Adaptive and Flexible Model-Based AI for Deep Receivers in Dynamic Channels

Tomer Raviv, Sangwoo Park, Osvaldo Simeone, Yonina C. Eldar, and Nir Shlezinger

Abstract

Artificial intelligence (AI) is envisioned to play a key role in future wireless technologies, with deep neural networks (DNNs) enabling digital receivers to learn how to operate in challenging communication scenarios. However, wireless receiver design poses unique challenges that fundamentally differ from those encountered in traditional deep learning domains. The main challenges arise from the limited power and computational resources of wireless devices as well as from the dynamic nature of wireless communications which causes continual changes to the data distribution. These challenges impair conventional AI based on highly-parameterized DNNs, motivating the development of adaptive, flexible, and light-weight AI for wireless communications, which is the focus of this article. We consider how AI-based design of wireless receivers requires rethinking of three main pillars of AI: architecture, data, and training algorithms. In terms of architecture, we review how to design compact DNNs via model-based deep learning. Then, we discuss how to acquire training data for deep receivers without compromising spectral efficiency. Finally, we review efficient, reliable, and robust training algorithms via meta-learning and generalized Bayesian learning. Numerical results are presented to demonstrate the complementary effectiveness of each of the surveyed methods. We conclude by presenting opportunities for future research on the development of practical deep receivers.

Introduction

Wireless communication technologies are subject to escalating demands for connectivity and throughput, with rapid growth in media transmissions, including images, videos, and, in the near future, augmented and virtual reality. Furthermore, transformative applications such as the Internet of Things (IOT), autonomous driving, and smart manufacturing are expected to play major roles in 5G-defined deployments of ultra-reliable and low-latency communication (URLLC) and massive machine-type communications (mMTC) services. To accommodate these scenarios, communication systems must meet strict performance requirements in reliability, latency, and complexity.

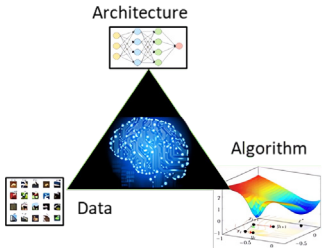
To facilitate meeting these performance requirements, emerging technologies such as mmWave and THz communication, holographic multiple-input multiple-output (MIMO), spectrum sharing, and intelligent reconfigurable surfaces (IRSs) are currently being investigated. While these technologies may support desired performance levels, they also introduce substantial design and operating complexity. For instance, holographic MIMO hardware is likely to introduce non-linearities on transmission and reception; the presence of IRSs complicates channel estimation; and classical communication models may no longer apply in novel settings such as the mmWave and THz spectrum, due to violations of far-field assumptions and lossy propagation. This article addresses the latter source of complexity by focusing on efficient design of receiver processing.

Traditional receiver processing design is model-based, relying on simplified channel models, which, as mentioned, may no longer be adequate to meet the requirements of next generation wireless systems. The rise of deep learning as an enabler technology for artificial intelligence (AI) has revolutionized various disciplines, ranging from computer vision and natural language processing (NLP) to speech refinement and biomedical signal processing. The ability of deep neural networks (DNNs) to learn mappings from data has spurred growing interest in their usage for receiver design in digital communications [1, 2]. DNN-aided receivers, referred to henceforth as *deep receivers*, have the ability to succeed where classical algorithms may fail. Specifically, deep receivers can learn a detection function in scenarios having no well established physics-based mathematical model, a situation known as *model-deficit*; or in settings for which the model is too complex to give rise to tractable and efficient model-based algorithms, a situation known as *algorithm-deficit*. Consequently, deep receivers have the potential to meet the constantly growing requirements of wireless systems.

Several core challenges arise from the fundamental differences between wireless communications and traditional AI domains such as computer vision and NLP, limiting the widespread applicability of deep learning in wireless communications. The first challenge is attributed to the nature of the *devices* employed in communication systems. Wireless communication receivers are highly constrained in terms of their computational ability, battery consumption, and memory resources. On

Tomer Raviv and Nir Shlezinger are with Ben-Gurion University of the Negev, Israel; Sangwoo Park and Osvaldo Simeone are with King's College London, U.K.; Yonina C. Eldar is with Weizmann Institute of Science, Israel.

Digital Object Identifier: 10.1109/MWC.012.2300242



Pillar	Method	Literature
Architecture	Deep unfolding	[3]–[5]
	DNN-aided inference	[6]
Data	Self-supervised training	[6], [7]
	Data augmentation	[8], [9]
Training Algorithm	Meta-learning	[10]–[12]
	Bayesian learning	[13]
	Modular training	[10]

FIGURE 1. A summary of methods surveyed in this article that adapt the three pillars of AI to the requirements of deep wireless.

Several core challenges arise from the fundamental differences between wireless communications and traditional AI domains such as computer vision and NLP, limiting the widespread applicability of deep learning in wireless communications.

the other hand, deep learning inherently relies on highly parameterized architectures, assuming the availability of powerful devices, for example, high-performance computing servers.

A second challenge stems from the nature of the wireless communication *domain*. Communication channels are dynamic, implying that the receiver task, dictated by the data distribution, changes over time. This makes the standard receivers pipeline of data collection, annotation, and training highly challenging. Specifically, DNNs rely on (typically labelled) data sets to learn from the underlying unknown, but stationary, data distributions. For example, machine translation tasks, requiring the mapping of an origin language into a destination language, do not change over time, enabling the collection of a large volume of training data and the deployment of a pretrained, static, DNN. This is not the case for wireless receivers, whose processing task depends on the time-varying channel, restricting the size of the training data set representing the task.

The two challenges outlined above imply that successfully applying AI for wireless receiver design requires deviating from conventional deep learning approaches. To this end, there is a need to develop communication-oriented AI techniques, which are the focus of this article. Previous tutorials on AI for communications, for example, [1, 2], have primarily concentrated on surveying challenges and applications of conventional deep learning methods in the context of communication systems. In contrast, the present article focuses on the design of practical and effective deep receivers that address the specific limitations imposed by the use of data- and resource-constrained wireless devices and by the dynamic nature of the communication channel.

We commence by motivating the development of AI systems that are *light-weight*, and thus operable on power and hardware limited devices, as well as *adaptive and flexible*, enabling online on-device adaptation. As illustrated in Fig. 1, we then propose that AI-based wireless receiver design requires revisiting the three main pillars of AI, namely: the *architecture* of AI models; the *data* used to train AI models; and the *training algorithm* that optimizes the AI model for generalization, that is, to maximize performance *outside* the training set (either on the same distribution or for a completely new one).

For each of these AI pillars, we survey candidate approaches for facilitating the operation of the deep receivers.

We first discuss how to design light-weight trainable architectures via *model-based deep learning* [14]. This methodology hinges on the principled incorporation of model-based processing, obtained from domain knowledge on optimized communication algorithms, within AI architectures.

Next, we investigate how labelled data can be obtained without impairing spectral efficiency, that is, without increasing the pilot overhead. To this end, we show how receivers can generate labelled data by *self-supervision* aided by existing communication algorithms; and how they may further enrich data sets via *data augmentation* techniques that utilize invariance properties of communication systems.

Finally, we cover training algorithms for deep receivers that are designed to meet requirements in terms of efficiency, reliability, and robust adaptation of wireless communication systems, avoiding overfitting from limited training data while limiting training time. These methods include communication-specific *meta-learning* as well as *generalized Bayesian learning* and *modular learning*.

To illustrate the individual and complementary gains of the reviewed approaches, we provide a numerical study considering finite-memory single-input single-output (SISO) channels as well as multi-user MIMO systems. We conclude by discussing the road ahead, as well as key research challenges that are yet to be addressed to enable adaptive and flexible light-weight deep receivers.

Deep Receivers in Dynamic Channels

Harnessing the potential of deep learning in wireless systems requires communication-specific AI schemes that are adaptive, flexible, and light-weight. The light-weight requirement follows from the power and computational constraints of wireless devices, while the need for adaptivity and flexibility is entailed by the dynamic nature of wireless channels. Classical model-based receiver processing is inherently adaptive and flexible: The receiver periodically estimates the channel using the available pilots, and then uses this estimate to adapt the operation of the receiver baseband chain, which is a direct function of the channel coefficients. In contrast, for deep receivers, the dependence of the weights of the DNN on the channel state is indirect, and hence designing flexible, channel adaptive, DNNs-based processing is a non-trivial task.

Current state of the art on deep receivers encompasses the following three main approaches to address channel variations.

A1 Joint Learning: The most straightforward approach amounts to optimizing a single DNN model to maximize performance on average over a broad range of channel conditions. Methods in this class train a DNN using data corresponding to an extensive set of expected channel realizations, aiming to learn a mapping that is tailored to the distribution of the channel. Accordingly, joint learning may be thought of as seeking the optimal

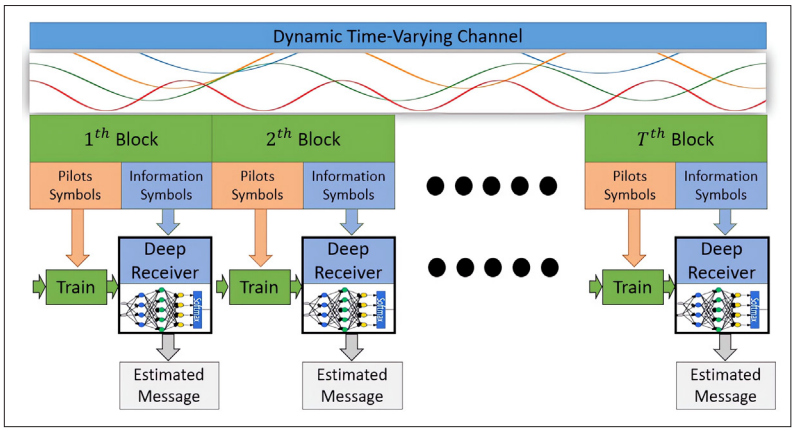


FIGURE 2. Overall illustration of online training of deep receivers in time-varying channels.

non-coherent receiver, which is agnostic to the current channel realization. As a result, performance degradation as compared to a *coherent receiver* is generally to be expected.

A2 Channel as Input: An alternative approach uses an instantaneous estimate of the channel as an additional input to the DNN [15]. Among the main drawbacks of this approach are the limited flexibility in accommodating different system dimensions, for example, number of antennas or number of users, and the lack of structure in the way different inputs, such as received signals and channel state information, are handled.

A3 Online Training: As illustrated in Fig. 2, in online training, decoded data from prior blocks is used, alongside new pilots, to adapt the deep receiver to channel variations. This class of approaches inherits the limitations of *continual learning*, such as catastrophic forgetting, and is generally not suitable to ensure fast adaptation.

The mentioned shortcomings of the three existing approaches reviewed above motivate a fundamental rethinking of the application of machine learning tools to wireless receivers along the three directions illustrated in Fig. 1:

- The architecture of the DNN should be carefully selected on the basis of domain knowledge to reduce data requirements, while also ensuring efficient implementation of the model. This amounts to improvements in terms of the *inductive bias* on which learning is based.
- The data used for learning should be augmented, when possible, by leveraging the inherent redundancies of encoded signals.
- The training algorithm should make use of historical data while also preparing for quick adaptation to changing channel conditions.

In the following sections we review candidate approaches for each of these aspects, as summarized in Fig. 1.

Architecture

Standard neural architectures employed in AI systems for communication are based on highly-parameterized, unstructured, deep neural models. However, these networks tend to be highly parameterized, and since deep receivers should adapt to time-varying conditions using limited training data, this type of architectures is typically undesirable. In this section, we introduce ways to design tailored model architectures by leveraging domain knowledge with the goal of improving adaptivity and data efficiency. Later we will also study data-driven approaches for the optimization of the inductive bias — also known as meta-learning — and see how they can be combined with model-driven architectures introduced in this section to further reduce the generalization gap.

In *model-based deep learning*, DNN architectures that are inspired by model-based algorithms are tailored to the particular problem of interest [14]. In the context of deep receivers, the dominant model-based deep learning methodologies are *deep unfolding* and *DNN-aided inference*, which are illustrated in Fig. 3 and discussed next.

Many model-based algorithms used by wireless receivers rely on iterative optimizers that operate by gradually improving an optimization variable based on an objective function. Deep unfolding converts an iterative optimizer into a discriminative AI model by introducing trainable parameters within each of a fixed number of iterations [14]. Training a deep unfolding architecture can thus adapt an iterative optimizer on the basis of available data for a given problem of interest. As we detail next, the aim is addressing model and/or algorithmic deficiencies of the original algorithm.

Specifically, deep unfolding enhances iterative optimizers in the following ways (see [14] for further details).

Learned Hyperparameters: Iterative optimizers often include hyperparameters, such as step-sizes, damping factors, and regularization coefficients, that are typically tuned by hand by the designer and shared among all iterations. Deep unfolding can treat such hyperparameters as trainable parameters. This is useful to cope with forms of *algorithm deficiency*, whereby an iterative algorithm requires too many iterations or struggles to converge to a suitable decision. For example, the work [3] showed that unfolding the orthogonal approximate message passing algorithm for MIMO detection, and learning iteration dependent scaling coefficients, notably improves performance, requiring only a few iterations.

Learned Objective: Deep unfolding can also enhance an iterative algorithm by tuning the objective functions approximately optimized at each iteration. This optimization addresses *algorithm deficiencies*, in a manner similar to the optimization of hyperparameters, as well as *model deficiencies* by adapting the design criterion to observed data, rather than to assumptions about the model. A representative example is the MMNet architecture proposed in [4] for unfolding MIMO detection. MMNet, which is based on proximal gradient steps, parameterizes the gradient computation procedure at each iteration, effectively using an iteration-dependent design objective.

DNN Conversion: An iterative optimizer can be converted into a trainable abstract architecture by incorporating a DNN module within each iteration in order to implement some functionality of the solver. When the iterative solver operates on a graph, as is the case for message passing algorithms, the solver can be unfolded into a graph neural network (GNN). GNNs support compact parameterizations by reusing DNN modules across different nodes and edges of the graph. DNN conversion is suitable for handling *model deficiency*, since the DNN modules learn how to best realize model-independent internal computations at

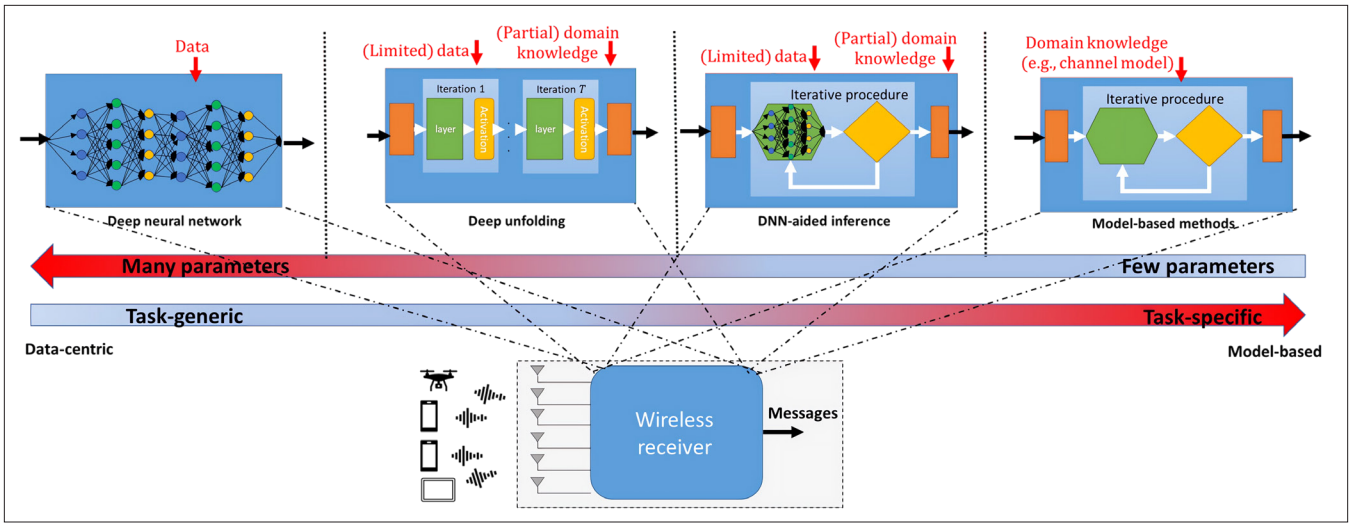


FIGURE 3. Illustration of model-based, data-driven, and model-based deep learning framework for deep receivers.

each iteration. For instance, DeepSIC proposed in [5] is derived from the iterative soft interference cancellation (SIC) MIMO detection algorithm with the introduction of DNN models for implementing each stage of interference cancellation and soft detection in a manner agnostic to the underlying channel model.

DNN-Aided inference refers to a family of model-based deep learning methods that incorporate DNNs into model-based methods that do not implement iterative processing. A representative example is the ViterbiNet equalizer proposed in [6]. Viterbi equalization is applicable to any finite-memory channel, as long as one can compute the conditional distribution of channel output given the corresponding input, also known as likelihood. Based on this observation, ViterbiNet implements the Viterbi algorithm while using a DNN to compute the likelihood. In this way, ViterbiNet addresses *model deficiencies* by operating in a channel-model-agnostic manner and requiring only the conventional finite-memory modelling assumption to hold.

Data

The amount of data obtained from pilots is typically insufficient to train an AI model for a deep receiver. This motivates the introduction of strategies that *expand* the available labelled training data set without requiring the transmission of more pilots. As we detail in this section, existing techniques apply either self-supervised learning or data augmentation.

With *self-supervised learning*, training data is extended using the redundancy of transmitted signals either at the symbol level or at the codeword level. In contrast, in *data augmentation*, the goal is to enrich the given labelled data set by leveraging invariance properties of the data. As summarized in Fig. 4, these approaches can be potentially combined, and integrated with a number of different architectures and training algorithms.

Codeword-level self-supervision exploits the presence of channel coding to generate labelled data from channel outputs. It uses error correction codes to correct detection errors, and then utilizes the corrected data as labelled data for training, as long as the codewords are decoded successfully [6, 7].

Symbol-level self-supervision obtains labelled data from information symbols without relying on channel decoding. This is useful since some symbols can be correctly detected even the decoding on the overall codeword fails. Symbol-level self-supervision hence requires reliable soft detection measures to indicate the degree to which each information symbol may be considered to be correctly received.

Data augmentation techniques enrich training sets by leveraging known invariances in the data. While data augmentation is common in AI, existing methods are highly geared toward image and language data, and do not address the nature of data and computing devices for wireless communications. For instance, for image classification, one can use a single image to generate multiple images with the same label by rotating or clipping the original image. Such augmentation techniques do not have obvious counterparts for wireless communications data, such as a sequence of channel outputs observed by a receiver. Furthermore, data augmentation in computer vision often relies on complex generative DNNs, whose implementation may be problematic for hardware-limited wireless devices.

Data augmentation for digital communications has been explored in [8], and more recently in [9]. The techniques studied in [9] leverage symmetry in digital constellations, independence between the noise and the transmitted symbols, and invariance to constellation-preserving rotations exhibited by wireless channels. These methods may also be applicable to other tasks such as DNN channel estimators, while other problems generally require the identification of distinct task-specific invariances.

Training

Training algorithms address the optimization of the parameters of the neural architecture based on the data, with the goal of identifying models with satisfactory generalization performance. The performance of a training algorithm depends, in practice, on the choice of the loss function; the optimization algorithm; and the relevance and quality of the data used to evaluate the training loss. In this section, we review communication-oriented approaches for designing adaptive data-efficient training algorithm for deep receivers based on

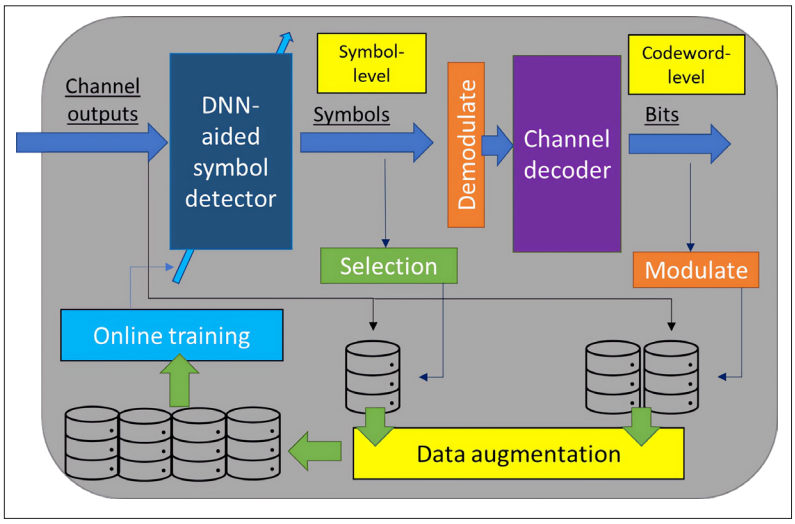


FIGURE 4. Data acquisition pipeline for deep receivers without impairing spectral efficiency.

meta-learning, generalized Bayesian learning, and modular learning. These methods are all associated with the training procedure. Meta-learning and modular training aim to achieve convergence with fewer training samples and iterations. The former is applicable to any architecture, and may leverage an additional buffer to store past data when implemented in an online fashion; while modular training incorporates architectural elements inspired by known solutions to related problems. Bayesian learning targets improvements in reliability, at the cost of storage and computing overheads due to the adoption of ensembling. The choice of any of these approaches, and combinations thereof, depends on the specific objectives and constraints of the communication system at hand.

Meta-learning is a general framework that seeks to obtain a *data-efficient* training procedure applicable for multiple tasks of interest [12]. A training procedure that is data-, or sample-, efficient is able to achieve a small generalization gap, while using a small amount of training data. Meta-learning and model-based learning are two complementary approaches that reduce the generalization gap under a fixed amount of training data: The former is data-driven and typically optimizes the training algorithm, while the latter is model-driven and optimizes the architecture. While meta-learning encompasses a variety of conceptually distinct methods, the prominent approaches for application to deep receivers are gradient-based meta-learning and hypernetwork-based meta-learning.

Gradient-Based Meta-Learning: Gradient-based meta-learning optimizes some of the hyperparameters of a first-order training algorithm. While in principle, one could "meta-learn" any hyperparameter, such as the learning rate, optimizing the initial weights of the DNNs has been found to be extremely beneficial for boosting adaptation and flexibility of training procedures in many applications, including wireless communications [12]. DNN initialization is a form of inductive bias, since the parametric function space of the DNN becomes restricted by enforcing adherence to the initialization through a limited number of gradient-based updates. Meta-learning can be combined with a model-based inductive bias, as demonstrated in [10].

Hypernetwork-Based Meta-Learning: Gradient-based meta-learning requires running a number of (stochastic) gradient updates. An alternative approach that does not require in real-time any additional optimization for adaptation to new tasks incorporates a hypernetwork in the system, alongside the main DNN. The hypernetwork takes as input the available data, or any other context information, regarding the task of interest, and produces at the output the weights of the main DNN. More precisely, typically, only a subset of weights of the main DNN are updated; and/or each output of the hypernetwork affects simultaneously a group of weights, for example, in the same layer, of the main DNN. Hypernetwork-based meta-learning has been applied successfully in wireless communication systems, including for beamforming and MIMO detection [11].

Bayesian learning is the gold standard for training strategies that aim at producing AI models offering a *reliable* assessment of the uncertainty of their decisions. Such reliable AI models must output confidence measures that reflect the true accuracy of their decisions. Bayesian learning boosts reliability by treating the model parameters as random variables, and by accordingly maintaining a *distribution* over the weights of a DNN. This distribution is meant to capture *epistemic uncertainty* in the presence of limited training data.

Bayesian learning involves particle-based, deterministic or stochastic, procedures, or optimization over the parameters of the distribution in the model parameter space. Such optimization addresses a training criterion that includes an information theoretic regularizer enforcing closeness to a prior distribution.

For deep receivers, boosting the reliability of a DNN model allows the latter to provide informative soft decision to downstream DNN or model-based modules, for example, for soft decoding. This makes it possible for the different modules of a deep receiver to "trust" the outputs of other modules.

Generalized forms of Bayesian learning allow for a flexible choice of the regularization function, as well as of the data fitting part of the training objective. Such methods were shown to be useful in wireless systems for their capacity to deal with model misspecification and outliers [13].

Modular learning exploits the interpretable structure of hybrid model-based deep receivers to facilitate rapid learning from limited data. As opposed to meta-learning and Bayesian learning, modular learning is specific to model-based deep learning architectures. It builds on the fact that, unlike blackbox DNNs, in model-based deep learning architectures, one can often assign a concrete functionality to different trainable sub-modules of the architecture, and not just to its input and output. Each functionality may then be adapted at different rates and times, as some functionalities may require rapid adaptation, while the others may be kept unchanged over a longer time scale.

This approach was applied in [10] for online adaptation of the DeepSIC MIMO receiver of [5]. There, the ability to associate different users with sub-modules of the deep receivers was leveraged to carry out the online training of sub-modules associated with users that are identified as being

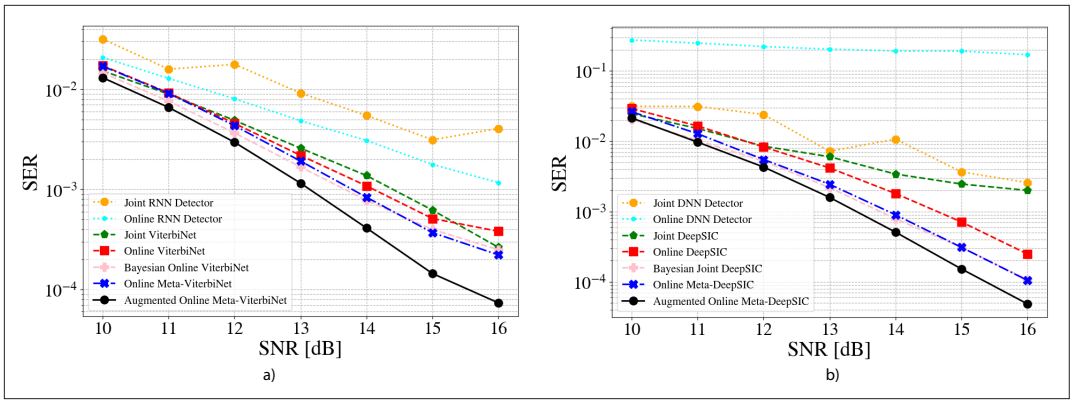


FIGURE 5. Average SER after transmission of 300 blocks in a time-varying channel as a function of SNR: a) SISO-SER as SNR; b) MIMO-SER vs. SNR.

characterized by faster dynamics. The method was shown to dramatically reduce the number of gradient-based updates and the amount of data needed for online training.

Numerical Results

In this section we showcase the impact of schemes designed to facilitate light-weight, adaptive, and flexible AI across the three AI pillars highlighted throughout this article. We focus on finite-memory SISO channels (with 4 taps) and memoryless 4 × 4 multi-user MIMO time-varying channels with binary phase shift keying (BPSK) and quadrature phase shift keying (QPSK) symbols (The source code used in our experiments is available at https://github.com/tomerraviv95/facilitating-adaptation-deep-receivers). The exact mathematical description of the channel models can be found in [9, Sec. V.C].

Architecture

In each channel, we consider a model-based DNN architecture, as well as black-box DNN, having roughly three times more parameters. For the SISO channel, with a finite channel memory of L symbols, we compare ViterbiNet [6] with a recurrent neural network (RNN)-based symbol detector with a window size of L, followed by a linear layer and the softmax function. For the MIMO channel, the DeepSIC receiver [5] with three iterations is compared to a fully connected DNN composed of four layers with ReLU activations followed by the softmax layer.

Data

For each coherence duration, 200 pilot symbols are available. We compare standard training with training that leverages data augmentation. For the latter scheme, at each time step, the pilot data is enriched with 600 artificial symbols via a constellation-conserving projection and a translation preserving transformation [9].

Training

We consider the following training methodologies.

Joint Training: The receiver is trained offline, using 5000 symbols simulated from a multitude of channel realizations. No additional training is done at run time.

Online Training: The receiver is trained initially using 200 symbols, and then it adapts online by utilizing either the pilot data or the augmented pilot data.

Online Meta-Learning: The training algorithm is optimized via meta-learning that utilizes accumulated training data from previous channel realizations, while adaptation takes place via few gradient-based updates from the online meta-learned initialization [10].

Bayesian Learning: The training process produces a probability distribution over the parameters of the architecture, which is used for ensembling (with five randomly generated models) during both training and testing [13].

Results

Figures 5a and b depict the average symbol error rate (SER) as a function of signal-to-noise ratio (SNR). While standard black-box models suffer from large generalization gaps due to the limited availability of training data, deep receivers with model-based architectures, namely ViterbiNet [6] and DeepSIC [5], demonstrate successful detection performance by adapting to the time-varying channel in an online manner.

The performance is further improved by optimizing the *training algorithm* via meta-learning, or by implementing ensembling via Bayesian learning, as well as by increasing the *data* size via data augmentation. Overall, these results indicate that the reviewed methods are complementary, contributing to the challenges of adapting to time-varying channels in different ways. This leads to the conclusion that designing AI models for communications can benefit from a rethinking of deep learning tools across all three AI pillars.

In the MIMO setup (Fig. 5b), online learning with a conventional DNN fails to provide satisfactory results, whereas joint learning partially succeeds. This result stems from fact that the constellation size encompasses $4^4 = 256$ distinct symbols, causing online learning with only 200 to underperform due to insufficient number of samples per class. This setup demonstrates that even online re-training may prove ineffective with extremely limited labeled data using a black-box architecture. However, the presented approaches manage to handle this extreme low-data regime by exploiting the inductive bias of model-based deep learning, or by enriching the data with augmentations.

Future Research Directions

We conclude by identifying some representative directions for future research.

AI algorithms are typically computationally complex and power hungry. This follows from the large number of parameters, as well as from the lengthy training procedure, involving multiple iterations and frequent data access.

Deciding When to Train

The schemes surveyed enable efficient online training. A key open question is how to determine when to train online. Periodically re-training, for example, at each coherence period, may be excessively complex, particularly when channel variations are relatively smooth. Deep receivers can benefit from monitoring mechanisms that determine when to adapt the model and/or meta-learn the inductive bias. This can be achieved using data drift detection, a topic widely studied in the machine learning literature. While some drift detectors can be applied to communication systems, advanced mechanisms that leverage communication-specific characteristics require further development.

Fitting the Architecture to the Scenario

Deep receivers are often composed of multiple layers, wherein each element takes part in the computation. Thus, even for relatively light-weight architectures, full model computations may incur computational overhead exceeding the limited resources available, particularly for some edge devices. This is typically tackled via pruning methods, which remove redundant model parts to balance complexity and performance. While most existing pruning methods find a single, input-independent, model, deep receivers may preferinput-dependent, adaptive pruning methods, adapting complexity to the current requirements.

Hardware-Aware and Power-Aware AI

AI algorithms are typically computationally complex and power hungry. This follows from the large number of parameters, as well as from the lengthy training procedure, involving multiple iterations and frequent data access. The schemes surveyed in this article address these challenges from an algorithmic perspective by limiting the architecture parameterization and/or the number of training iterations. These algorithmic advances should be complemented by advances in computing hardware platforms, such as in-memory computing, and by hardware-software co-design methods.

Continual Bayesian Learning

Bayesian learning was introduced for deep receivers thanks to the potential gains that are enabled by the deployment of more reliable AI modules. Another advantage of Bayesian learning is its capacity to support continual learning by updating the parameter distribution. Integrating online adaptation with Bayesian learning may further enhance the performance of deep receivers.

Interpretable and Explainable AI for Deep Receivers

The deployment of AI modules in communication systems would be significantly facilitated by the implementation of mechanisms that ensure trust and transparency. Trust can be established by equipping AI models with the capacity to validate their outputs, making it possible to diagnose issues and to identify performance bottlenecks. Transparency may be supported via interpretable AI modules that leverage model-based algorithms having processing steps rigorously derived from optimality criteria [14].

Acknowledgment

This project has received funding from the Israeli 5G-WIN consortium, the European Union's Horizon 2020 research and innovation program under grants no. 646804-ERC-COG-BNYQ, as well as 725731, and by the European Union's Horizon Europe project CENTRIC (101096379). Support is also acknowledged for the Israel Science Foundation under grant no. 0100101, and for an Open Fellowship of the EPSRC with reference EP/W024101/1.

References

[1] L. Dai *et al.*, "Deep Learning for Wireless Communications: An Emerging Interdisciplinary Paradigm," *IEEE Wireless Commun.*, vol. 27, no. 4, 2020, pp. 133–39.
[2] W. Tong and G. Y. Li, "Nine Challenges in Artificial Intelligence and Wireless Communications for 6G," *IEEE Wireless Commun.*, vol. 29, no. 4, 2022, pp. 140–45.
[3] H. He *et al.*, "Model-Driven Deep Learning for MIMO Detection," *IEEE Trans. Signal Process.*, vol. 68, 2020, pp. 1702–15.
[4] M. Khani *et al.*, "Adaptive Neural Signal Detection for Massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, 2020, pp. 5635–48.
[5] 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, 2021, pp. 1349–62.
[6] N. Shlezinger *et al.*, "ViterbiNet: A Deep Learning Based Viterbi Algorithm for Symbol Detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, 2020, pp. 3319–31.
[7] M. B. Fischer *et al.*, "Adaptive Neural Network-Based OFDM Receivers," *Proc. IEEE SPAWC*, 2022.
[8] L. Huang *et al.*, "Data Augmentation for Deep Learning-Based Radio Modulation Classification," *IEEE Access*, vol. 8, 2019, pp. 1498–1506.
[9] T. Raviv and N. Shlezinger, "Data Augmentation for Deep Receivers," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, 2023, pp. 8259–74.
[10] T. Raviv *et al.*, "Online Meta-Learning for Hybrid Model-Based Deep Receivers," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, 2023, pp. 6415–31.
[11] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep Hypernetwork-Based MIMO Detection," Proc. IEEE SPAWC, 2020.
[12] L. Chen *et al.*, "Learning with Limited Samples: Meta-Learning and Applications to Communication Systems," *Foundations and Trends in Signal Processing*, vol. 17, no. 2, 2023, pp. 79–208.
[13] M. Zecchin *et al.*, "Robust Bayesian Learning for Reliable Wireless AI: Framework and Applications," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 4, 2023, pp. 897–912.
[14] 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, 2022, pp. 115,384–98.
[15] M. Honkala, D. Korpi, and J. M. Huttunen, "DeepRx: Fully Convolutional Deep Learning Receiver," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, 2021, pp. 3925–40.

Biographies

Tomer Raviv (tomerraviv95@gmail.com) is currently pursuing his Ph.D degree in electrical engineering in Ben-Gurion University.

Sangwoo Park (sangwoo.park@kcl.ac.uk) is currently a research associate at the Department of Engineering, King's Communications, Learning and Information Processing (KCLIP) lab, King's College London, United Kingdom.

Osvaldo Simeone (osvaldo.simeone@kcl.ac.uk) is a Professor of Information Engineering with the Centre for Telecommunications Research at the Department of Engineering of King's College London, where he directs the King's Communications, Learning and Information Processing lab.

Yonina C. Eldar [F] (yonina.eldar@weizmann.ac.il) is a Professor in the Department of Math and Computer Science, Weizmann Institute of Science, Israel, where she heads the center for Biomedical Engineering and Signal Processing. She is a member of the Israel Academy of Sciences and Humanities, and a EURASIP Fellow.

Nir Shlezinger (nirshl@bgu.ac.il) is an Assistant Professor in the School of Electrical and Computer Engineering in Ben-Gurion University, Israel.