Single-Step Online Adaptation of Modular Bayesian Deep Receivers with Streaming Data

Yakov Gusakov
School of ECE
Ben-Gurion University
Be'er-Sheva, Israel
gusakovy@post.bgu.ac.il

Osvaldo Simeone
Engineering Depratment
King's College
London, U.K
osvaldo.simeone@kcl.ac.uk

Tirza Routtenberg
School of ECE
Ben-Gurion University
Be'er-Sheva, Israel
tirzar@bgu.ac.il

Nir Shlezinger
School of ECE
Ben-Gurion University
Be'er-Sheva, Israel
nirshl@bgu.ac.il

Abstract

Deep neural network (DNN)-based receivers offer a powerful alternative to classical model-based designs for wireless communication, especially in complex and nonlinear propagation environments. However, their adoption is challenged by the rapid variability of wireless channels, which makes pre-trained static DNN-based receivers ineffective, and by the latency and computational burden of online stochastic gradient descent (SGD)-based continual learning. In this work, we propose an online learning framework that enables rapid low-complexity adaptation of DNN-based receivers. Our approach is based on two main tenets. First, we cast online learning as Bayesian tracking in parameter space, enabling a single-step adaptation which deviates from multi-epoch SGD. Second, we focus on modular DNN architectures enabling parallel and localized Bayesian updates, where each module maintains its own variational posterior. Simulations with practical dynamic communication channels demonstrate state-of-the-art error rate performance.

1 Introduction

The ever-increasing demand for wireless data motivates the integration of emerging technologies such as massive and holographic multiple-input multiple-output (MIMO) [1] and reconfigurable intelligent surfaces [2]. While these innovations expand network capacity and functionality, they substantially complicate receiver design, challenging classical model-based signal processing [3, 4]. A promising alternative is to incorporate deep neural networks (DNNs) into the receiver, enabling model-agnostic operation in unknown or nonlinear environments [5, 6].

Despite their potential, deep receivers face two fundamental challenges [7]: (i) wireless channels are inherently time-varying, quickly invalidating mappings learned offline; and (ii) devices operate under strict computational and power constraints, limiting frequent retraining. Conventional offline DNN training relies on large datasets and joint learning [5, 8], which struggle to generalize to unseen channel distributions. An alternative is *online learning*, which adapts the receiver continuously using data from current operating conditions [9]. Online learning solutions have been explored via compact model-based architectures [10–12], meta-learning [13, 14], change detection [15], protocol-aware self-supervision [16–18], data augmentation [19, 20], and Bayesian learning [21–24]. Yet, these

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

methods rely on numerous steps of stochastic gradient descent (SGD) variants, limiting adaptation within the coherence time of dynamic channels.

In this work, we introduce a novel online learning paradigm for deep receivers that combines *modular architectures* and *Bayesian DNNs*. By casting online training as *Bayesian tracking* of DNN parameters [25], recursive Bayesian filtering enables lightweight updates in response to time-varying channels. Modular designs composed of compact sub-networks allow *module-wise* updates, evaluated asynchronously in a pipelined fashion [26]. Finally, we treat received signals as *streaming data*, learning on each incoming pilot without aggregating data to enhance adaptivity.

Our contributions are summarized as follows:

- **Online learning as Bayesian tracking:** We formulate deep receiver adaptation as Bayesian tracking of time-varying DNN parameters via variational inference [27], enabling rapid updates.
- **Natural-gradient-based updates:** We leverage natural-gradient Bayesian learning [28], which corresponds to a single-step extended Kalman filter (EKF) update for real-time adaptation.
- **Complexity-aware modular adaptation:** Module-wise updates exploit the receiver's modular design for asynchronous, localized, and scalable online learning with minimal resource overhead.
- **Extensive experimental validation:** We demonstrate rapid adaptation and state-of-the-art performance under dynamic channels simulated with COST2100 [29] and QuaDRiGa [30].

2 System Model and Preliminaries

2.1 Communication System Model

Channel Model. We consider an uplink MIMO system with K single-antenna users transmitting to an N-antenna receiver. At time t, each user k sends a symbol $s_t^{(k)} \in \mathcal{S}$, drawn from a constellation $|\mathcal{S}| = 2^B$ that encodes B bits. Denote the corresponding bit vector as $\boldsymbol{b}_t^{(k)} \in \{0,1\}^B$, and define

$$\boldsymbol{b}_t = \left[(\boldsymbol{b}_t^{(1)})^\top, \dots, (\boldsymbol{b}_t^{(K)})^\top\right]^\top, \quad \boldsymbol{s}_t = [s_t^{(1)}, \dots, s_t^{(K)}] \in \mathcal{S}^K. \tag{1}$$

The received signal, $\boldsymbol{r}_t \in \mathbb{C}^N$, is characterized by a generic time-varying conditional distribution $\mathbb{P}_t(\boldsymbol{r}_t|\boldsymbol{s}_t)$ representing dynamic channel conditions. Unlike classical block-fading models [31], we allow the channel to vary smoothly within each allocation window of T transmissions, capturing continuous dynamics and channel aging effects [32]. Each allocation window comprises two phases:

1. *Synchronization:* A number T_{sync} of predetermined symbols are transmitted for initial calibration.
2. *Data transmission:* Unknown payload data symbols with scarce, periodically inserted pilots.

Deep Receiver. The demodulator is a DNN with parameters $\boldsymbol{\theta}_t$, mapping the received signal $\boldsymbol{r}_t$, to a vector $\boldsymbol{\ell}_t \in \mathbb{R}^{KB}$ of bit probabilities $\ell_{t,b}^{(k)} = p(b_{t,b}^{(k)} = 1|\boldsymbol{r}_t, \boldsymbol{\theta}_t)$.

2.2 Problem Formulation

We aim to design an online learning algorithm that updates $\boldsymbol{\theta}_t$ using incoming pilots $(\boldsymbol{r}_t, \boldsymbol{s}_t)$ to minimize the bit error rate, while tracking smoothly time-varying, possibly nonlinear channels under practical constraints of low latency and complexity.

2.3 Preliminaries

Bayesian Deep Learning. Bayesian neural networks (BNNs) treat parameters $\boldsymbol{\theta}$ as random variables with a prior distribution $p(\boldsymbol{\theta})$ [33, 34]. Based on the observed data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, variational inference approximates the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ through a parameterized distribution $q_{\boldsymbol{\psi}}(\boldsymbol{\theta})$, with parameters $\boldsymbol{\psi}$, which are optimized by minimizing the evidence lower bound (ELBO):

$$\mathcal{L}_{\mathcal{D}}^{\text{ELBO}}(\boldsymbol{\psi}) = -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}}} [\log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\theta})] + D_{\text{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \| p(\boldsymbol{\theta})), \tag{2}$$

where $D_{\text{KL}}(\cdot\|\cdot)$ denotes the Kullback-Leibler distance. During inference, J samples $\{\boldsymbol{\theta}_j\}_{j=1}^J$ can be drawn from distribution $q_{\boldsymbol{\psi}}$ to obtain an ensemble $\{p\left(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_j\right)\}_{j=1}^J$. BNNs reduce overfitting and yield well-calibrated uncertainty estimates [35].

Modular Deep Receivers. Structured neural architectures assign sub-networks to specific tasks [9, 26]. As a notable example, *DeepSIC* unfolds the iterative soft interference cancellation (SIC) algorithm into Q layers of K user-specific modules [11] with parameters $\boldsymbol{\theta}_t = \{\boldsymbol{\theta}_t^{(k,q)}\}_{k,q=1}^{K,Q}$ at time t. The module for user k at iteration q produces the soft estimates

$$\boldsymbol{\ell}_t^{(k,q)} = h_{\boldsymbol{\theta}_t^{(k,q)}}\big(\boldsymbol{x}_t^{(q)}\big), \quad \boldsymbol{x}_t = [\boldsymbol{r}_t^\top, (\boldsymbol{\ell}_t^{(1,q-1)})^\top, \ldots, (\boldsymbol{\ell}_t^{(K,q-1)})^\top]^\top \tag{3}$$

a function of the previous layer estimates of all K users. Final estimates for user k are $\boldsymbol{\ell}_t^{(k)} = \boldsymbol{\ell}_t^{(k,Q)}$. The modular design of DeepSIC naturally lends itself to localized and parallelizable updates.

3 Modular Online Bayesian Learning

The proposed solution departs from conventional multi-epoch SGD schemes, formulating online training of BNNs as the tracking of a *parameter-space dynamical system*, where the received pilots act as streaming measurements. In this formulation, we perform online learning via recursive Bayesian filtering, exploiting model uncertainty to replace multiple gradient updates with a *single-shot* update. We leverage modular architectures to obtain Bayesian updates that are localized and parallel, enabling efficient and scalable online adaptation. The ability to update our model based on a single symbol in a single step allows us to view pilots as *streaming data*, namely, we do not aggregate data or wait for coherence durations, but immediately update the model on each incoming pilot to continuously learn.

3.1 State Space Formulation

We model online learning of deep receivers as a *dynamic system*. Consider a generic DNN-based receiver that maps its input $\boldsymbol{x}_t$ to bit-wise soft estimates of the true labels $\boldsymbol{b}_t \in \{0,1\}^B$. This formulation also applies to a single DeepSIC module (omitting the superscript k, q).

To capture channel variations in the parameter space, we define $\{\hat{\boldsymbol{\theta}}_t\}_{t=1,2,\ldots}$ as the random process describing the ground truth, or optimal, DNN parameters. Assuming smooth temporal variations, we adopt a Gaussian random walk model

$$p(\hat{\boldsymbol{\theta}}_t \mid \hat{\boldsymbol{\theta}}_{t-1} = \boldsymbol{\theta}_{t-1}) = \mathcal{N}(\gamma\boldsymbol{\theta}_{t-1}, \sigma^2\boldsymbol{I}), \tag{4}$$

where $\sigma^2 > 0$ controls the evolution rate and $\gamma \in (0,1]$ dictates the memory of the process. Values of the parameter γ near 1 favor stability in slow-varying channels, while smaller values allow faster adaptation. The initial parameters are assumed Gaussian, $p(\hat{\boldsymbol{\theta}}_0) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.

At time t, the output of the DNN with parameters $\hat{\boldsymbol{\theta}}_t$ is denoted $\boldsymbol{\ell}_t = h_{\hat{\boldsymbol{\theta}}_t}(\boldsymbol{x}_t)$, and defining a vector of independent Bernoulli variables, such that

$$p(b_{t,i} = 1 \mid \boldsymbol{x}_t, \hat{\boldsymbol{\theta}}_t) = \ell_{t,i}. \tag{5}$$

Equations (4)–(5) define a *nonlinear, non-Gaussian state space model (SSM)* with Gaussian state evolution. Online learning is thus cast as *tracking the time-evolving parameter vector* $\hat{\boldsymbol{\theta}}_t$ based on all observations up to time t, i.e., the pilots and DNN inputs $\{(\boldsymbol{x}_\tau, \boldsymbol{b}_\tau)\}_{\tau=1}^t$.

3.2 Bayesian Online Learning

We model online learning as the sequential estimation of desired DNN parameters $\hat{\boldsymbol{\theta}}_t$ evolving as a Markov process (4) and generating noisy observations (5) of the information bits. Each new data point (pilot) updates the belief $\boldsymbol{\theta}_t$ about $\hat{\boldsymbol{\theta}}_t$. We adopt *Bayesian deep learning*, considering BNNs with Gaussian variational distributions, i.e., $q_{\boldsymbol{\psi}_t}(\boldsymbol{\theta}_t) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. This allows efficient online updates using the current $(\boldsymbol{x}_t, \boldsymbol{b}_t)$, accounting for uncertainty in both past parameters and new data.

CM-EKF. Treating online learning as state tracking motivates Kalman-type updates. The conditional-moments EKF (CM-EKF) [36] recursively updates the posterior in a *predict–update* fashion. Predicting $\boldsymbol{\psi}_t$ from $\boldsymbol{\psi}_{t-1}$ using the Gaussian evolution model (4) yields

$$\boldsymbol{\mu}_{t|t-1} \triangleq \mathbb{E}[\hat{\boldsymbol{\theta}}_t \mid \hat{\boldsymbol{\theta}}_{t-1} = \boldsymbol{\theta}_{t-1}] = \gamma \boldsymbol{\mu}_{t-1}, \tag{6}$$

$$\boldsymbol{\Sigma}_{t|t-1} \triangleq \text{Cov}[\hat{\boldsymbol{\theta}}_t \mid \hat{\boldsymbol{\theta}}_{t-1} = \boldsymbol{\theta}_{t-1}] = \gamma^2 \boldsymbol{\Sigma}_{t-1} + \sigma^2 \boldsymbol{I}. \tag{7}$$

Updating incorporates a first-order linearization around $\boldsymbol{\mu}_{t|t-1}$:

$$\overline{h}_{\hat{\boldsymbol{\theta}}_t}(\boldsymbol{x}_t) \approx h_{\boldsymbol{\mu}_{t|t-1}}(\boldsymbol{x}_t) + \boldsymbol{H}_t(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\mu}_{t|t-1}), \tag{8}$$

where $\boldsymbol{H}_t$ is the network Jacobian. The posterior is then updated as

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{b}_t - h_{\boldsymbol{\mu}_{t|t-1}}(\boldsymbol{x}_t)), \tag{9}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{K}_t \boldsymbol{H}_t \boldsymbol{\Sigma}_{t|t-1}, \tag{10}$$

with Kalman gain $\boldsymbol{K}_t$ [36]. Notably, CM-EKF updates require a single forward and backward pass per observation, in contrast to multiple SGD iterations, with complexity $\mathcal{O}(BP^2 + B^3)$, where P is the number of trainable parameters. As P is often much larger than the number of bits per symbol B, complexity is dominated by $\mathcal{O}(BP^2)$ for typical settings.

Bayesian Online Natural Gradient. Alternatively, BNN parameters can be updated via a single natural gradient step of the expected log-likelihood, as in Bayesian online natural gradient (BONG) [28]:

$$\boldsymbol{\psi}_t = \boldsymbol{\psi}_{t|t-1} + \boldsymbol{F}^{-1}_{\boldsymbol{\psi}_{t|t-1}} \nabla_{\boldsymbol{\psi}_{t|t-1}} \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}_{t|t-1}}}[\log p(\boldsymbol{b}_t | \boldsymbol{x}_t, \boldsymbol{\theta})]. \tag{11}$$

For Gaussian variational distributions, this reduces to

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t|t-1} + \boldsymbol{\Sigma}_t \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}_{t|t-1}}}[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{b}_t | \boldsymbol{x}_t, \boldsymbol{\theta})], \tag{12}$$

$$\boldsymbol{\Sigma}_t^{-1} = \boldsymbol{\Sigma}^{-1}_{t|t-1} - \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}_{t|t-1}}}[\nabla^2_{\boldsymbol{\theta}} \log p(\boldsymbol{b}_t | \boldsymbol{x}_t, \boldsymbol{\theta})]. \tag{13}$$

The expectations in the BONG update can be approximated using an empirical Fisher (EF) approach, replacing the Hessian with the outer product of the gradient to reduce computational complexity [37]. Under the alternative linearized Gaussian likelihood approximation (8), the BONG update coincides with the CM-EKF updates, yielding a computationally stable one-step online learning rule [28].

3.3 Modular Bayesian Online Adaptation

Modeling the learning dynamics as an SSM with Gaussian BNNs enables reliable one-shot online learning. However, the associated complexity grows at least quadratically with the number of DNN weights P, limiting rapid adaptation for large networks. Here, we show how modular architectures such as DeepSIC reduce complexity and latency via modular updates, parallelization, and pipelining.

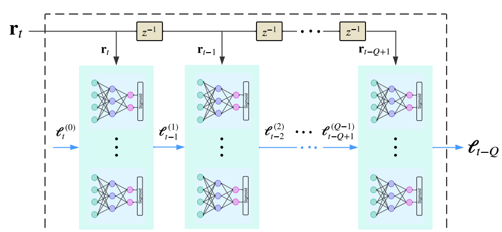

Figure 1: Illustration of Pipelined DeepSIC.

Module-Wise Updates: Each DeepSIC module maintains its Gaussian posterior $q^{(k,q)}_{\boldsymbol{\psi}_t} = \mathcal{N}(\boldsymbol{\mu}^{(k,q)}_{t-1}, \boldsymbol{\Sigma}^{(k,q)}_{t-1})$. Each module predicts the kth user bits $\boldsymbol{b}^{(k)}_t$ and thus can be treated as a separate module-wise SSM. Modules are updated independently via one-shot Bayesian learning. During inference, we use the mean as a plug-in approximation $\boldsymbol{\theta}^{(k,q)}_t = \boldsymbol{\mu}^{(k,q)}_t$ to avoid sampling overhead.

Parallelization: Updates of different modules can be computed concurrently. Since all K modules at iteration q share the input $\boldsymbol{x}^{(q)}_t$, one can leverage multicore or vectorized hardware to achieve up to a K-fold speed-up over sequential processing.

Pipelining: DeepSIC iterates over $q = 1, \ldots, Q$. Instead of processing each sample through all Q iterations before the next, one can pipeline Q samples such that iteration q processes sample $t - (q-1)$ (Fig. 1). This reduces overhead by up to a Q-fold, introducing minimal latency: at time t, the network outputs soft estimates of bits at $t - Q + 1$, which is acceptable in coded communications.

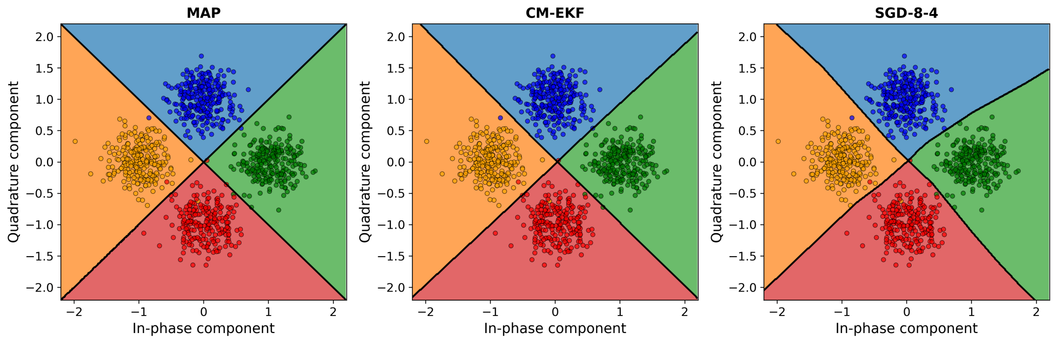

Figure 2: Decision zones after 500 blocks (45° rotation).

Algorithm Summary: Upon receiving pilot $(\boldsymbol{r}_t, \boldsymbol{b}_t)$, each module applies predict and update steps using $(\boldsymbol{x}_t^{(q)}, \boldsymbol{b}_t^{(k)})$, where $\boldsymbol{x}_t^{(q)}$ is defined in (3). Algorithm 1 summarizes the procedure, combining module-wise parallel filtering with pipelined execution. Each module-wise Predict-Update-Inference involves two forward passes and one backward pass. Using CM-EKF, the cost is $\mathcal{O}(BP^2)$ per module, where P is module size (much smaller than full DNN). End-to-end DeepSIC would scale as $\mathcal{O}(BQ^2K^2P^2)$, while modular updates scale as $\mathcal{O}(BQKP^2)$. With parallel computation, this can be reduced to $\mathcal{O}(BP^2)$, i.e., up to $\times QK$ overhead reduction.

Algorithm 1: Modular Online Bayesian Training at Time t

Require: Pilot queue $\{(\boldsymbol{r}_{t-i}, \boldsymbol{b}_{t-i})\}_{i=0}^{Q-1}$
Require: Variational parameters $\boldsymbol{\psi}_{t-q}^{(k,q)}$ for $k \in [1, K], q \in [1, Q]$

```
ℓ_t^(0) ← [0.5, ..., 0.5]^⊤, x_t^(0) ← [r_t^⊤, ℓ_t^(0)⊤]^⊤
for q = 1, ..., Q do
    τ ← t − (q − 1)
    for all k = 1, ..., K do
        Predict: ψ_{τ|τ−1}^(k,q) ← predict(ψ_{τ−1}^(k,q))
        Update: ψ_τ^(k,q) ← update(ψ_{τ|τ−1}^(k,q), x_τ^(q−1), b_τ^(k))
        Inference: ℓ_τ^(k,q) ← h_{μ_τ^(k,q)}(x_τ^(q−1))
    end for
    x_τ^(q) ← [r_τ^⊤, ℓ_τ^(q)⊤]^⊤
end for
```

Ensure: Updated variational parameters $\boldsymbol{\psi}_{t-q+1}^{(k,q)}$ for all k, q

4 Experimental Study

We evaluate the proposed online learning methodologies on three fronts: (i) a synthetic single-user linear channel to verify rapid adaptation of CM-EKF; (ii) modular vs. non-modular architectures on a realistic multi-user MIMO channel; (iii) streaming-data learning performance on linear and nonlinear multi-user MIMO channels. Our main DNN-aided receiver is based on DeepSIC [11], with each module a compact two-layer multi-layer perceptron (MLP).

Learning Algorithms. We compare the following online learning algorithms:

1. **SGD-I-J**: I mini-batch SGD epochs with batch size J on frequentist models.
2. **GD-I**: I Online gradient descent (GD) [38] iterations per sample on frequentist models.
3. **BBB-DIAG-I**: Online Bayes-by-backprop (BBB) [39] with diagonal Gaussian variational distribution $q_{\boldsymbol{\psi}_t} = \mathcal{N}(\mu_t, \text{diag}(\sigma_t))$ using I iterations per sample.
4. **BONG-EF**: BONG with linearized Hessian approximation (Sec. 3.2).
5. **CM-EKF**: Proposed extended Kalman-based Bayesian update (Sec. 3.2).

Only SGD processes pilots in batches; all others update per sample. Performance is reported under frequentist inference for fair comparison.

Linear Rotation Channel. We first consider a single-user QPSK rotation channel:

$$\boldsymbol{r}_{t,i} = \begin{bmatrix} \cos\varphi_t & -\sin\varphi_t \\ \sin\varphi_t & \cos\varphi_t \end{bmatrix} \boldsymbol{s}_{t,i} + \boldsymbol{u}_{t,i}, \quad \varphi_t = 2\pi\alpha t,\ \alpha = 2.5 \cdot 10^{-4}.$$

Here $\boldsymbol{u}_{t,i} \sim \mathcal{N}(0, \sigma_u^2 I)$, $\sigma_u^2 = 1/16$. Neural decoders have a single hidden layer of 10 ReLU units and sigmoid output. Fig. 2 visualizes the decision zones of the maximum *a-posteriori* (MAP) rule

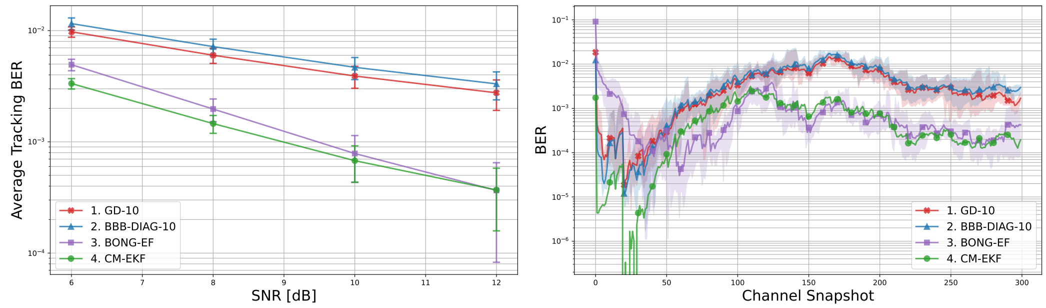

Figure 4: COST2100: Left BER vs SNR; Right BER vs channel snapshot at 10dB.

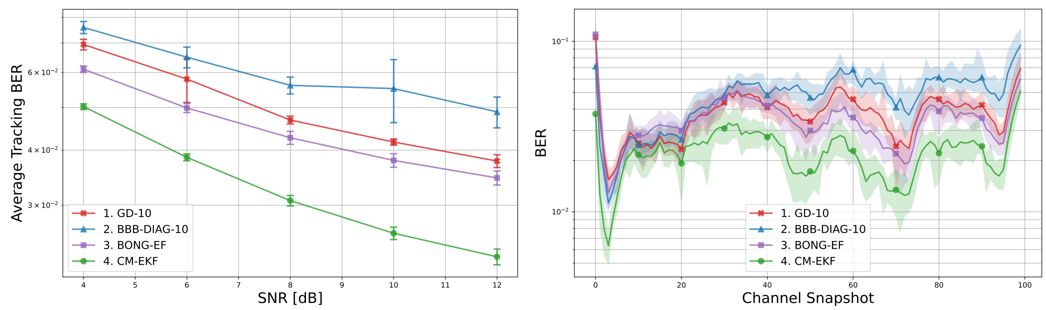

Figure 5: Nonlinear QuaDRiGa: Left BER vs SNR; Right BER vs channel snapshot at 12dB.

and the neural receivers after 500 blocks. CM-EKF closely matches the MAP decisions, while SGD distorts zones due to overfitting.

Modular vs Non-Modular Architectures. We compare modular DeepSIC ($P = 458$ per module) to a ResNet ($P = 32{,}766$). Data comes from QuaDRiGa 3GPP LoS scenario with $K = 3$ users, $N = 5$ antennas. Each snapshot has 64 symbols; first 4 snapshots for synchronization. Fig. 3 shows average BER vs. SNR. Modular architectures allow faster synchronization and better tracking. CM-EKF outperforms other methods, while BBB-DIAG and SGD are substantially less effective on ResNet.

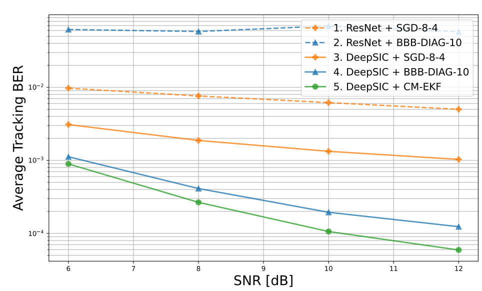

Figure 3: Modular vs non-modular learning.

Streaming-Data Algorithms. We evaluate BBB-DIAG-10, CM-EKF, BONG-EF, and GD-10 on linear COST2100 [29] and nonlinear QuaDRiGa channels. Linear: $K = 3$, $N = 5$, 300 snapshots, 2 pilots per snapshot after 2 synchronization snapshots. Nonlinear: $\tanh$ distortion on QuaDRiGa channels, 100 snapshots, 16 pilots per snapshot after 4 synchronization snapshots.

For COST2100 (Fig. 4), CM-EKF and BONG-EF outperform BBB-DIAG and GD. BONG-EF synchronizes more slowly due to the Hessian approximation instability. For nonlinear QuaDRiGa (Fig. 5), CM-EKF decisively outperforms all alternatives, achieving rapid synchronization and stable tracking; BONG-EF is slower, while BBB-DIAG struggles to maintain low error rates.

5 Conclusions

We proposed a modular online Bayesian learning framework for deep receivers operating over time-varying channels. By casting online learning as an SSM and adopting recursive Bayesian updates, we demonstrated that reliable one-shot adaptation can be achieved, replacing multi-epoch training with single-step updates. Furthermore, by leveraging modular receiver architectures, we showed how module-wise Bayesian adaptation enables parallelization and pipelining while treating pilots as streaming data, thereby substantially reducing both complexity and latency. Our methodology addresses the fundamental challenges of nonlinearity, non-stationarity, and delay constraints.

References

[1] C. Huang, S. Hu, G. C. Alexandropoulos, A. Zappone, C. Yuen, R. Zhang, M. Di Renzo, and M. Debbah, "Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends," *IEEE Commun. Mag.*, vol. 27, no. 5, pp. 118–125, 2020.

[2] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1546–1577, 2021.

[3] N. Shlezinger, G. C. Alexandropoulos, M. F. Imani, Y. C. Eldar, and D. R. Smith, "Dynamic metasurface antennas for 6G extreme massive MIMO communications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 106–113, 2021.

[4] N. Lang, Y. Gabay, N. Shlezinger, T. Routtenberg, Y. Ghasempour, G. C. Alexandropoulos, and Y. C. Eldar, "Wideband tHz multi-user downlink communications with leaky wave antennas," *IEEE Trans. Wireless Commun.*, 2025, early access.

[5] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[6] L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo, "Deep learning for wireless communications: An emerging interdisciplinary paradigm," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 133–139, 2020.

[7] W. Tong and G. Y. Li, "Nine challenges in artificial intelligence and wireless communications for 6G," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 140–145, 2022.

[8] J. Xia, D. Deng, and D. Fan, "A note on implementation methodologies of deep learning-based signal detection for conventional MIMO transmitters," *IEEE Trans. Broadcast.*, vol. 66, no. 3, pp. 744–745, 2020.

[9] 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 Commun.*, vol. 31, no. 4, pp. 163–169, 2024.

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

[11] N. Shlezinger, R. Fu, and Y. C. Eldar, "DeepSIC: Deep soft interference cancellation for multiuser MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1349–1362, 2021.

[12] N. Shlezinger and Y. C. Eldar, "Model-based deep learning," *Foundations and Trends® in Signal Processing*, vol. 17, no. 4, pp. 291–416, 2023.

[13] S. Park, H. Jang, O. Simeone, and J. Kang, "Learning to demodulate from few pilots via offline and online meta-learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 226 – 239, 2020.

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

[15] N. Uzlaner, T. Raviv, N. Shlezinger, and K. Todros, "Asynchronous online adaptation via modular drift detection for deep receivers," *IEEE Trans. Wireless Commun.*, vol. 24, no. 5, pp. 4454–4468, 2025.

[16] F. A. Aoudia and J. Hoydis, "End-to-end learning for OFDM: From neural receivers to pilotless communication," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1049–1063, 2021.

[17] M. B. Fischer, S. Dörner, S. Cammerer, T. Shimizu, H. Lu, and S. Ten Brink, "Adaptive neural network-based OFDM receivers," in *IEEE Signal Processing Advances in Wireless Communications (SPAWC)*, 2022.

[18] R. Finish, Y. Cohen, T. Raviv, and N. Shlezinger, "Symbol-level online channel tracking for deep receivers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8897–8901.

[19] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2019.

[20] T. Raviv and N. Shlezinger, "Data augmentation for deep receivers," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8259–8274, 2023.

[21] O. Simeone, *Machine Learning for Engineers*. Cambridge University Press, 2022.

[22] K. M. Cohen, S. Park, O. Simeone, and S. Shamai, "Bayesian active meta-learning for reliable and efficient AI-based demodulation," *IEEE Trans. Signal Process.*, vol. 70, pp. 5366–5380, 2022.

[23] M. Zecchin, S. Park, O. Simeone, M. Kountouris, and D. Gesbert, "Robust Bayesian learning for reliable wireless AI: Framework and applications," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 9, no. 4, pp. 897–912, 2023.

[24] T. Raviv, S. Park, O. Simeone, and N. Shlezinger, "Uncertainty-aware and reliable neural MIMO receivers via modular Bayesian deep learning," *IEEE Trans. Veh. Technol.*, 2025.

[25] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*. OUP Oxford, 2012, vol. 38.

[26] N. Shlezinger, Y. C. Eldar, and S. P. Boyd, "Model-based deep learning: On the intersection of deep learning and optimization," *IEEE Access*, vol. 10, pp. 115 384–115 398, 2022.

[27] S. Farquhar and Y. Gal, "A unifying Bayesian view of continual learning," *arXiv preprint arXiv:1902.06494*, 2019.

[28] M. Jones, P. Chang, and K. Murphy, "Bayesian online natural gradient (BONG)," *Advances in Neural Information Processing Systems*, vol. 37, pp. 131 104–131 153, 2024.

[29] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. De Doncker, "The COST 2100 MIMO channel model," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 92–99, 2012.

[30] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "Quadriga: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, 2014.

[31] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, 2003.

[32] K. T. Truong and R. W. Heath, "Effects of channel aging in massive mimo systems," *Journal of Communications and Networks*, vol. 15, no. 4, pp. 338–351, 2013.

[33] A. G. Wilson and P. Izmailov, "Bayesian deep learning and a probabilistic perspective of generalization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4697–4708, 2020.

[34] L. Chen, S. T. Jose, I. Nikoloska, S. Park, T. Chen, and O. Simeone, "Learning with limited samples: Meta-learning and applications to communication systems," *Foundations and Trends® in Signal Processing*, vol. 17, no. 2, pp. 79–208, 2023.

[35] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. Suppl 1, pp. 1513–1589, 2023.

[36] F. Tronarp, A. F. García-Fernández, and S. Särkkä, "Iterative filtering and smoothing in nonlinear and non-Gaussian systems using conditional moments," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 408–412, 2018.

[37] J. Martens, "New insights and perspectives on the natural gradient method," *J. Mach. Learn. Res.*, vol. 21, no. 1, Jan. 2020.

[38] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *International Conference on International Conference on Machine Learning*, 2003, p. 928–935.

[39] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*. PMLR, 2015, pp. 1613–1622.