Adaptive T-Detector exploiting Outer SISO Decoder in Time-Selective Channels

Adaptive T-Detector exploiting Outer SISO Decoder in Time-Selective Channels

Martina Magnaldi and Guido Montorsi
Dep. of Elec. and Telecomm. (DET), Politecnico di Torino, Italy
martina.magnaldi@polito.it, guido.montorsi@polito.it

***Abstract*—We present a trainable and adaptive detector (the T-Detector) with a core trainable BCJR (T-BCJR) that leverages the extrinsic information provided by an outer SISO decoder to enhance its tracking capability in time-varying channels. The T-Detector is designed to be channel- and modulation-agnostic, performing key detection tasks such as channel shortening, Maximum Likelihood (ML) sequence detection, and Log-Likelihood (LL) computation for soft-input channel decoding. Its processing complexity remains equal to that of conventional detection methods. The exploitation of the outer decoder enhances system performance and reduces pilot overhead.**

***Index Terms*—BCJR algorithm, model-based neural networks, Maximum Likelihood equalization, adaptive equalization, channel shortening, soft decoder, time-selective channel**

I. Introduction

Recent research has proposed the use of model-based neural networks (NNs) for digital receivers, aiming to combine the flexibility and adaptability of deep learning (DL) with the structure of traditional digital signal processing (DSP) blocks. These networks are data-trainable and can adapt to a variety of channel models, while maintaining a processing complexity comparable to that of conventional receivers. This approach addresses the limitations of purely data-driven DL techniques in physical layer signal processing [1], where traditional DSP blocks are entirely replaced by generic neural networks—often resulting in high computational complexity and limited interpretability. An excellent survey of model-based DL approaches is presented in [2], while [3] discusses key challenges faced by these emerging networks. Specific for the detector, we can mention [4], [5], and [6], [7], [8], which integrated NNs for channel state information (CSI) estimation into the traditional detection algorithms and filter estimators. Additionally, [9] proposes data augmentation for digital communication while [10] presents a meta-learning algorithm for fast adaptation to time-varying channels with minimal pilots. Taking into account the decoder in particular, [11] presents an extensive review of advances in deep learning-based decoding. Notable works on joint detection and decoding include [12], which proposes a neural network system optimized with bit-wise mutual information (BMI) for bit-metric decoding (BMD) and iterative demapping and decoding, and [13], which uses Graph NNs (GNNs) for joint equalization-decoding (JED) with a parallel flooding schedule channels with inter-symbol interference (ISI).

Within the same area of study, in [14], we proposed a trainable and adaptive detector for ISI channels which includes the T-BCJR ("T" stands for trainable) [15] as the core block. This T-Detector is based on the channel shortening approach [16], [17], using a convolutional linear layer to mimic the presence of the shortening filter before the T-BCJR. The proposed detector is completely trainable from its output LLs (binary or non binary), providing the adaptability needed to handle unknown channels, unknown constellation sets and potential mismatches. It incorporates the CSI estimation and it is suitable for static and slow time-varying channels where it achieves the performance of the optimal classical detector when suitable pilot density is used.

This work extends what has been presented in [14] by proposing a complete receiver scheme with the T-Detector followed by a soft-input soft-output (SISO) binary decoder. The T-Detector's output binary LLs, π_u^O, are fed to the SISO decoder, which processes them and outputs both recovered bits and extrinsic LLs, π_u^I. These extrinsic LLs are then used to train the detector. We employ Bit-Interleaved Coded Modulation (BICM) to facilitate the exchange of messages between the detector and the binary SISO decoder. It is important to clarify that we are not optimizing the decoder or using an iterative scheme. Our focus is on leveraging the extrinsic information from the decoder to train the detector, compensating for the lack of pilots and minimizing system overhead. This differs from [13], which performs joint optimization of the detector and decoder. Results are presented for a time-varying channel, considering different pilot densities, normalized Doppler frequencies and power control schemes.

The remainder of the paper is organized as follows. In Section II, we introduce the system model and present the T-BCJR and the T-Detector. Section III discusses how we exploit the outer SISO decoder. Simulation results for static and time-varying channels are presented in Section IV. Finally, Section V concludes the paper and outlines future work.

II. Background

A. The system model

Fig. 1 shows the block diagram of the considered system model. We assume a channel encoder and a modulator where $\mathcal{M}$ denotes the modulation set. The ISI time-varying channel is modeled as a tapped delay line (TDL) with a time-varying channel impulse response (CIR) h_k and additive white Gaussian noise. Hence, the discrete-time channel can

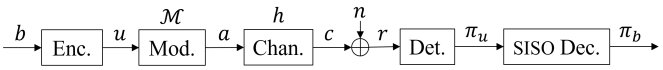


Fig. 1: Block diagram of the system model with an ISI channel.

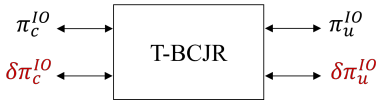


Fig. 2: The T-BCJR (formerly RNN BCJR).

be mathematically described by:

$$r_k = \sum_{l=0}^{L_C} h_{k,l} a_{k-l} + n_k \tag{1}$$

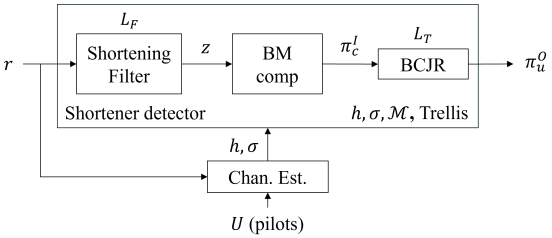
where L_C is the channel memory, a_k are the transmitted coded symbols, $n_k \sim \mathcal{CN}(0, \sigma^2)$ denotes the complex Gaussian noise, and $h_{k,l}$ denotes the time-varying complex channel coefficient.

B. The Recurrent Neural Network (RNN) BCJR

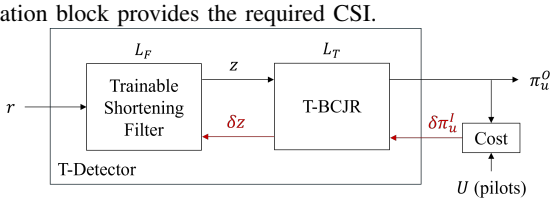
In [15], we introduced the RNN BCJR, a trainable version of the SISO additive BCJR algorithm. Its block diagram is shown in Fig. 2. The RNN BCJR propagation step coincides with the regular additive BCJR, as generalized in [18]. The output extrinsic LLs $\pi^O_{u/c}$, are obtained as a function of the input LLs $\pi^I_{u/c}$. The novelty of the RNN BCJR lies in that the parameters defining the BCJR model—specifically the trellis structure, input metric computation, and output LL computation—are all trainable, to minimize a cost function based on the output LLs. To this end, two ports are introduced to convey the sequence of cost function gradients for training the RNN BCJR ($\delta\pi^I$). In [15], we showed how to perform δ backpropagation to train the BCJR model and to deliver the output δ^O. Compared to other techniques proposed in the literature, the RNN BCJR offers several significant advantages: 1) the propagation step of RNN BCJR *is* the BCJR. There are no special-purpose integrated NNs that must be trained separately. As a result, the propagation step has the same complexity as the BCJR. 2) Starting from the cost function on the output LLs, the backpropagation recursions are computed throughout the RNN BCJR, ultimately generating the final output extrinsic deltas δ^O. This allows embedding the RNN BCJR in end-to-end trainable decoding networks, with propagation steps from observations to estimated LLs and backward training from output ground truth LLs back to train all concatenated blocks. The bidirectional nature of the training ports, similar to that of messages, allows to introduce the concept of iterative *trainable* decoding networks along the same lines as in [19].

C. The T-Detector

In [14], we showed how to use the RNN BCJR to construct a trainable detector for ISI channels based on the "channel shortening" approach [16][1].



(a) The detector based on channel shortening. An external channel estimation block provides the required CSI.



(b) The correspondent adaptive T-Detector.

Fig. 3: The shortener and the adaptive T-Detector.

In [14] and [15], we referred to our structures as RNN BCJR and RNN detector. However, the term "RNN" may be misleading, since these models are trainable versions of classical structures rather than traditional RNNs. To avoid confusion, we now refer to them as T-BCJR and T-Detector.

Fig. 3a shows a classical detector based on shortening. It includes a preliminary channel shortening filter of length L_F placed before a BCJR trellis processor with memory L_T. The "branch metric (BM) computation" block computes the metrics for the BCJR using the output z of the shortening filter as:

$$\pi^I_{c,k}(e) = \frac{1}{\sigma^2}\left[\Re(z_k a_k^*) - \frac{\gamma_0 \|a_k\|^2}{2} - \Re\left(a_k^* \sum_{i=1}^{L_T} a_{k-i}\gamma_i\right)\right].$$

This structure offers trade-offs between complexity and performance by tailoring the shortening filter memory L_F and the trellis memory L_T. Indeed, the shortening filter reduces the effective length of the CIR by mitigating the ISI, simplifying the following ML equalization based on trellis processing. When $L_T = 0$, meaning no trellis processing, the detector becomes the Minimum Mean Square Error (MMSE) filter, while when L_T equals the channel memory L_C, the detector becomes the optimal and more complex ML detector. In [16], the author derives the optimal shortener coefficients and metrics to use within the BCJR. These parameters depend on the impulse response of the channel h, the noise variance σ^2, and the modulation set $\mathcal{M}$. In a realistic time-varying scenario, an additional channel estimation block must be inserted to estimate the values of h and σ^2, from which the optimal filter coefficients and metrics must be computed. The periodicity of such estimation has to be chosen much smaller than the coherence time. To eliminate the need for

[1]In this reference, the author derives the structure assuming a MIMO channel model. The interpretation in a ISI setting is straightforward.

this additional block, we introduce our adaptive T-Detector, as illustrated in Fig. 3b. We replace the channel shortening filter with a trainable convolutional linear layer, denoted by $\mathbf{W}_{sh}$, that mimics its function. The "BM comp." block is now integrated within the T-BCJR module, which receives as input the signal z and directly computes the edge metrics ϵ_i using the trainable matrix $\mathbf{W}_C$ with an additional bias $\mathbf{b}_C$:

$$\epsilon_{i,k}(e) = \frac{1}{\sigma^2}\Re(z_k a_k^*) - \frac{1}{\sigma^2} f(e) \rightarrow \epsilon_{i,k} = \mathbf{W}_C z_k + \mathbf{b}_C.$$

In the context of ISI channels, the matrices $\mathbf{W}_{sh}$, $\mathbf{W}_C$, and $\mathbf{b}_C$—all extensively introduced in [14]—are trainable parameters. The propagation step has the same structure and complexity as the classical one. But now, all blocks are trainable from a cost function, defined on the delivered output of the T-BCJR.

The cost function we use for training is the instantaneous conditional entropy of the bit LLs delivered by the detector, assuming uniformly distributed inputs:

$$C(\pi_u^O) = \max_x{}^{*} \pi_u^O(x) - \pi_u^O(U), \tag{2}$$

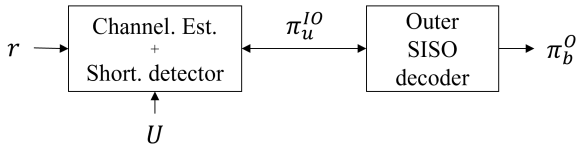
where U is the "ground truth", i.e. the transmitted input bit. By derivation of the cost function, we obtain the δ used for training:

$$\delta\pi_u^I = \nabla C(\pi_u^O) = \mathrm{AMAX}^*(\pi_u^O) - 1_U, \tag{3}$$

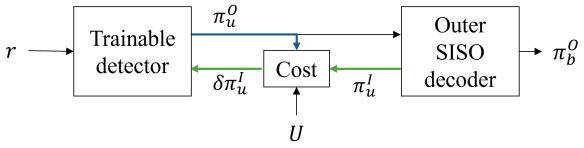
where 1_U denotes the one-hot column vector with a one in position U and $\mathrm{AMAX}^*(\pi_{\mathrm{u}}^{\mathrm{O}})$ is defined as $\left(\frac{e^{\pi(i)}}{\sum_j e^{\pi(j)}}\right)_{i=1}^{N}$. For the δ computation, we exploit interspersed pilots U, which are inserted with a given density and periodicity. In [14], we showed how to extend training from the T-BCJR to the channel shortening block. Our findings demonstrated that, with a sufficient number of interspersed pilots, the T-Detector can learn the optimal coefficients, with performance comparable to the classical detector with ideal CSI. After an initial training from scratch, it effectively adapts to slow time-varying channels with a reasonable pilot density.

III. Exploiting the outer SISO decoder

In modern receivers, an outer SISO decoder is typically employed at the detector's output to reliably decode transmitted information bits. The integration of an outer SISO decoder has led to the development of the concept of turbo equalization, also known as turbo detection [20]. In this approach, the detector iteratively uses the extrinsic information provided by the decoder, as shown in Fig. 4a. The presence of iterations makes the iterative detector more complex and gives some performance advantages. In most practical schemes, however, this possibility is not exploited as the use of proper binary Gray labeling and constellations allows for minimizing the loss of the non-iterative BICM schemes. In our implementation, as shown in Fig. 4b, the output LL of the T-Detector, π_u^O, feeds both the cost function and the following outer SISO decoder. The latter outputs on one side, the LL on the decoded bit, π_b^O, and, on the other side, the extrinsic LL, π_u^I, which provides new independent information. These extrinsic LLs



(a) The classical detector with outer SISO decoder. Iterative detection can use extrinsic information from the outer decoder.



(b) The T-Detector utilizing both pilots U and extrinsic information π_u^I from the outer decoder for training. No iterative scheme.

Fig. 4: The original and adaptive receivers with outer decoder.

are sent back and used in the cost function to calculate the backpropagation recursions, $\delta\pi_u^I$, and train the T-Detector. This new information is used when pilots are not present to guarantee a continuous training process and faster tracking in time-varying conditions. Eq. 3 now becomes:

$$\delta\pi_u^I = \begin{cases} \mathrm{AMAX}^*(\pi_u^O) - 1_U & \text{if pilots,} \\ \mathrm{AMAX}^*(\pi_u^O) - \mathrm{AMAX}^*(\pi_u^T) & \text{if no pilots.} \end{cases} \tag{4}$$

where $\pi_u^T = \pi_u^O + \pi_u^I$. We utilize ground truth when pilots are available. In the absence of pilots, we substitute the ground truth with the a posteriori probability (APP) π_u^T by combining the extrinsic information from the detector π_u^O with that from the decoder π_u^I. From the cost function, we can also compute $\delta\pi_u^O$, analogous to $\delta\pi_u^I$, which could potentially be used to train the decoder. However, this is unnecessary in our case. The cost function is inherently bidirectional, potentially enabling the joint training of both the detector and the decoder.

The training of the T-Detector, both in the initial acquisition and in the tracking phase, is performed as in classic detectors such as the adaptive stochastic gradient (ASG)-MMSE. At the transmitter, a pilot field of length P alternates with a data payload field of length N_d. The pilot density is defined as $\epsilon_P \triangleq \frac{P}{N_d+P} \leq 1$. The δ sequence used for training the detector is generated only at pilot positions and set to zero elsewhere. As explained in [15], the T-Detector is trained continuously, adapting to channel variations without separate training phases. At cold start, its weights are initialized randomly. Higher pilot density enables faster adaptation, while lower density reduces overhead at the cost of slower learning. The use of the extrinsic information from an outer decoder compensates for the lack of pilots supporting ongoing adaptation of the detector in dynamic channel conditions.

IV. Simulation Results

To test the proposed scheme with a realistic time-varying ISI channel model, we selected the Proakis-C channel with impulse response h = [0.227, 0.460, 0.688, 0.460, 0.227] and

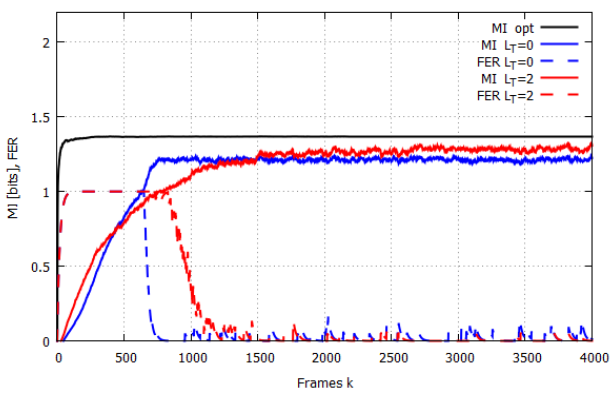


Fig. 5: Evolution of MI and FER in static channel. SNR = 6 dB.

length L_C = 5 as in [13]. The modulation set is the 4QAM with m = 2 bit per symbol. For this channel, the optimal ML detector based on the BCJR algorithm requires $2^{m(L_C-1)}$ = 256 states.

Leveraging the outer decoder's output to train the detector constrains the decoder block size. More precisely, because the T-Detector updating can only occur after an entire codeblock is received, the channel coherence time must be significantly longer than the duration of a codeblock transmission. To ensure this condition in all simulations, we pick a small (200,100) LDPC code (R_c = 1/2) from the 5GNR family as outer code.

In Fig. 5, we report, for the static channel, the time series of the running averages (RAs) of MI of LLs at the detector output (solid lines), and FER at the LDPC decoder output (dashed lines). Results are presented for the optimal ML detector (black curve) and the proposed T-detector with L_T = 0, 2 (1 and 16 states) and L_F = 10 (blue and red curves). The pilot density is set to ϵ_P = 0.1 and the SNR to 6 dB. Weights for the T-detector were randomly initialized. Initialization from scratch takes longer as L_T increases. Once the MI exceeds the threshold of MI = mR_c = 1, the outer decoder provides reliable extrinsic LLs. This is reflected by the FER curve dropping to zero, and, at the same time, the training process accelerates due to the greater availability of effective pilots. This mechanism is exploited as long as the MI of the detector remains above 1. Pilots are no longer required in this condition, and T-Detector updating can continue relying only on the decoder output.

In the time-varying scenario, the channel taps are replaced with time-varying complex Gaussian processes with Jake's spectrum, and Doppler frequency, normalized to the sampling rate, f_D. The power of each process is the square of its respective tap amplitude in h. In Fig. 6, we show the total BER and MI metrics for complete simulations, $f_D = 10^{-6}$ and SNR $\in [0, 10]$ dB. We report the performances of the reference ML detector with 256 states and perfect CSI (black lines), and those of the T-Detector (L_T = 2, L_F = 10) exploiting the extrinsic information (U1, solid lines) or relying

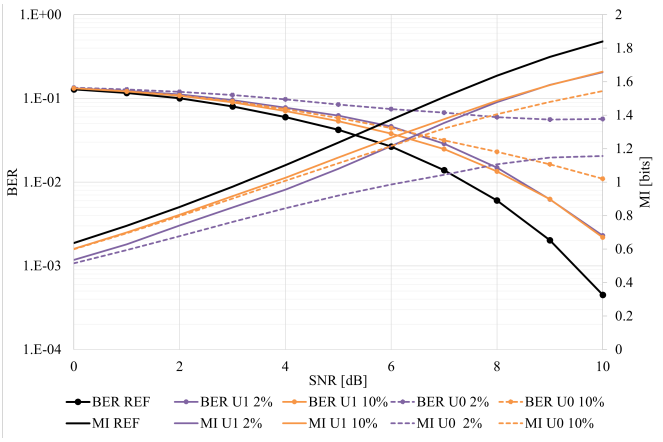


Fig. 6: BER (with markers) and MI (without markers) vs SNR for pilot density 2% (violet) and 10% (orange). $f_D = 10^{-6}$. In black the reference ML detector performances.

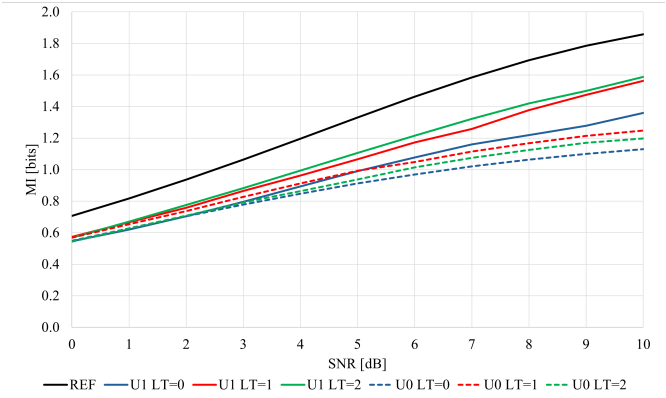


Fig. 7: MI vs SNR for pilot density 10%. $f_D = 10^{-5}$, L_T = 0, 1, 2. In black the reference performances of ML detector.

only on pilots (U0, dashed lines). With 10% pilot density (orange curves), leveraging the LLs from the outer decoder has minimal impact, as the high pilot density is sufficient to track channel variations effectively. In contrast, when the pilot density is 2% (violet curve), utilizing the LLs from the outer decoder provides a significant gain, especially for target MI larger than 1. The BER curves at the decoder output provide similar information as the MI curves, showing that BER threshold is closely related to the value where the MI exceeds the codulator rate. In Fig. 7, we show the MI metrics from complete simulations, $f_D = 10^{-5}$ and a fixed 10% pilot density. The results compare the reference ML detector with the adaptive T-Detector for L_T = 0, 1, 2 (1, 4, and 16 states), exploiting (U1) or not exploiting (U0) the extrinsic information from outer decoder. A significant performance degradation is observed across all L_T values when decoder output is not exploited. Furthermore, while the performance of the detector using decoder output improves with increasing L_T, the performance without decoder output degrades from L_T = 1 (red lines) to L_T = 2 (green lines), due to the low

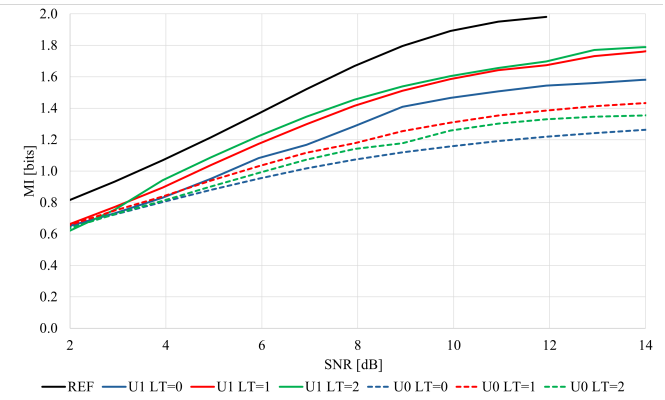


Fig. 8: MI vs SNR for pilot density 10%. $f_D = 10^{-5}$, $L_T = 0, 1, 2$. Power control is active.

pilot density and the larger number of coefficients to update.

These results show that leveraging LLs from the outer decoder enhances the T-Detector's tracking ability when MI exceeds the codulator rate. Conversely, when MI drops below this threshold, updates rely solely on pilots. In realistic, (slowly) time-varying system scenarios, link adaptation is typically adopted to avoid this situation—by adjusting the codulator rate or transmit power to keep the MI aligned with the codulator rate.

In Fig. 8, we report the simulation results in the same channel scenarios of Fig. 7, but with a power adaptation control ensuring constant received power. In this more realistic scenario, the observations remain consistent: using the outer decoder to support detector training reduces pilot overhead and lowers the receiver SNR threshold by at least 2 dB for target MI values above 1.

V. Conclusions

The T-Detector [14] is a trainable version of the classical detector based on the channel shortening approach. As such, it inherits its flexibility and allows for trade-off between complexity and performance. Unlike the classical version, the T-Detector does not require a separate channel estimation block and the computation of optimal coefficients, and it is robust to channel mismatch. The T-Detector's adaptability to channel variations relies on the overhead from interspersed pilots, which are used for continuous updates. To reduce pilot overhead, we have demonstrated in this paper that extrinsic information from an outer decoder can replace the need for pilots. We have shown how to compute the cost function from the extrinsic LLs provided by the external outer decoder and demonstrated the significant reduction in pilot density and SNR receiver threshold achievable with this approach. The proposed strategy has a minimal impact on the receiver complexity, as it only requires postponing the T-Detector updating after the decoding.

Acknowledgement

This work was supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP E13C22001870001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

References

[1] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[2] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proceedings of the IEEE*, 2023.

[3] T. Raviv, S. Park, O. Simeone, Y. C. Eldar, and N. Shlezinger, "Adaptive and flexible model-based ai for deep receivers in dynamic channels," *IEEE Wireless Communications*, 2024.

[4] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3319–3331, 2020.

[5] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Data-Driven Factor Graphs for Deep Symbol Detection," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2682–2687.

[6] N. Shlezinger, R. Fu, and Y. C. Eldar, "DeepSIC: Deep Soft Interference Cancellation for Multiuser MIMO Detection," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1349–1362, 2021.

[7] J. Yang, Q. Du, and Y. Jiang, "Neural network-assisted receiver design via learning trellis diagram online," *IEEE Transactions on Communications*, vol. 70, no. 12, pp. 8075–8085, 2022.

[8] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.

[9] T. Raviv and N. Shlezinger, "Data augmentation for deep receivers," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 8259–8274, 2023.

[10] T. Raviv, S. Park, O. Simeone, Y. C. Eldar, and N. Shlezinger, "Online meta-learning for hybrid model-based deep receivers," *IEEE Transactions on Wireless Communications*, vol. 22, no. 10, pp. 6415–6431, 2023.

[11] T. Matsumine and H. Ochiai, "Recent advances in deep learning for channel coding: A survey," *IEEE Open Journal of the Communications Society*, 2024.

[12] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, and S. Ten Brink, "Trainable communication systems: Concepts and prototype," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5489–5503, 2020.

[13] J. Clausius, M. Geiselhart, D. Tandler, and S. ten Brink, "Graph neural network-based joint equalization and decoding," in *2024 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2024, pp. 1203–1208.

[14] M. Magnaldi and G. Montorsi, "The RNN BCJR Detector in Time-Varying Channels," in *IEEE Wireless Communications and Networking Conference*. IEEE, Mar. 2025.

[15] G. Montorsi and B. Ripani, "RNN BCJR: a fully trainable version of the additive BCJR algorithm," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 3696–3701.

[16] F. Rusek and A. Prlja, "Optimal channel shortening for MIMO and ISI channels," *IEEE transactions on wireless communications*, vol. 11, no. 2, pp. 810–818, 2011.

[17] D. D. Falconer and F. Magee Jr, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell System Technical Journal*, vol. 52, no. 9, pp. 1541–1562, 1973.

[18] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A Soft-Input Soft-Output APP module for iterative decoding of concatenated codes," *IEEE Communications Letters*, vol. 1, no. 1, pp. 22–24, 1997.

[19] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks," *European transactions on telecommunications*, vol. 9, no. 2, pp. 155–172, 1998.

[20] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European transactions on telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.