], NB convolutional codes (NB-CCs) were optimized through the maximization of the cumulated Euclidean distance (CED) between codewords when mapped to quadrature amplitude modulation (QAM), having the same order, q , of the considered GF. Based on this NB-CC new design, NB turbo codes (NB-TCs) were proposed [9], [10] with their share of improvement when associated with high order modulations [11]. Then, low-complexity decoding algorithms

were introduced for NB-TCs [12], [13], paving the way for practical hardware implementations.

The application of a non-uniform distribution for the distances between constellation symbols can bring its share of improvements for a given NB-CC when adapted to the distance spectrum of the code. In this case, a considerable number of possible constellation shapes should be enumerated in order to find the best non-uniform constellation for a given NB-CC. However with increasing constellation sizes, this enumeration becomes computationally intractable.

On another note, the advancement in machine learning (ML) techniques was shown to offer solutions to complex mathematical problems through application dependent models. Indeed, ML has found its way to the telecommunications society, by replacing existing systems with deep neural networks (DNN) [14], as well as by improving the design of FEC codes [15], [16]. In this paper, we apply ML techniques in order to provide viable solutions to the problem of non-uniform constellation design dedicated for NB-codes.

This paper is structured as follows: Section II presents the structure of the used NB-TC in our study. The description and advantages of the considered Deep Q-learning (DQN) algorithms are provided in Section III. An application example of the DQN to the design of a non-uniform 64-QAM constellation is detailed in Section III-C. Corresponding results are shown in Section IV. Section V concludes the paper.

II. STRUCTURE OF NB-TCs

The proposed techniques in this paper can be applied to the association of any NB code and high order constellation. As an application example, we will consider NB-TCs over GF(q), previously developed by the authors of this paper.

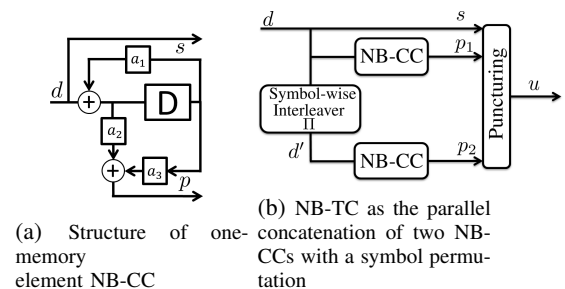


Fig. 1: Structure of the considered NB-CCs and TCs

Fig. 1 shows the structures of the one-memory element NB-CCs, and their parallel concatenation to constitute the considered NB-TCs. Systematic s and parity p symbols are mapped to a q -QAM constellation, of the same order as the considered GF(q) for the NB code. In Fig. 1a, a_1 , a_2 and a_3 are the GF(q) recursion and feed-forward coefficients designed to optimize the Euclidean distance spectrum conditioned by the q -QAM constellation. Without loss of generality, NB-TCs over GF(64) and 64-QAM constellations are considered in what follows. The primitive polynomial used to generate the considered GF(64) is $P_{\text{GF}(64)}(D) = 1 + D^2 + D^3 + D^5 + D^6$. The worst and the best NB-CCs over GF(64) in terms of their CED spectra for a uniform 64-QAM are provided in Table I. Denoted by C_1 and C_2 respectively, these codes are taken as comparison reference to validate the efficiency of the proposed techniques.

TABLE I: Best and worst obtained codes over GF(64), with the two first terms of the squared Euclidean distance spectra d_1^2 and d_2^2 and the corresponding multiplicities $n(d_1)$ and $n(d_2)$.

Code	C_1	C_2
(a_1, a_2, a_3)	(41, 2, 0)	(31, 5, 18)
d_1^2 (d_{min}^2)	0.38	1.52
$n(d_1)$	238422	652698
d_2^2	0.57	1.61
$n(d_2)$	230886	1084014

III. DEEP Q LEARNING ALGORITHM: APPLICATION TO CODE-MODULATION MAPPING

A. Reinforcement learning: introduction

Reinforcement learning (RL) as presented in [17] is a self-taught Markov Decision Process (MDP) constructed in a well-defined environment. This environment is constructed from four initial properties illustrated in Fig. 2:

- S_i represents the current state of the agent chosen from the set of states \mathcal{S} defined by the environment.
- A_i is the action applied at time i by the agent. The set of possible actions is defined by \mathcal{A} .
- $R_{A_i}(S_i, S_{i+1})$ represents the reward attributed to the agent once action A_i that changes its state from S_i to S_{i+1} has been taken.
- $P_{A_i}(S_u, S_v)$ denotes the probability of arriving to state S_v once action A_i has been taken from original state S_u at time i .

At the end of the learning/decision process, the goal of RL is to achieve the best reward R . To this end, the agent needs to discriminate between and choose actions in order to maximize the reward with the least amount of processing time.

B. Q-learning and Deep Q-learning: solving complex problems

The Q-learning algorithm [18] was proposed to train agents how to optimally act in a defined environment. While in a given state, S_i , the agent tries an action, A_i , evaluates its reward, $R_{A_i}(S_i, S_{i+1})$, and the resulting next state S_{i+1} .

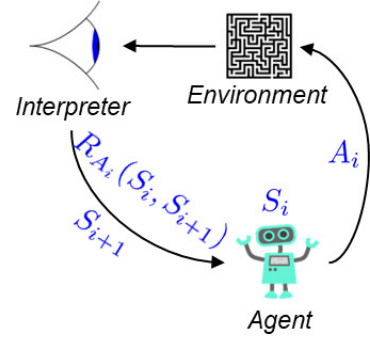


Fig. 2: RL environment with corresponding properties

After sufficiently observing the environment and the results of the set of all possible actions from all the states, the agent will be able to judge on the best overall actions to apply. These actions are taken in compliance with a policy π that maximizes the total discounted reward while considering the importance of the received reward at each time instance. This importance depends on the discount factor γ and the action-values $Q(x, a)$, where $x \in \mathcal{S}$, $a \in \mathcal{A}$ and is given by:

$$Q^{new}(x, a) = Q(x, a) + \alpha [R_a(x, y) + \gamma \max_b Q(y, b) - Q(x, a)] \quad (1)$$

where y is the resulting state from action a at state x and $\max_b Q(y, b)$ is the maximum future reward that could be obtained after arriving to state y by applying action b . The parameter α , with $0 < \alpha \leq 1$, denotes the learning rate.

The agent applies a greedy strategy. It explores the environment by choosing random actions, calculating their corresponding Q values then storing them in a table (*exploration phase*). In the following *exploitation phase* and for each state x , the agent relies on the Q -values from the table to choose the action resulting in the maximum reward. This exploration/exploitation strategy aims at steering the Q-learning algorithm towards achieving the highest total reward values.

The speed of convergence and the obtained reward values for the algorithm depend on the size of the Q -table and the number of exploratory random actions. In turn, these latter depend upon the sizes of the possible action and state sets. However for large sets of actions and/or states, creating and filling the Q -table become of high complexity both in terms of computations and memory requirements. As a possible solution, DQN based on replacing the Q -table by Convolutional Neural Networks (CNNs) was proposed in [19]. Trained using Eq. (1) and a set of input states and random actions, the CNN learns how to map an action to an input state in order to obtain the highest Q value.

C. Application to code modulation mapping: multi-agent DQN

In order to achieve the best coded modulation capacity, the output symbols of the proposed NB-TCs in Section II are to be mapped to a high-order constellation of the same order. In our case, NB-TCs over GF(64) are mapped to a

64-QAM constellation, initially respecting a gray mapping and shown in Fig. 3. As mentioned in Section II, the best NB-CC is the one maximizing the minimum CED respecting the positioning of the constellation symbols. Although the distance spectrum of the best NB-CC is already optimized for the considered labeling of the constellation symbols as in Fig. 3, this spectrum can be further optimized by adapting the distances between different constellation symbols to the code.

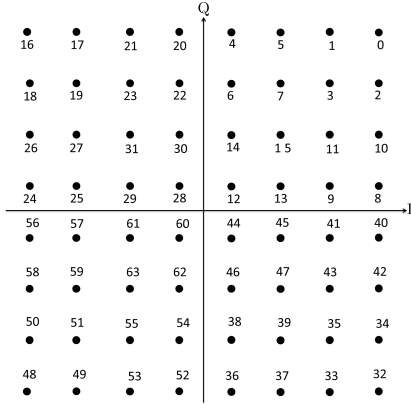


Fig. 3: 64-QAM constellation following a gray mapping

To achieve the best positioning for constellation symbols without modifying the mapping of Fig. 3, a large number of possible constellations needs to be evaluated. Indeed, due to the mutual dependence between constellation point positions and hence the resulting Euclidean distances, the simple application of a DQN would not be able to respect all imposed constraints. Therefore, we propose to apply the multi-agent DQN algorithm [20] to jointly optimize the positions of all constellation symbols. It involves training multiple DQN agents in parallel in order to jointly optimize a single total reward called the *team reward*. In our study case over $\text{GF}(64)$ and a 64-QAM constellation, we apply 64 DQN learning agents in parallel to optimize the minimum Euclidean distance of the NB-CC.

This proposal enables optimizing in parallel the minimum distance for each non-binary constellation symbol while considering joint effects for symbol positioning in terms of a constant total average energy per symbol and code spectrum. We recall that the minimum CED of a NB-CC can be calculated from short Diverging-Converging (DC) sequences as explained in [9]. Therefore, the contribution of each non-binary symbol to the minimum distance of the code can be assessed through the enumeration of all short DC sequences that have at least one of their transitions labeled by the NB symbol in question. To this end, we propose a coefficient $\delta(s_i), \forall s_i \in \text{GF}(q)$, that represents the contribution of the NB symbol s_i in the distance spectrum of the NB-CC:

$$\delta(s_i) = \frac{d_{\min}(s_i)}{n(d_{\min}(s_i))} \quad (2)$$

where $d_{\min}(s_i)$ is the minimum distance observed from short DC sequences having s_i as label of one of its transitions, and

$n(d_{\min}(s_i))$ is the number of occurrences of this minimum distance. Whenever the minimum distance, $d_{\min}(s_i)$ increases or its number of occurrences $n(d_{\min}(s_i))$ decreases, the effect of the NB symbol s_i on the distance spectrum and hence performance of the NB-CC is reduced: this corresponds to large values of $\delta(s_i)$. Consequently, the value of $\delta(s_i)$ gives an insight about the impact of s_i on the performance of the code. We use it as the reward function for the agent in-charge of choosing the position the symbol s_i in signal space. Moreover, the final goal is to optimize a *team reward* which we define as the maximization of $\min(\delta(s_i)), \forall s_i \in \text{GF}(q)$. By maximizing the minimum value of $\delta(s_i)$, we are able to optimize the spectrum of the resulting coded modulation (minimum distance and multiplicity) which represents one step further from the sole maximization of the minimum distance.

IV. NUMERICAL RESULTS

We have applied the multi-agent DQN algorithm to optimize the association of the code and the modulation following the proposed model in Section III-C. Constraints were introduced to the trained agents in order to apply actions that preserve the mapping in Fig. 3. In other words, if \mathbb{S}_i is the set of direct neighbors of symbol s_i in the constellation, the agent in-charge of choosing the best position for s_i should always choose actions that preserve \mathbb{S}_i .

A. Obtained constellations and impact on the NB-TCs

Fig. 4a shows a modified constellation that respects the mapping of a 64-QAM while maximizing the minimum distance of the NB-CC C_1 of Table I. Figures 4b and 4c show the impact of each NB symbol in the error prone sequences of the code computed according to Eq. (2). The value of $1 - \delta(s_i)$ is computed, normalized for each NB symbol s_i and associated with a color. The larger the impact of s_i on the spectrum of the code, the warmer its color and vice versa generating a heat map. Thanks to the application of the proposed method, we can observe from Fig. 4c that for the obtained non-uniform constellation, the effect of the majority of the constellation symbols is reduced by increasing the minimum distance in which they participate and/or by reducing their occurrence in error-prone sequences. Consequently, a mapping of the considered NB code symbols to the modified constellation of Fig. 4a can be expected to lead to an improved error correcting performance.

Similarly, Fig. 5a depicts the modified constellation that improves the distance spectrum of code C_2 . By comparing figures Fig. 5b and Fig. 5c, the negative impact of some NB symbols is shown to be mitigated thanks to the proposed constellation. Hence, error correcting performance resulting from a mapping to the modified constellation from Fig. 5a is expected to be superior to a mapping to conventional 64-QAM.

Note that since C_1 represents the worst code over $\text{GF}(64)$, its margin of improvement is larger than for code C_2 . Moreover, Fig. 4b shows that the q NB symbols do not have the

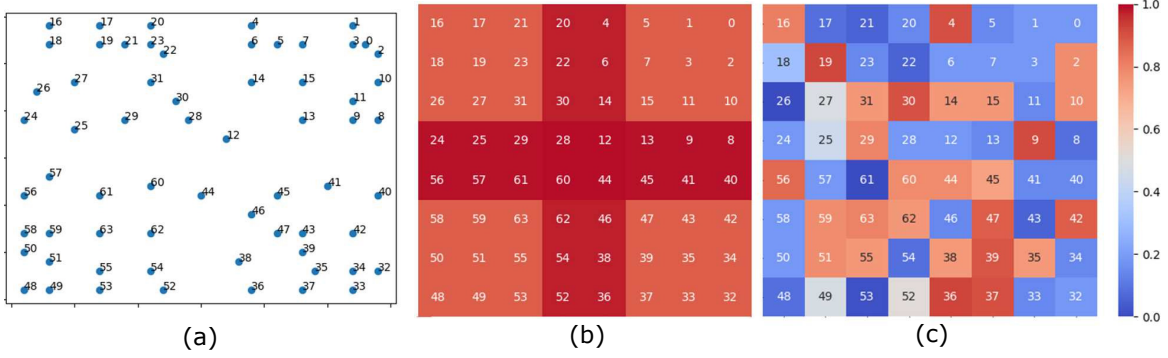


Fig. 4: (a) Optimized symbol position for 64-QAM constellation and code C_1 . (b) and (c) Heat-maps representing the contribution of each NB symbol in the minimum distance of the code for a conventional 64-QAM and for the modified 64-QAM, respectively.

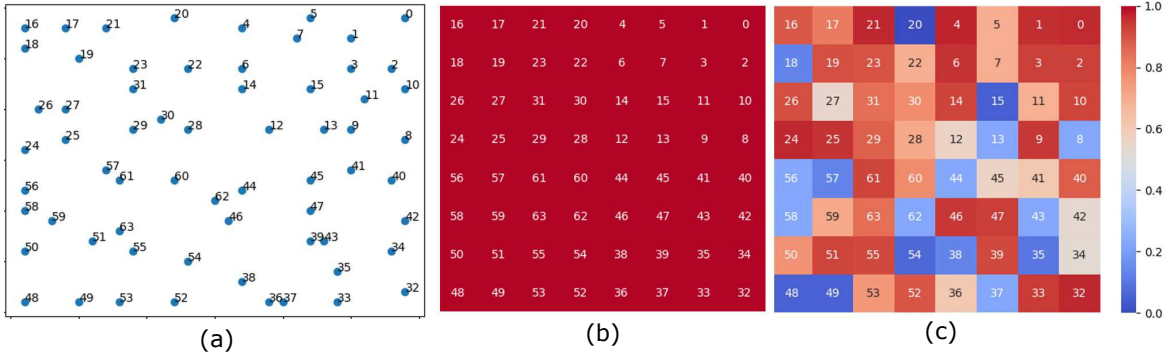


Fig. 5: (a) Optimized symbol position for 64-QAM constellation and code C_2 . (b) and (c) Heat-maps representing the contribution of each NB symbol in the minimum distance of the code for a conventional 64-QAM and for the modified 64-QAM, respectively.

same impact on the minimum distance of code C_1 leading already to a non-uniform distribution. However, this is not the case for code C_2 where all symbols equally participate to the minimum distance as seen in Fig. 5b. These facts explain the obtained results shown in Fig. 4c and Fig. 5c where the improvement for the heat map of code C_1 is larger than for code C_2 going from a classical to the proposed non-uniform QAM. Also, heat maps for code C_1 cannot be fairly compared with the ones of code C_2 since plotted values have been normalized to each code's minimum distance.

B. Performance evaluation of the proposed code-optimized non-uniform constellation

In order to assess the impact of the newly defined positions of the high-order constellation symbols, the error correcting performance of the considered NB-TCs must be observed.

In the case of the NB-TC using C_1 from Table I as a constituent code, Almost Regular Permutation (ARP) interleaver is used as a permutation function respecting the constraints of [10], [21]. A comparison between the error correcting performance of the resulting NB-TC mapped to the conventional 64-QAM constellation and to the modified 64-QAM of Fig. 4a is depicted in Fig. 6. When applying the modified constellation, the error floor region of the considered NB-TC is lowered by about one decade, achieving a Frame Error Rate (FER) of 10^{-7} .

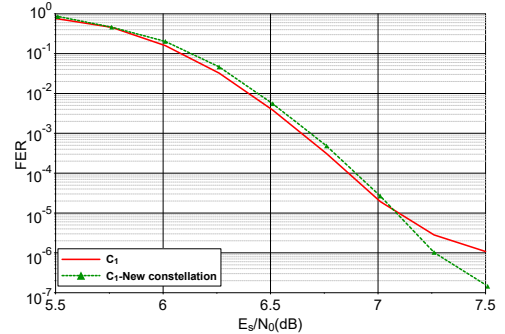


Fig. 6: Error correcting performance of NB-TCs constructed from code C_1 , with ARP interleaver, and comparison between two 64-QAM constellations

Fig. 7 shows the performance comparison between a conventional 64-QAM constellation and the proposed non-uniform constellation of Fig. 5a for the NB-TC with component code C_2 of Table I. Note, that since C_2 has the best CED spectrum, a very low error floor region in the order of 10^{-12} of FER can be achieved by the constructed NB-TC when considering algebraic interleavers as shown in [11]. In order to avoid long simulation times, we have simulated C_2 -based NB-TC using uniform interleavers [22]. The resulting higher error floor regions lead to a better observability of the following effect: the error floor region of the considered NB-

TC is shown to be interestingly lowered when mapped to the modified constellation as shown in Fig. 7. Indeed, this gain can be expected to be preserved or even increased when applying an ARP interleaver, lowering the expected error floor well beyond 10^{-12} .

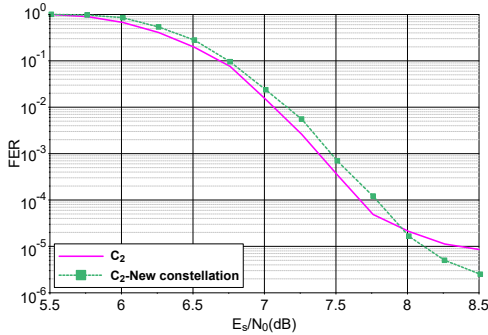


Fig. 7: Error correcting performance of NB-TCs constructed from code C_2 , with uniform interleaver, and comparison between two 64-QAM constellations

Having a worse distance spectrum than C_2 , NB-TCs constructed from C_1 achieve a better error correcting performance in the waterfall region compared to C_2 , independently from the used permutation function. Its main drawback remains the presence of a higher error floor in the order of 10^{-6} of FER, even when considered with a well constructed ARP interleaver. An optimized code-dependent constellation as proposed in this work reduces this error floor region by more than one decade of FER to an acceptable level. Indeed, the resulting FEC scheme may be preferable in scenarios that do not require a very low error floor, but rather a better performance at low SNR levels. On the other hand for scenarios requiring ultra-reliable communications, NB-TCs constructed from C_2 using ARP interleaver and associated to the optimized non-uniform constellation are expected to offer a reliable error correcting performance at least down to 10^{-13} of FER. In conclusion, the proposed approach for designing code-optimized constellations has the capacity to significantly lower the error floor region and can offer an additional trade-off for communication system designers to choose the appropriate code for a given communication scenario.

V. CONCLUSION

In this paper, we have proposed a new approach to optimize non-uniform constellations for NB codes, targeting better coded modulation capacities. In particular, deep Q-learning algorithms are chosen to solve the intractable problem of enumerating all possible candidate high-order constellations. On the example of NB-TCs, we show that designed non-uniform constellations improve the error correcting performance asymptotically. As a proof-of-concept, this example clearly illustrates how ML techniques can provide solutions to complex code design tasks. In a next step, we consider jointly optimizing the code construction and the positions of constellation symbols through DQN algorithms.

ACKNOWLEDGMENT

This work was partially funded by the French National Research Agency project QCSP (ANR-19-CE25-0013-01).

REFERENCES

- [1] "EPIC project: Enabling practical wireless Tb/s communications with next generation channel coding,," <https://epic-h2020.eu/>.
- [2] A. Süral, E. G. Sezer, Y. Ertuğrul, O. Arikan, and E. Arikan, "Terabits-per-second throughput for polar codes," in *IEEE 30th Intern. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, Istanbul, Turkey, Sept. 2019.
- [3] C. Kestel, L. Johannsen, O. Griebel, J. Jimenez, T. Vogt, T. Lehnigk-Emden, and N. Wehn, "A 506Gbit/s polar successive cancellation list decoder with CRC," in *IEEE 31st Intern. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, London, UK, Sept. 2020.
- [4] S. Weithoffer, R. Klaimi, C. Abdel Nour, N. Wehn, and C. Douillard, "Low-complexity Computational Units for the Local-SOVA Decoding Algorithm," in *IEEE 31st Intern. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, London, UK, Sept. 2020.
- [5] —, "Fully Pipelined Iteration Unrolled Decoders-The Road to Tb/s Turbo Decoding," in *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing-(ICASSP 2020)*, Barcelona, Spain, May 2020.
- [6] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, 1998.
- [7] M. C. Davey and D. J. MacKay, "Low density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, June 1998.
- [8] R. Mori and T. Tanaka, "Non-binary polar codes using Reed-Solomon codes and algebraic geometry codes," in *IEEE Information Theory Workshop*, Dublin, Ireland, Sept. 2015.
- [9] R. Klaimi, C. Abdel Nour, C. Douillard, and J. Farah, "Design of Low-Complexity Convolutional Codes over GF(q)," in *IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec 2018.
- [10] R. Garzón Bohórquez, R. Klaimi, C. Abdel Nour, and C. Douillard, "Mitigating correlation problems in turbo decoders," in *10th Intern. Symp. on Turbo Codes Iter. Inf. (ISTC)*, Hong Kong, China, Dec. 2018.
- [11] R. Klaimi, "Study of non-binary turbo codes for future communication and broadcasting systems," Ph.D. dissertation, Ecole nationale supérieure Mines-Télécom Atlantique, 2019. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-02543195/>
- [12] R. Klaimi, C. Abdel Nour, C. Douillard, and J. Farah, "Low-complexity decoders for non-binary turbo codes," in *10th Intern. Symp. on Turbo Codes Iter. Inf. Proc. (ISTC)*, Hong Kong, China, Dec. 2018.
- [13] H. Le Blevec, R. Klaimi, S. Weithoffer, C. Abdel Nour, and A. Baghdadi, "Low Complexity Non-binary Turbo Decoding based on the Local-SOVA Algorithm," in *11th Intern. Symp. on Turbo Codes Iter. Inf. Proc. (ISTC)*, Montréal, Canada, Sept. 2021.
- [14] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cognitive Commun. and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [15] L. Huang, H. Zhang, R. Li, Y. Ge, and J. Wang, "AI coding: Learning to construct error correction codes," *IEEE Trans. on Commun.*, vol. 68, no. 1, pp. 26–39, 2019.
- [16] H. Kim, S. Oh, and P. Viswanath, "Physical layer communication via deep learning," *IEEE J. on Sel. Areas in Inform. Th.*, vol. 1, no. 1, pp. 5–18, 2020.
- [17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.
- [18] C. J. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge, England, 1989.
- [19] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
- [20] P. Sunehag, G. Lever, A. Grusl, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.
- [21] R. Garzón-Bohórquez, C. Abdel Nour, and C. Douillard, "Protograph-based interleavers for punctured turbo codes," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 1833–1844, 2018.
- [22] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409–428, 1996.