Deep Neural Network-Based Detection and Partial Response Equalization for Multilayer Magnetic Recording

Ahmed Aboutaleb[1], Amirhossein Sayyafan[1], Krishnamoorthy Sivakumar[1], Benjamin Belzer[1], Simon Greaves[2], Kheong Sann Chan[3], and Roger Wood[1], *Life Fellow, IEEE*

[1]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164 USA
[2]Research Institute of Electrical Communication (RIEC), Tohoku University, Sendai 980-8577, Japan
[3]Nanjing Institute of Technology, Nanjing 211167, China

To increase the storage capacity limit of magnetic recording channels, recent studies proposed multilayer magnetic recording (MLMR): the vertical stacking of magnetic media layers. MLMR readback waveforms consist of the superposition of signals from each layer recovered by a read head placed above the upper layer. This article considers the problem of equalization and detection for MLMR comprising two layers. To this end, we use MLMR waveforms generated using a grain switching probability (GSP) model that is trained on realistic micromagnetic simulations. We propose three systems for equalization and detection. The first is a convolutional neural network (CNN) equalizer followed by an MLMR Viterbi algorithm (VA) for detection. We show that this system outperforms the traditional 2-D linear minimum mean squared error (2-D-LMMSE) equalizer. The second system uses CNNs for equalization and separation of signals from each layer, which is followed by a regular VA. The third system contains CNNs trained to directly provide soft bit estimates. By interfacing the CNN detector with a channel decoder, we show that an areal density gain of 16.2% can be achieved by a two-layer MLMR system over a one-layer system.

Index Terms— **Convolutional neural network (NN) (CNN), detection, dual-layer recording, multilayer magnetic recording (MLMR), partial response equalization, two-dimensional magnetic recording (TDMR), Viterbi algorithm (VA).**

I. Introduction

THE hard disk drive (HDD) industry stores data at areal densities close to the capacity limit of the one-dimensional magnetic recording (1-DMR) channel [1]. New technologies are emerging to increase density, including heat-assisted magnetic recording (HAMR), microwave-assisted magnetic recording (MAMR), and two-dimensional magnetic recording (TDMR). TDMR employs 2-D signal processing to achieve significant density gains, without changes to the existing magnetic media. Simple versions of TDMR are already being shipped in HDD products. However, because HAMR and MAMR require substantial changes to the read/write head and the magnetic media, their deployment will occur after TDMR.

Recent encouraging studies [2]–[6] propose multilayer magnetic recording (MLMR): the vertical stacking of an additional magnetic media layer to a TDMR system to achieve further density gains. Using a realistic grain switching probability (GSP) model to generate waveforms in a two-layer MLMR system as in [2]–[4], this article investigates methods based on deep neural networks (NNs) (DNNs) for equalization and detection for MLMR. This article addresses only the processing of readback data and not the write process.

The first work to design detection methods for MLMR is [5]. Therein, Chan *et al*. model readback in a multi-layer system, where each layer is a 1-D recording structure. Such a system can potentially be realized using perpendicular magnetic recording (PMR), including the use of MAMR to write selectively on each layer. The model and simulations use a convolutional model to generate waveforms. The readback waveforms are measured only above the upper layer but consist of a superposition of signal contributions from both the upper and lower layers. The challenge is to detect correctly the bits on each layer from such superposition waveforms.

Building upon the work in [5], Chan *et al.* [6] extend the MLMR system to also include inter-track interference (ITI) in the received waveforms. In realistic waveforms, ITI is caused by the tight packing of tracks using shingled magnetic recording (SMR) to increase areal density [1]. An appropriate Viterbi algorithm (VA) is designed for maximum likelihood (ML) detection, and another least-squares (LS)-based detection method is included for comparison. Working in parallel, Greaves *et al.* [2], [3] generate MLMR waveforms by using a micromagnetic model to train a GSP model. In addition to incorporating the superposition of signals from the upper and lower layer (including the ITI) in the readback waveforms, this GSP model uses realistic media noise models and incorporates transition noise during the write process that was not considered in previous models.

High-density magnetic recording channels suffer from inter-symbol interference (ISI) that can span many bits in the down-track direction. The number of trellis states required by a VA is exponential in the length of the ISI. Thus, to avoid the trellis state explosion problem, in a typical detection system, an equalizer precedes a VA to reduce the effective length of the ISI. The equalization is traditionally implemented using the linear minimum mean squared error (LMMSE)

Manuscript received August 3, 2020; revised October 19, 2020; accepted November 5, 2020. Date of publication November 17, 2020; date of current version February 18, 2021. Corresponding author: K. Sivakumar (e-mail: siva@wsu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMAG.2020.3038435.

Digital Object Identifier 10.1109/TMAG.2020.3038435

equalizer actualized as finite impulse response (FIR) filter for its simplicity. The LMMSE equalizer is MSE optimal for additive white Gaussian noise (AWGN) channels with ISI (see [7]). Matcha and Srinivasa [8] study 2-D LMMSE equalization and pattern-dependent noise prediction (2-D-PDNP) for TDMR systems. In MLMR, the presence of data-dependent media noise is further exacerbated by inter-layer interference (ILI). Hence, preliminary experiments on MLMR waveforms revealed poor bit error rate (BER) performance achieved by the conventional 2-D-LMMSE equalizer followed by the VA system.

To achieve high recording densities, NN-based media noise prediction, equalization, and detection are proposed in [9]–[13]. Sayyafan *et al.* [9] designed hybrid Bahl–Cocke–Jelinek–Raviv (BCJR) and DNN systems for iterative detection over 1-D channels. Among these DNN-based systems, the convolutional NN (CNN) system outperforms a conventional method comprised of a BCJR followed by a PDNP in terms of detector BERs. Furthermore, the BCJR-CNN system avoids the state explosion problem due to the PDNP and is not limited by the linear auto-regressive model assumed by the PDNP.

Nair and Moon [10] (and references therein) investigate equalization using NNs for high-density 1-D channels. For such channels, their study shows that non-linear NN equalizers outperform the LMMSE equalizer when the readback waveforms include non-linear distortions, such as bit transition shifts, partial erasures, and jitter noise. For TDMR channels, Luo *et al.* [13] show that an NN equalizer-VA system outperforms a 2-D-LMMSE-VA system in terms of BER. Shen and Nangare [11] propose using an NN with one hidden layer and non-linear tangent hyperbolic activation functions to equalize TDMR readback waveforms. Their results show that, under both mean squared error (MSE) and cross-entropy (CE) adaptation, the non-linear NN equalizer outperforms the 2-D linear equalizer while requiring a modest increase in complexity. Shen *et al.* [12] design DNN-based detection systems for high-density TDMR systems. In comparison with a conventional system consisting of 2-D-LMMSE followed by a 2-D-BCJR and a 2-D-PDNP, a 2-D-LMMSE equalizer followed by a CNN detector achieves a higher information areal density gain. Thus, NNs are crucial for achieving high-density gains in magnetic recording channels.

The novel contributions of this article are summarized as follows.

1) We propose the design of and training method for a hybrid CNN equalizer followed by an MLMR VA. We test this system using realistic readback waveforms generated via a GSP model. We show the proposed system outperforms the conventional 2-D-LMMSE equalizer-VA systems in terms of both MSE and detector BER.
2) We propose a CNN equalizer/separator system that equalizes and separates readback waveforms from each layer. The separation of signals allows using regular VAs for each layer.
3) We design a CNN detector system for MLMR. We show that this system provides a good performance–complexity tradeoff. We also discuss good practice techniques for tuning the CNN performance during the design stage.
4) We investigate the density gain achieved by our CNN equalizer/separator-VA and CNN detector systems on realistic GSP model waveforms for MLMR. By interfacing the detection systems with an irregular repeat-accumulate (IRA) decoder, we provide a realistic estimate of the areal density gain of 16.2% for a two-layer magnetic recording system over a one-layer magnetic recording system.

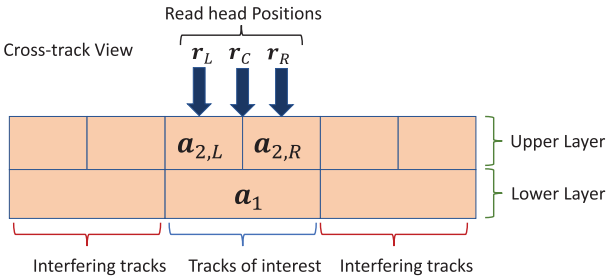


Fig. 1. Cross-track view of an MLMR system. The bit sequences on the top left, top right, and bottom tracks are denoted by $\mathbf{a}_{2,L}$, $\mathbf{a}_{2,R}$, and $\mathbf{a}_1$, respectively. Reading sequences taken at read head positions left, center, and right are denoted by $\mathbf{r}_L$, $\mathbf{r}_C$, and $\mathbf{r}_R$, respectively.

II. System Model

A. GSP Model

Fig. 1 shows a cross-track view of the MLMR system considered. On the upper layer, six tracks are written at a track pitch (TP) of 48 nm and a bit length (BL) of 11 nm. On the lower layer, three tracks are written at a TP of 96 nm and a BL of 22 nm. Hence, the system stores one bit on the lower layer for every four bits on the upper layer. The magnetic media thicknesses on the upper and lower layers are 5 and 7 nm, respectively. A non-magnetic 2 nm thick layer separates the upper and lower layers. Consistent with the notation in [5], [6], the lower layer is indexed by 1 and the upper layer by 2. For the tracks of interest, we denote the bit sequences written on the upper left and right tracks by $\mathbf{a}_{2,L}$ and $\mathbf{a}_{2,R}$, respectively, and the bit sequence on the lower track by $\mathbf{a}_1$. Readings are obtained every 11 nm down-track at the left, center, and right cross-track positions. Measured reading sequences are denoted by $\mathbf{r}_L$, $\mathbf{r}_C$, and $\mathbf{r}_R$, respectively. The readback waveforms are generated according to the GSP model in [2], [3]. We use the GSP model to test the methods discussed in all our simulation results. The GSP model used in this article is detailed in [2, Sec. III]. To motivate the use of CNNs for MLMR, we summarize important aspects of write and readback processes assumed by the GSP model.

The write process introduces the data-dependent noise in the readback waveforms, which is explained as follows. The input coded bits determine the polarity of the write head field applied to grains in the recording medium. In this work, the TP and BL are 48 nm $\times$ 11 nm and 96 nm $\times$ 22 nm for the top and bottom layers, respectively. The grains in the recording media were modeled as Voronoi cells with an average grain size of 8 nm, an average grain pitch of 9 nm, and a grain

size distribution of 17%. The distribution of grain sizes and shapes gives rise to transition jitter in the readback waveforms. The GSP model gives the probability that the magnetization of grain in the recording medium will switch in the direction of the head field when a bit is written. The switching probability of grain depends on the position of the grain relative to the write head, i.e. the strength of the write head field acting on the grain, and also on the magnetization of nearby grains via magnetostatic interactions. For example, the magnetization of grain i is affected by the magnetization of grains contained within the neighborhood of grain i, denoted by $\mathcal{N}_i$. In this article, $\mathcal{N}_i$ includes grains in the same layer within a 30 nm radius of grain i and also the grain immediately above/below grain i in the other layer.

If the nearby grains in the same layer are magnetized in the same direction as grain i, the magnetostatic field from the nearby grains will increase the probability that the magnetization of grain i will switch. Conversely, for two vertically stacked grains in different layers, the magnetostatic field will reduce the probability of switching if the grain magnetizations are parallel and increase it when the magnetizations are antiparallel. By taking account of the magnetization of grains in $\mathcal{N}_i$, the switching probability of grain i becomes dependent on the data sequence written to the medium; as a result, in some cases, the magnetization of grains within a written bit will not switch into the desired direction, or the magnetization of a grain in a previously written bit might be inadvertently switched, resulting in partial erasure.

During the read process, the read head is placed above the upper layer. The readback signal for cross-track position m observed for the down-track bit cell n is denoted by $r_m[n]$. The discrete-time reading $r_m[n]$ includes the effect of down-track ISI and cross-track ITI, as well as the superposition of signals from different layers. Let a'_k denote the magnetization pattern for layer k, and $h_k[i, j]$ the 2-D ISI/ITI reader response to magnetization pattern on layer k, calculated using the method in [2] and [3]. Then, $r_m[n]$ is the superposition of signal contributions from each layer. The signal component of each layer is the convolution of the reader sensitivity response with the magnetization pattern. More precisely, for the two-layer system, $r_m[n]$ is computed as

$$r_m[n] = \sum_{k=1}^{2} \sum_{i,j} h_k[i, j] a'_k[m - i, n - j] + n_m[n] \qquad (1)$$

where $n_m[n]$ is AWGN with zero-mean and variance σ_e^2 to model the reader head's electronics' noise.

The GSP model is trained on realistic micromagnetic simulations comprising tens of thousands of written bits. As detailed in [2], the waveforms generated by the GSP model match very closely the micromagnetic waveforms. The GSP model provides a computationally efficient, yet accurate, method of generating readback waveforms.

The conventional 2-D-LMMSE equalizer is optimal in terms of MSE only in the absence of the pattern-dependent media noise, partial erasures, and transition jitter [10]. Accurately characterizing the bit-grain interactions and transition jitter is difficult. Hence, an optimal model-based detector is challenging to design. In contrast, NNs learn to adapt their learnable parameters based on the raw readings without any assumption about the channel model except that the inputs and true outputs tuples are sampled from a common joint distribution [15]. Hence, NNs are better equipped to handle the mentioned distortions. Furthermore, an NN with one hidden layer and sigmoid activation is a universal function approximator, i.e., given enough hidden nodes, it can accurately approximate any continuous mapping from the inputs to the true outputs [16]. Recent studies [17] (and references therein) have established that CNNs with rectified-linear unit (ReLU) activation with enough hidden layers are also universal function approximators.

The caveat to using CNNs (and NNs) is that the labeled training data are needed to adapt the learnable parameters of the network. Also, the training data should be randomly generated (or sampled) so that the joint probability distribution of the inputs and true outputs is well-represented (see [15, Ch. 6]). Fortunately, labeled data generated with random input bit sequences can be readily generated for magnetic recording applications, e.g., using the GSP model or measurements from HDDs. Another aspect of NNs that must be considered is their computational complexity. High complexity CNNs can provide significant performance improvements over model-based methods. Hence, the desired complexity of the CNN must be balanced with expected performance improvement based on the available computational resources.

B. 2-D-LMMSE Equalizer-1024-State MLMR VA

For use over the GSP model, where ISI spans about 17 bits down-track, the 2-D-LMMSE equalizer minimizes the MSE between ideal PR signals and the actual output of the equalizer. The ideal partial response (PR) signals can be based on the ideal convolutional model in [6] and are given by

$$\mathbf{y}_L = w_1 \mathbf{g}_1 * \mathbf{a}_1 + \mathbf{g}_2 * \mathbf{a}_{2,L} \qquad (2)$$

$$\mathbf{y}_C = \mathbf{g}_1 * \mathbf{a}_1 + w_2 \mathbf{g}_2 * \mathbf{a}_{2,L} + w_2 \mathbf{g}_2 * \mathbf{a}_{2,R} \qquad (3)$$

$$\mathbf{y}_R = w_1 \mathbf{g}_1 * \mathbf{a}_1 + \mathbf{g}_2 * \mathbf{a}_{2,R} \qquad (4)$$

where w_1 and w_2 are weights tuned during training, and $\mathbf{g}_1$ and $\mathbf{g}_2$ are 1-D targets of appropriate length that are adapted during training. For interfacing with the 1024-state VA in [6], the lengths of $\mathbf{g}_1$ and $\mathbf{g}_2$ are seven and three, respectively.

The 2-D-LMMSE equalizer consists of three 2-D filters $\mathbf{F}_i = [\mathbf{f}_{i,L}, \mathbf{f}_{i,C}, \mathbf{f}_{i,R}]$, $i = 1, 2, 3$—one filter per PR target bit stream in (2)–(4). For example, for obtaining $\mathbf{F}_1$, we solve for

$$\mathbf{F}_1 = \underset{\mathbf{f}_{1,L}, \mathbf{f}_{1,C}, \mathbf{f}_{1,R}}{\arg\min} \; \mathbb{E}\left[\mathbf{y}_L - \left(\mathbf{f}_{1,L} * \mathbf{r}_L + \mathbf{f}_{1,C} * \mathbf{r}_C + \mathbf{f}_{1,R} * \mathbf{r}_R\right)\right]^2 \qquad (5)$$

where $\mathbb{E}$ denotes the expectation operator. The derivation for solving for $\mathbf{F}_1$ is shown in the Appendix. Similar training is used to obtain $\mathbf{F}_2$ and $\mathbf{F}_3$ except that $\mathbf{y}_C$ and $\mathbf{y}_R$ are used, respectively, in place of $\mathbf{y}_L$. Once optimal $\mathbf{F}_i$'s are found, the equalized signal is passed to a constrained MSE solver that optimizes w_1, w_2, $\mathbf{g}_1$, and $\mathbf{g}_2$ for fixed equalizers under energy or monic constraints on the target masks. The trellis state definitions and branch metric computation for such a

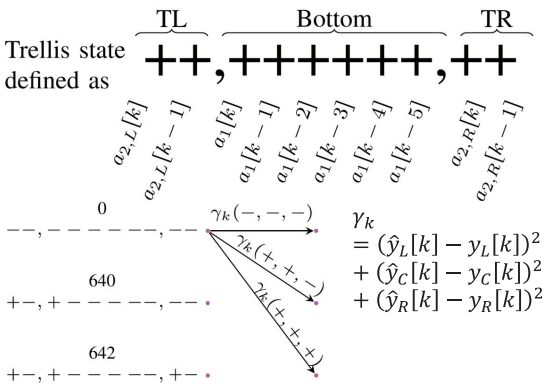


Fig. 2. State definitions and branch metric computation for the 1024-state VA designed for an ideal MLMR system. The branch metric is computed as the sums of the squared Euclidean distance between the equalizer output (denoted as $\hat{y}[k]$) and the ideal PR target signal (denoted as $y[k]$).

model are shown in Fig. 2. This article uses this 1024-state VA in conjunction with linear and non-linear equalizers on GSP model waveforms.

III. Proposed Equalization and Detection Systems

We propose three methods for detection and equalization of bit sequences $\mathbf{a}_1$, $\mathbf{a}_{2,L}$, and $\mathbf{a}_{2,R}$ from readings $\mathbf{r}_L$, $\mathbf{r}_C$, and $\mathbf{r}_R$.

A. Nonlinear CNN Equalization-VA System

1) Motivation for Non-Linear Equalization for MLMR: Consider the equalization of a signal that consists of the superposition of two responses $\mathbf{r} = \mathbf{h}_1 * \mathbf{a}_1 + \mathbf{h}_2 * \mathbf{a}_2$. To reduce the number of states to a manageable amount before interfacing with a VA, an equalizer is used. We would like the equalizer output to approximate the PR target $\mathbf{y} = \mathbf{g}_1 * \mathbf{a}_1 + \mathbf{g}_2 * \mathbf{a}_2$, where the lengths of $\mathbf{g}_1$ and $\mathbf{g}_2$ are less than the lengths of $\mathbf{h}_1$ and $\mathbf{h}_2$, respectively. Applying a linear equalizer $\mathbf{f}$ to $\mathbf{r}$ gives

$$\hat{\mathbf{y}} = \mathbf{f} * \mathbf{r} \tag{6}$$

$$= \mathbf{f} * \mathbf{h}_1 * \mathbf{a}_1 + \mathbf{f} * \mathbf{h}_2 * \mathbf{a}_2 \tag{7}$$

where the second equality follows because of the principle of superposition and since $\mathbf{f}$ is linear. From (7), we observe that the same $\mathbf{f}$ is tasked with mapping $\mathbf{h}_1$ to $\mathbf{g}_1$ and $\mathbf{h}_2$ to $\mathbf{g}_2$.

Whereas, if we apply a non-linear equalizer $\mathcal{F}\{\cdot\}$ to $\mathbf{r}$, the equalized signal is given by

$$\hat{\mathbf{y}} = \mathcal{F}\{\mathbf{r}\}. \tag{8}$$

The non-linear equalizer is not restricted by the superposition principle. Hence, the equalized signal can generally be comprised of the sum of different mappings applied on $\mathbf{r}$ to approximate the individual components of the ideal PR signal, that is

$$\hat{\mathbf{y}} = \mathcal{F}_1\{\mathbf{r}\} + \mathcal{F}_2\{\mathbf{r}\} \tag{9}$$

where $\mathcal{F}_1\{\cdot\}$ and $\mathcal{F}_2\{\cdot\}$ are generally different mappings that focus on equalizing $\mathbf{r}$ to $\mathbf{g}_1 * \mathbf{a}_1$ and $\mathbf{g}_2 * \mathbf{a}_2$, respectively. For the CNN equalizer, it can be intuitive to consider that the mappings $\mathcal{F}_1\{\cdot\}$ and $\mathcal{F}_2\{\cdot\}$ are subset weights and non-linear activation functions of $\mathcal{F}\{\cdot\}$. Fig. 3 illustrates applying linear and non-linear equalizers to $\mathbf{r}$.

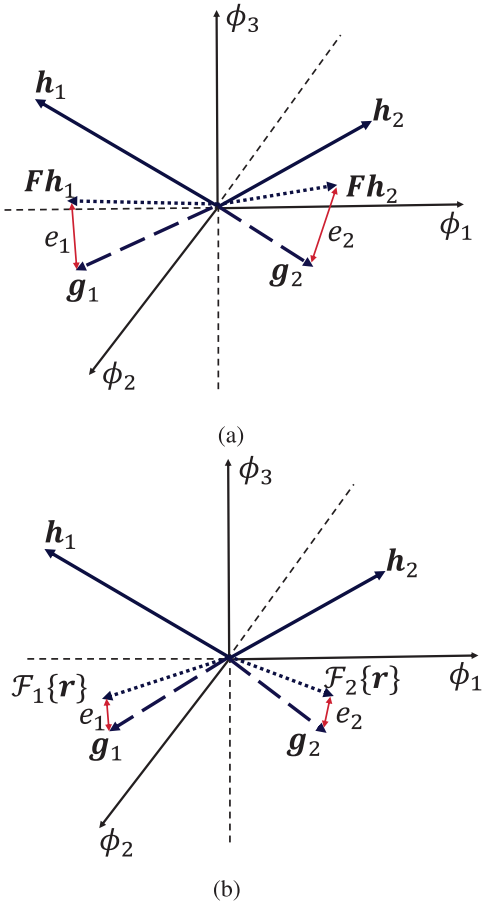


Fig. 3. Actual and target responses are represented with respect to some arbitrary basis functions ϕ_i. LMMSE equalization maps the response from each layer to target PR signals using the same mapping due to the principle of superposition, whereas nonlinear CNN equalization potentially uses different mappings for mapping $\mathbf{r}$ to each component of the target signal, resulting in smaller magnitudes for the error vectors e_1 and e_2. (a) Linear equalization. (b) Nonlinear CNN equalization.

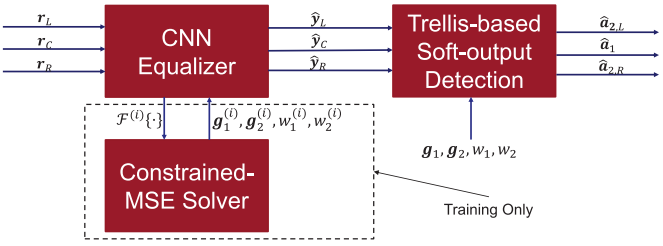


Fig. 4. CNN equalizer-VA system.

2) Training the Non-Linear Equalizer CNN: Fig. 4 shows the nonlinear CNN equalizer followed by a VA system. During training, the CNN equalizer iterates with a constrained MSE solver to adjust the target PR masks. We use the VA in [6], which is the ML detector for an ideal MLMR channel that does not consider the signal-dependence of the noise caused by the write process and the magnetic medium.

The CNN equalizer training assumes fixed target masks and superposition weights. Using stochastic gradient descent on mini-batches of length N (see [18]), the equalizer CNN

minimizes the average MSE J_{MSE} between its output and the ideal PR waveforms y_L, y_C, and y_R, defined as

$$J_{\text{MSE}} = \frac{1}{3N} \sum_{j\in\{L,C,R\}} \sum_{k=0}^{N-1} \big(\hat{y}_j[k] - y_j[k]\big)^2. \tag{10}$$

Once the CNN equalizer has converged and the MSE does not decrease significantly, the CNN equalizer's output (now fixed) is passed to a constrained MSE solver that finds new targets and superposition weights that further decrease the MSE subject to certain constraints. The constraints imposed provide upper and lower bounds on the energy of the masks and include a monic constraint $\mathbf{u}_i^T \mathbf{g}_i = 1$, $i = 1, 2$, where the monic vector $\mathbf{u}_i$ consists of zeros except at the central element that is set to one, i.e., $\mathbf{u}_i = [0, \ldots, 0, 0, 1, 0, 0, \ldots, 0]$. We find that including the monic constraint improves considerably the detector BER. Also, upper bounds on the magnitudes of the superposition weights are incorporated. More precisely, the constrained MSE problem can be written as

$$\underset{\mathbf{g}_1,\mathbf{g}_2,w_1,w_2}{\text{minimize}} \; J_{\text{MSE}} \tag{11a}$$

$$\text{subject to } c_{i,\min} \le \|\mathbf{g}_i\|_2^2 \le c_{i,\max}, \tag{11b}$$

$$\mathbf{u}_i^T \mathbf{g}_i = 1, \quad |w_i| \le p_i, \; i = 1, 2 \tag{11c}$$

where $c_{i,\min}$ and $c_{i,\max}$ are the lower and upper bounds on the masks' energies, and p_i's are upper bounds on the magnitudes of the superposition weights. The energy and magnitude constraints prevent the optimizer from possibly reaching a trivial solution (via lower bound) or diverging to infinite-energy masks (via upper bound). The specific values for the bounds can be tuned during a design stage.

After the constrained MSE solver finds a solution, the new targets and weights are fixed and used to generate new ideal PR signals using (2)–(4) for continuing the training of the CNN equalizer. Then, once the CNN equalizer has converged and achieved a new minimum MSE, its (fixed) output is fed to the constrained MSE solver to get new targets and superposition weights. The process repeats until the learnable parameters of the CNN, the targets, and superposition weights have converged, yielding no more improvements in MSE. Note that this iterative training reduces the search space and avoids possible divergence due to having both the CNN and targets adapting simultaneously.

Fig. 5 shows the architecture of the CNN equalizer used. The inputs to the CNN equalizer are observations of readings $\mathbf{r}_L$, $\mathbf{r}_C$, and $\mathbf{r}_R$ over a sliding window of size 3×17. The type of convolutional operation performed is "same," which means that the input is padded with zeros at the beginning of the input sequence such that the size of the output is the same as the size of the input. Hence, the output of each convolutional layer is $3 \times 17 \times D$, where D is the number of groups of convolutional filters; D is 27, 9, and 3 for the first, second, and third convolutional layers. The number of filters per group at the current layer is equal to the number of groups of convolutional filters in its preceding layer. The first layer consists of 27 groups of convolutional filters. Each convolutional filter is a 2-D FIR filter of size 3×11. The second layer contains nine groups of convolutional filters. Each

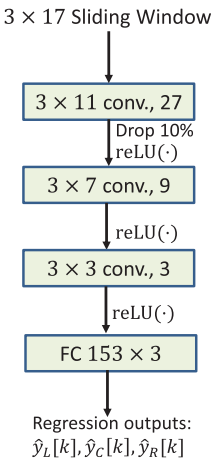


Fig. 5. CNN equalizer architecture.

group consists of 27 convolutional filters since the previous layer outputs samples of size 3×17 per filter group. The size of each filter is 3×7. The third layer contains three groups of convolutional filters. Each group consists of nine convolutional filters since the previous layer contains nine convolutional filter groups. The filters on the third layer are size 3×3.

To reduce the chance of over-fitting, a dropout layer is applied at the output of the first convolutional layer [19]. During training, dropout randomly zeroes the outputs of the convolutional layer according to a defined probability of dropout p, preventing the CNN from over-relying on certain hidden outputs for estimation. This procedure has the effect of sampling from a collection of possible thinned CNNs and training these thinned CNNs. During testing, no outputs are zeroed out. However, the weights of the layer preceding dropout are multiplied by $1 - p$. This is approximately equivalent to averaging the performance of all thinned CNNs. The first convolutional layer uses the largest number of filters. Hence, it contains a relatively large number of learnable parameters that can cause over-fitting of the learned parameters to the training data [15, Sec. 5.2]. Thus, we use a dropout layer after the first convolutional layer. The dropout probability is tuned during the design stage to avoid over- or under-fitting. Note that adding too many dropout layers or including dropout layers with large p may lead to under-fitting which results in the CNN achieving suboptimal performance.

Batch normalization is used at the output of the convolutional layers to reduce CNN's sensitivity to random weights initialization and provide some regularization benefits [20]. It computes the element-wise means and variances for each output dimension over a mini-batch and then normalizes each output so that each output dimension is zero-mean and has a unit variance. At the end of the training, the batch normalization means and variances are computed over the entire training data set and stored for normalization during the testing stage.

Following batch normalization, a non-linear activation function is applied. We use the ReLU activation function, defined by the element-wise relation $f(x) = \max(0, x)$. The ReLU activation function lowers implementation complexity

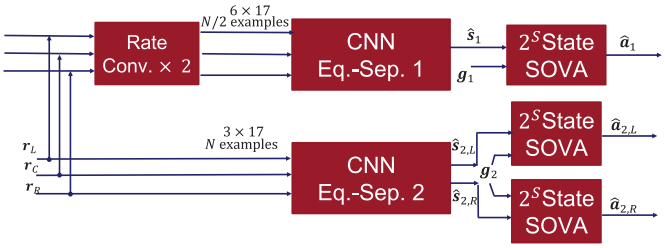


Fig. 6. CNN equalizer/separator-VA system.

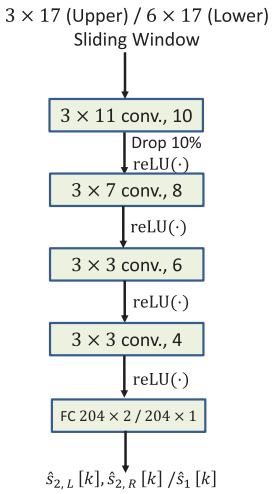


Fig. 7. CNN architecture used for the equalizer/separator-VA system.

compared with the conventional sigmoid activation and ameliorates the issue of vanishing gradients encountered when using sigmoid and other activation functions [21]. The output of the last convolutional layer is flattened to a column vector of size 153×1 before being fed to the last layer, which is a fully connected (FC) layer comprised of a matrix of size 153×3 and three bias variables. The outputs of the FC layer are the estimates $\hat{y}_L[k]$, $\hat{y}_C[k]$, and $\hat{y}_R[k]$.

B. Nonlinear CNN Equalization/Separation-VA System

To simplify the optimization of the parameters in (11), we also propose a CNN equalizer/separator system that delegates the problem of signal separation to the equalizer instead of the detector. In this system, the CNN is trained to both equalize and separate the signals from each layer. Thus, the target optimization stage can focus on optimizing only the individual target masks. Fig. 6 shows a block diagram of the CNN equalizer/separator-VA system. The equalized and separated signal are passed to regular 1-D-2^S-state VA detectors. The ideal PR signals for this system are given by

$$\mathbf{s}_1 = \mathbf{g}_1 * \mathbf{a}_1 \tag{12}$$

$$\mathbf{s}_{2,L} = \mathbf{g}_2 * \mathbf{a}_{2,L} \tag{13}$$

$$\mathbf{s}_{2,R} = \mathbf{g}_2 * \mathbf{a}_{2,R} \tag{14}$$

where our implementation uses $(S+1)$-tap targets $\mathbf{g}_1$ and $\mathbf{g}_2$. Note that, in the previous CNN equalizer-1024-state VA system, the length of the bit sequence on the lower layer is repetition coded at rate-$1/2$ so that the PR target is compatible with the expected input signal into the MLMR 1024-state VA, whereas, for this system, such repetition coding is not needed because a separate VA is used for each layer.

1) Training: Two CNNs are trained for equalization on each layer. Assuming fixed target masks, the objective functions to minimize during the CNN training are given by

$$J_{\text{MSE},1} = \frac{2}{N} \sum_{k=0}^{N/2-1} (\hat{s}_1[k] - s_1[k])^2 \tag{15}$$

$$J_{\text{MSE},2} = \frac{1}{2N} \sum_{j\in\{L,R\}} \sum_{k=0}^{N-1} \left(\hat{s}_{2,j}[k] - s_{2,j}[k]\right)^2 \tag{16}$$

for the lower and upper layers, respectively. Once the training of the CNN equalizer has converged, the CNN outputs are fixed and used to adapt the targets to further minimize the MSE. In this case, the target optimization problem simplifies to (for $i = 1, 2$)

$$\underset{\mathbf{g}_i}{\text{minimize}}\ J_{\text{MSE},i} \tag{17a}$$

$$\text{subject to } \mathbf{u}_i^T \mathbf{g}_i = 1, c_{i,\min} \le \|\mathbf{g}_i\|_2^2 \le c_{i,\max} \tag{17b}$$

where the bounds on $\|\mathbf{g}_i\|_2^2$ restrict the search space to feasible solutions. Following the targets optimization, the CNN is retrained to further minimize the MSE assuming the new targets. This iteration process repeats until no further substantial reductions in MSE are observed. Compared with target optimization in the previous system in (11), the target optimization problem in (17) is simpler to tune.

2) CNN Architecture: Fig. 7 shows the CNN architecture used for the CNN equalizer/separator-VA system. The input to the CNN equalizer/separator for the upper layer is a 3×17 sliding window. Its outputs are the estimates of the PR signals in (13) and (14) for the left and right tracks of interest. For the lower layer, an input sliding window of length 6×17 is used since two samples per read head are taken for each lower layer bit. The output is an estimate of the ideal PR signal for the lower layer in (12).

C. CNN Detection System

1) Training the CNN Detectors: CNNs can be trained to detect bits from raw readings. For instance, the readback samples contained within a sliding window can be used to detect bits on the upper or lower layers. In this view, the samples contained within each interval of the sliding window constitute an image whose correct classification label is the true bit at the center of the window. Thus, bit detection can be represented as an image classification problem (see [12]). CNNs have been successful at accurately classifying images when appropriately trained [15]. Thus, CNN provides a promising method for bit detection over realistic digital storage channels when the training data are available, but the channel model is difficult to characterize.

Fig. 8 illustrates the CNN-only system that consists of CNNs for detection on each data stream. Here, the CNN performs the actual data detection rather than providing nonlinear equalization for a subsequent conventional detector. Readings

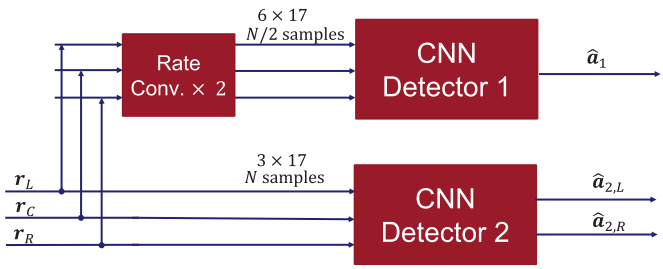


Fig. 8. CNN detector system.

within a 3×17 sliding window comprise input samples to the CNN for detecting the upper layer bits. Since each reader collects two samples per lower layer bit, and to maintain a 17-bit down-track footprint, a rate converter multiplexes these additional readings across-track, resulting in size 6×17 lower layer input samples. Each CNN detector accepts input samples within the sliding window and estimates its corresponding bit label. The detector CNN minimizes the CE loss function between the correct bit labels and its soft-output estimate. Consider estimating the kth bit $a_L[k]$, and let $\hat{a}_L[k]$ represent its estimate. Let the indicator function $\mathbb{1}_{a_L[k]=i} = 1$ if $a_L[k] = i$, $i = 0, 1$, and zero otherwise. Then, the CE loss is defined as

$$\mathcal{H}\{a_L[k], \hat{a}_L[k]\} = -\mathbb{1}_{a_L[k]=0} \log(\Pr\{\hat{a}_L[k] = 0\}) - \mathbb{1}_{a_L[k]=1} \log(1 - \Pr\{\hat{a}_L[k] = 0\}). \quad (18)$$

During training, stochastic gradient descent minimizes the average CE loss $J_{\text{CE}} = (1/N) \sum_{k=0}^{N-1} \mathcal{H}\{a_L[k], \hat{a}_L[k]\}$ computed over a length-N mini-batch (see [15, Section 8.1.3]). Let C_i, $i = 0, 1$, denote the ith output of the final $M \times 2$ FC layer, where M is the length of the vectorized output of the penultimate layer. Then, the soft bit estimate is obtained using the softmax activation function [15, eq. (6.29)]

$$\Pr\{\hat{a}_L[k] = 0\} = \frac{e^{C_0}}{e^{C_0} + e^{C_1}}. \quad (19)$$

2) CNN Detection Architectures: Fig. 9 shows CNN detector architectures. The CNN detector supersedes the equalizer (whether linear or non-linear) and the conventional trellis-based detector. Hence, its complexity is expected to be higher than the CNN equalizer. The increase in complexity can be due to increasing the width (the number of filters at each layer or the sizes of the filters) or depth (the number of hidden layers) of the CNN. Deeper CNNs can achieve higher classification accuracy in image recognition tasks compared with wider CNNs with the same complexity [22]. Increasing the depth of CNN usually leads to an increase in the training time required to observe accuracy improvements over shallower CNNs. We mitigate this increase in training time by using residual paths and replacing the ReLU activation with a leaky ReLU activation. Residual learning is proposed in [23] to exploit the improved learning capability enjoyed by deep CNN while easing the training and optimization stage. In regular CNNs, the input to the current layer is the output from the previous layer. In residual learning, the input to the current layer is the sum of the outputs from the previous layer and another preceding layer. Including such residual paths provides more direct paths from the input to the output. Hence, CNN can use relevant information from the input more directly to perform the bit detection task, instead of waiting for the previous layer to learn an appropriate representation of the input. Thus, residual learning can improve the image classification accuracy [23]. Hence, we leverage residual learning for designing the detection CNNs. Given the same training time, we find that, for deep CNNs, including the residual learning paths achieves slightly higher accuracies than CNNs with the same number of parameters but without residual paths.

Furthermore, ReLU activation may not be optimal in terms of training time. For instance, a hidden output activated by a ReLU can get stuck at zero for most of the input samples during training [24]. Thus, the gradient corresponding to the weights associated with such hidden output can also be zero for most of the training duration. Since training time is limited, this phenomenon may also result in the suboptimal final performance, which was observed for predecessor architectures of CNN Detector D1 in Fig. 9(a). To alleviate this issue, Mass *et al.* [24] proposed the leaky ReLU activation that is defined by the element-wise relation

$$\text{Leaky ReLU}(a, x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x \leq 0 \end{cases} \quad (20)$$

where a is tuned during the design stage and is typically set between zero and one. As long as $a > 0$, the hidden outputs are generally non-zero during training. Hence, the problem of having zero gradients is avoided. However, if $a = 1$, the CNN loses the improved capability for generalization due to using non-linear activation. Thus, a must be carefully tuned during the design stage.

IV. Simulation Results

We generate 100 blocks of waveforms based on the GSP model in [2] and [3]. Each block contains 82412 bits per track on the upper layer and 41206 bits per track on the lower layer. We use 60 blocks for training, 20 blocks for validation, and 20 blocks for testing. Stochastic gradient descent computes gradients and updates weights using mini-batches that are subsets of the training blocks. To prevent over- or under-fitting at the design stage, the performances of the CNN-based systems on the validation data set are used to tune the hyperparameters of the CNN and the adjustable properties training algorithm. The CNN hyperparameters are the properties of the CNN that are determined by the designer before training—including the CNN topology, the input size, the number of layers, the number and sizes of convolutional filters in each layer, the activation functions used, and the value of the dropout probability. Also, the validation results can be used to tune properties of the learning algorithm, such as the model regularization parameter and the learning rate, stop the training after a certain number of epochs have passed, or adjust the learning rate schedule [15, Ch. 7]. Finally, we assess the methods on the testing data set. To reflect realistic performance during implementation, no hyperparameters are tuned based on testing results.

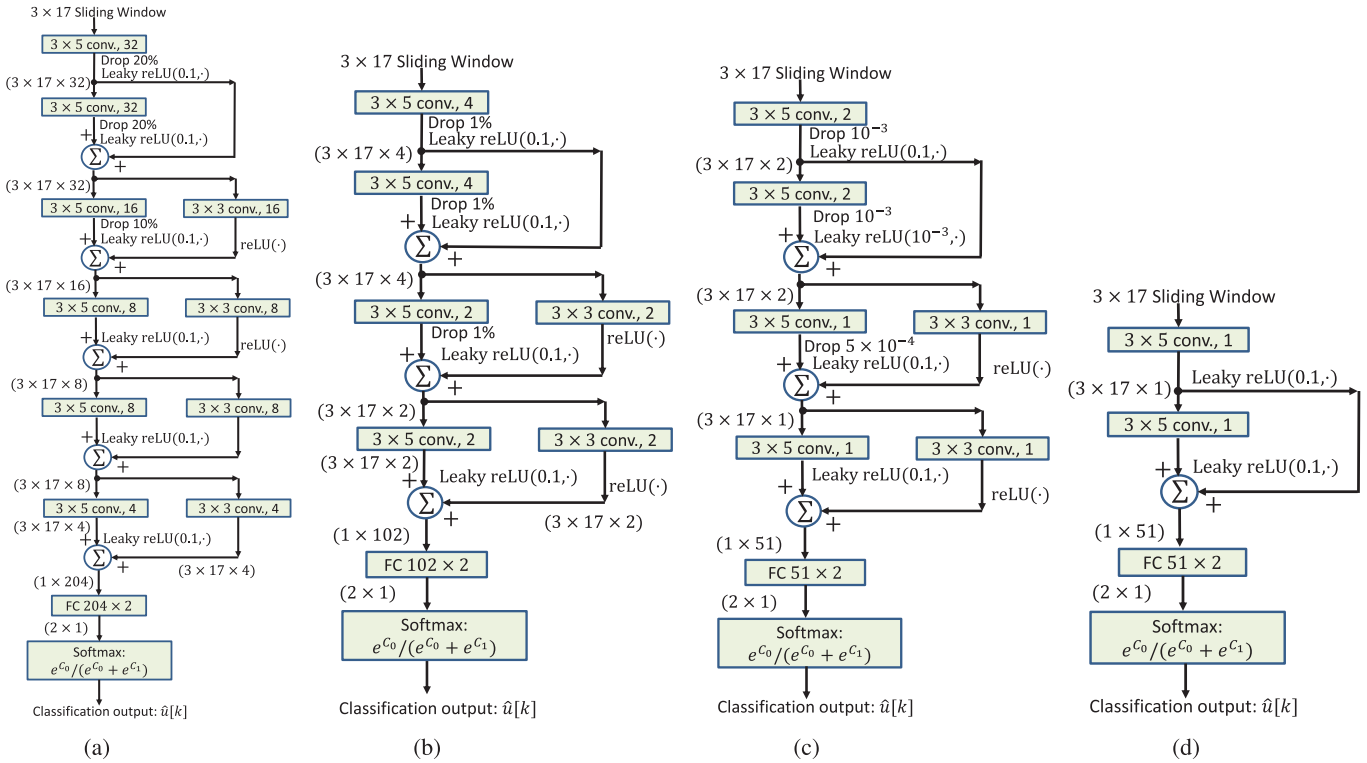


Fig. 9. CNN detector architectures incorporating residual paths. To illustrate their performance–complexity tradeoff, we show the detector BER on the upper layer achieved by each architecture in parenthesis. (a) CNN Detector D1 (6.610%). (b) CNN Detector D2 (6.758%). (c) CNN Detector D3 (6.867%). (d) CNN Detector D4 (7.304%).

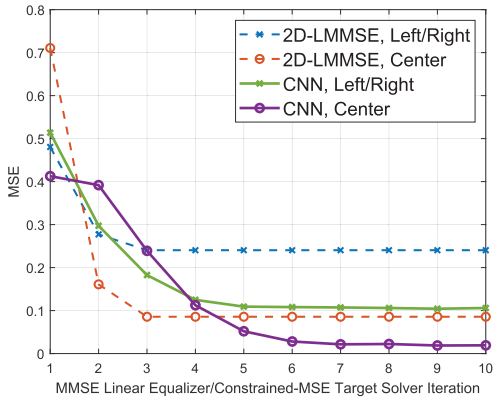
We use the Adam optimizer, a variant of stochastic gradient descent that uses the first- and second-order moments of the gradients to adapt the learning rates for individual learnable parameters [18]. Compared with standard stochastic gradient descent, Adam requires little additional memory and computations but provides significant performance gains. Consistent with previous empirical results on high-density magnetic recording data in [12], we also observed that Adam gives better error performance than the standard stochastic gradient descent algorithm and its variant root-mean-squared propagation (RMSProp) algorithm [25].

For training the CNN equalizer, the mini-batch length is set to 500 samples, and the learning rate is set to 10^{-3}. The CNN equalizer/separator system was first trained using a small mini-batch length of 100 samples during the first four iterations with constrained MSE solver to obtain most of the gains quickly. In the following iterations, the mini-batch length was increased to 500 samples to achieve further gains. For training the CNN detector D1, training was started using a mini-batch length of 100 samples, which was then gradually increased to 1000 samples as training iterations progressed. Increasing the batch size in this manner speeds convergence by obtaining most of the performance at the beginning of training before extracting final performance gains with more computationally expensive gradient steps. Such a scheme has been studied by Smith *et al.* [26] and shown to also reduce the chances of over-fitting. The learning rate was initialized to 10^{-3} and decreased to 10^{-5} in the final few iterations to ensure that possible small final performance gains are achieved. For training CNN detectors D2, D3, and D4, the mini-batch length is fixed to 1000 samples, and an adaptive learning rate is used.

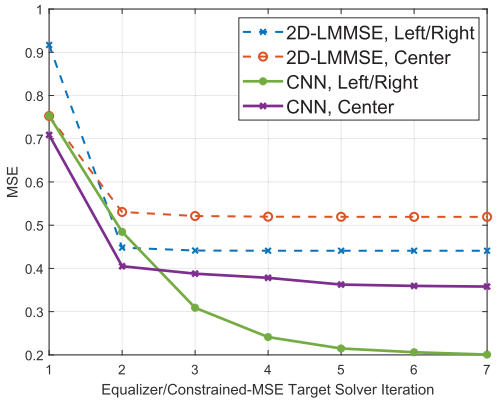
The 2-D-LMMSE and CNN equalizers require a delay of eight samples before they can provide outputs. Also, for optimal performance, the 1024-state VA requires starting from the state with all inputs set to -1. To remedy these issues and jump-start the equalizer-VA systems, eight samples are generated based on the ideal PR signal and using inputs of -1s.

A. Equalizer MSEs

The constraints imposed on the masks affect the MSE and BER performances obtained from equalizers. We consider two constraint scenarios. The first imposes only energy constraints and relaxes the monic constraint in (11). This scenario sets the lower and upper bounds as $c_{1,\min} = 0.25$, $c_{2,\min} = 0.5$, $c_{1,\max} = 2$, and $c_{2,\max} = 4$. In the second constraint scenario, the monic constraint is additionally imposed on both $\mathbf{g}_1$ and $\mathbf{g}_2$, which automatically sets $c_{1,\min} = c_{2,\min} = 1$. In both scenarios, the upper bounds on the superposition weights are set to the nominal values used in the ideal convolutional model in [6] for nominal cross-track positions of the read heads, i.e., $p_1 = 0.9092$ and $p_2 = 0.5$. Fig. 10 shows the MSE performance of the 2-D-LMMSE and the CNN equalizer. It is clear that the CNN equalizer outperforms the 2-D-LMMSE. Furthermore, although imposing the monic constraint increases the MSE, the detector BERs benefit considerably from including the



(a)



(b)

Fig. 10. MSE versus the equalizer iteration with a constrained MSE solver for linear and nonlinear CNN equalizers. (a) Unit energy constraint. (b) Monic constraint.

TABLE I

AVERAGE MSE AND BER COMPARISON FOR THE CNN EQ.-1024-STATE VA SYSTEM UNDER DIFFERENT CONSTRAINTS ON THE TARGETS

Constraint \ Metric	Average MSE	Upper BER	Lower BER
Energy Constraint	0.05037	0.08370	0.2506
Monic Constraint	0.2536	0.06733	0.1190

constraint in the equalization, especially for the lower layer's BER, as shown in Table I.

B. Effect of Residual Paths on Training the CNN Detector

To investigate the benefits of adding the residual paths to the CNN detector, we train CNN detector D0 and CNN detector D1, where CNN detector D0 is a deeper version of CNN detector D1 without the residual paths. CNN detector D0 is constructed based on CNN detector D1 such that both architectures have the same number of parameters. The layers along the residual paths (in CNN detector D1) are integrated along the main paths such that layers with the same number of filters are consecutive in CNN detector D0. Both architectures are trained in this comparison with a fixed mini-batch size

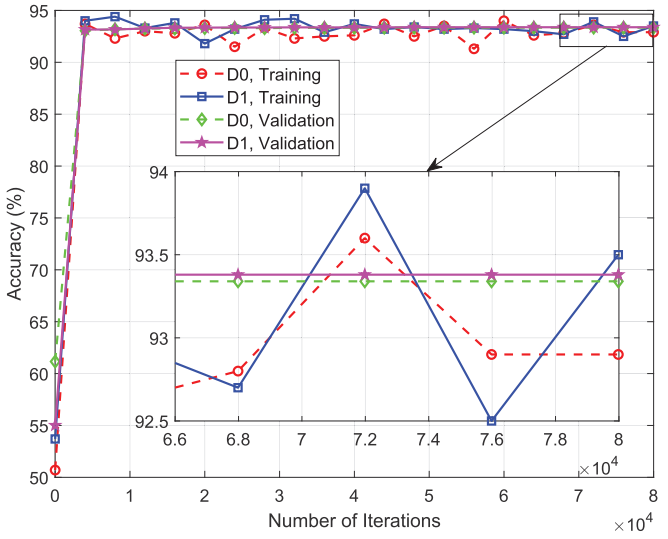


Fig. 11. Accuracy of bit detection versus the number of training iterations.

TABLE II

BERS FOR THE ONE-LAYER TDMR SYSTEM. THE INDIVIDUAL LAYERS' SIGNALS (INSTEAD OF THE SUPERPOSITION IN MLMR) ARE USED TO OBTAIN THE BERS

Method \ Layer	Upper	Lower
VA	0.07939	0.02805
CNN Detector	0.0563	$< 10^{-3}$

of 1000 samples at an initial learning rate of 0.01, which is set to decrease by a factor of 0.4 every two epochs. The total number of epochs for training is 16, and the training accuracy on a random mini-batch and the validation accuracy are plotted every 4000 iterations. In Fig. 11, the learning curves resulting from training CNN detector D0 and CNN detector D1 are shown. The training trajectory is similar for both architectures except that the validation accuracies saturate at about 93.34% and 93.38% for D0 and D1, respectively. This difference in final validation accuracy translates to BERs of 0.066492 and 0.066212 on the testing data set achieved by D0 and D1, respectively. Hence, using the residual paths in the CNN detector can provide small BER improvements, given the same number of learnable parameters.

C. Detector BERs

As a reference, in Table II, we evaluate the BER for one-layer TDMR systems comprised of the same dimensions as the layers of the MLMR system. Table III shows the detector BERs on each layer for the MLMR system using the baseline and proposed architectures. To use the 1024-state VA directly on raw readings, an MSE-optimal 2-D ISI/ITI mask is fitted to the readings and used for computing the branch labels. Due to the sources of distortions discussed in Section II-A, the BER performance of the 1024-state VA is poor. Preprocessing the waveforms with a 2-D-LMMSE equalizer improves the BERs out of the 1024-state VA.

TABLE III
DETECTOR BERS FOR THE MLMR SYSTEM

Method \ Layer	Upper BER	Lower BER
Channel BER	0.2247	0.2135
1024-State VA	0.1861	0.2080
2D-LMMSE-1024-State VA	0.1335	0.1812
CNN Eq.-1024-State VA	0.06733	0.1190
CNN Eq./Sep. S1-2-State VA	0.06874	0.1083
CNN Eq./Sep. S2-2-State VA	0.06811	0.1048
CNN Eq./Sep. S3-2-State VA	0.06854	0.1076
CNN Eq./Sep. S4-2-State VA	0.07079	0.1289
CNN Eq./Sep. S5-2-State VA	0.06978	0.1190
CNN Eq./Sep. S6-2-State VA	0.07223	0.1225
CNN Eq./Sep. S7-2-State VA	0.06809	0.1096
CNN Eq./Sep. S3-4-State VA	0.06985	0.1150
CNN Eq./Sep. S6-4-State VA	0.07071	0.1146
CNN Eq./Sep. S7-4-State VA	0.06889	0.1087
CNN Detector D1	0.06610	0.1020
CNN Detector D2	0.06758	0.1133
CNN Detector D3	0.06867	0.1290
CNN Detector D4	0.07304	0.1399

However, despite such improvement, relatively low code rates would need to be used to correct most errors. Using a CNN equalizer before the 1024-state VA results in more significant reductions in the BER. CNN detector D1 performs similar to the CNN equalizer-1024-state VA and the CNN equalizer/separator S1 two-state VA system. The small reductions in the CNN detector BERs are attributed to the fact that it trains directly on CE loss. A low average CE loss corresponds directly to a low BER and good log-likelihood ratios (LLRs) distribution. Minimizing the MSE under the monic constraint has been shown to provide close to minimum BERs in the work by Moon and Zeng [27]. Consistent with the thesis in [27], our CE-trained CNN detector provides the minimum BERs that are closely followed by the equalizer-VA systems' BERs.

Different architectures of the equalizer/separator-VA system with varying complexities are also investigated. The CNN equalizer/separator architectures S2–S7 are summarized in Table IV. The CNN equalizer/separator provides a good tradeoff between performance and complexity. Increasing the number of states of the VA from two to four corresponding to using a three-tap target provides minor BER improvements for the low-complexity architecture S6. However, the same increase in the number of states does not improve the BER performance of the relatively high complexity architecture S3. A further increase in the number of states to eight did not yield any reductions in the BER for S6. Hence, additional experiments with increasing the number of states of the VA for the equalizer/separator system were not pursued.

D. Density Results

We are interested in whether the effective information areal density of the MLMR is significantly higher than the information areal density of the TDMR one-layer system. Following CNN detector D1, we interfaced an IRA channel decoder that performs coset-decoding using appropriate code rates [28, Sec. IV-B]. We then adjusted the rates via code design and puncturing so that the decoder BER is less than 10^{-5}. This results in a maximum code rate of 0.7477 achieved by the one-layer TDMR system. In comparison, the maximum code rates achieved by the two-layer MLMR system are 0.7116 and 0.6289 on the upper and lower layers, respectively. Since there are four bits on the upper layer per one bit on the lower layer, the effective rate of the MLMR system is $0.7116 + 0.6289/4 = 0.8688$. Hence, the areal density gain of the MLMR system over the TDMR system is $(0.8688 - 0.7477)/0.7477 = 16.20\%$. Using CNN detector D2, the maximum code rates achieved are 0.7137 and 0.5945 for the upper and lower layers, respectively, in the two-layer structure. Thus, the small BER gains on the upper layer due to using D1 over D2 do not yield a higher information density on the upper layer. However, using D1 over D2 gives about 5.79% information density increase for the lower layer.

TABLE IV
SUMMARY OF THE TOPOLOGIES OF THE CNN EQ./SEP. ARCHITECTURES CONSIDERED. BATCH NORMALIZATION AND THE LEAKY RELU ACTIVATION (WITH $a = 0.1$) ARE USED BETWEEN THE CONVOLUTIONAL LAYERS

CNN Architecture	Topology Summary
CNN Eq./Sep. S1	Displayed in Fig. 7
CNN Eq./Sep. S2	[3 × 11, conv. 16], [3 × 3, conv. 8], [3 × 3, conv. 4]
CNN Eq./Sep. S3	[3 × 11, conv. 8], [3 × 3, conv. 4], [3 × 3, conv. 2]
CNN Eq./Sep. S4	[3 × 11, conv. 4], [3 × 3, conv. 2], [3 × 3, conv. 1]
CNN Eq./Sep. S5	[3 × 11, conv. 1], [3 × 3, conv. 8]
CNN Eq./Sep. S6	[3 × 11, conv. 1], [3 × 3, conv. 4]
CNN Eq./Sep. S7	[3 × 11, conv. 2], [3 × 3, conv. 4], [3 × 3, conv. 8]

To obtain a density gain with the CNN equalizer and separator system, appropriate IRA decoders are also interfaced with the CNN equalizer/separator S3 system. The maximum code rate achieved by the reference one-layer TDMR system is 0.7157. In contrast, the two-layer MLMR system achieves the maximum code rates of 0.6861 and 0.5911, respectively. Thus, the CNN equalizer/separator system achieves a density gain of 16.51% for two-layer magnetic recording over the one-layer system. Note that, for the two-layer system, the overall density achieved by the CNN detector is 18.56% higher than the overall density achieved by the CNN equalizer/separator system. The reason is that the CNN detector is trained to detect bits directly from readings using the CE loss. However, the CNN equalizer/separator system is trained on MSE. The MSE loss may not necessarily preserve all relevant information needed for optimal soft detection.

To observe the impact of the reader electronics' noise on the density gain, we added AWGN with variance σ_e^2 such that the SNR for coded bits is given by

$$\mathrm{SNR}_{\text{coded, AWGN}} = 10 \log_{10}\left(\frac{P_{\text{avg}}}{\sigma_e^2}\right) \tag{21}$$

TABLE V
COMPLEXITY COMPARISON IN TERMS OF THE NUMBER OF OPERATIONS PER BIT ESTIMATE. THE CNN DETECTOR ALSO REQUIRES TWO exp(·) COMPUTATIONS PER BIT ESTIMATE

Method \ Operation	Multiplications	Additions	Comparisons
2D-LMMSE Eq.-1024-State VA	2478	4935	6984
CNN Eq.-1024-State VA	58,840	61,298	7759
CNN Eq./Sep. S1-2-State VA	101,466	101,471	1787
CNN Eq./Sep. S2-2-State VA	116,670	116,675	1787
CNN Eq./Sep. S3-2-State VA	34,444	34,450	895
CNN Eq./Sep. S4-2-State VA	11,251	11,256	449
CNN Eq./Sep. S5-2-State VA	5495	5501	347
CNN Eq./Sep. S6-2-State VA	3362	3368	194
CNN Eq./Sep. S7-2-State VA	22,411	22,417	895
CNN Detector D1	1,088,057	1,085,000	9486
CNN Detector D2	33,227	32771	1224
CNN Detector D3	9629	9403	765
CNN Detector D4	2033	1960	153

where $P_{\text{avg}} = (1/3)(P_L + P_C + P_R)$, and $P_i = \mathbb{E}\{\mathbf{r}^2_{\text{Upper},i}\}$ for $i \in \{L, C, R\}$ is the average power of the upper layer's signal $\mathbf{r}_{\text{Upper},i}$ in the superposition reading $\mathbf{r}_i$. Note that the SNR is computed with respect to the upper layer's signal as in [6], which allows for a fair comparison between the one- and two-layer systems. We tested CNN detector D2 with $\text{SNR}_{\text{coded, AWGN}} = 20$ dB. This increased the detector BERs to 0.08243 and 0.1266 for the upper and lower layers, respectively. The maximum code rate obtained for the one-layer reference system is 0.7076. In comparison, the maximum code rates for the two-layer system are 0.6695 and 0.5577 for the upper and lower layers, respectively. Hence, the density gain of the two-layer system is 14.32%.

E. Complexity Comparison

This section presents a complexity comparison between the baseline 2-D-LMMSE equalizer-1024 state VA, the CNN equalizer-1024-state VA, and the CNN detector. Table V details the number of operations per bit estimate required by each system. For the CNN-based systems, the operations include the multiplications and additions due to 2-D-convolution, batch normalization, matrix multiplication at FC layers, leaky ReLU, and residual paths' associated additions. The comparisons performed by the CNN systems are only to obtain the sign of hidden outputs to compute the ReLU and leaky ReLU operations. For the VA, the multiplications and additions are due to branch and path metric computations. The comparisons in the VA are performed between accumulated path metrics to determine the surviving path at each trellis state.

Since the input to the CNNs is a sliding window over the readings, many of the hidden outputs are equal between sample estimates and need not be computed for every bit estimate in practice (see the discussion in [10, Sec. IV]). However, because such simplifications were not taken into account in our implementation, they are not reflected in our operations counting. Furthermore, our current implementation of the CNN detector estimates one bit during each sampling interval. In contrast, the VA estimates five bits every two sampling intervals, which saves complexity on the number of operations required per bit estimate. The CNN detector can also be configured to estimate more than one bit every sampling time using appropriate mapping of short bit sequences to classification classes. The CNN equalizer used with the 1024-state VA provides three equalized samples for every sampling interval. The complexity of the CNN detector for the lower layer can be reduced by decreasing the size of the input using an averaging filter over the extra samples per bit.

V. CONCLUSION

Using realistic readback waveforms for a two-layer magnetic recording system, we investigated the performances of a conventional linear equalizer, trellis-based detectors, and convolutional NNs (CNNs) for equalization and detection. Due to the presence of pattern-dependent media noise and jitter noise in the readback waveforms, the CNN systems outperformed the conventional system comprised of a linear equalizer followed by a trellis-based detector. The linear equalizer cannot handle well equalizing signals comprised of the superposition of two waveforms. As a non-linear equalizer/detector trained on raw readings and a universal function approximator, the CNN is better equipped to compensate for the sources of distortion and achieve better performance. The CNN systems provide a good performance–complexity tradeoff. We interfaced the CNN detector system with channel decoders to obtain a density estimate for the two-layer system. Our experiments yield a 16.2% areal density increase for two-layer magnetic recording over one-layer recording. When the reader electronics' noise is considered, a moderate complexity CNN detector achieves an areal density gain of 14.32%. Future research works can optimize the design of the channel encoder and decoder architectures for the two-layer system.

APPENDIX
DERIVATION OF THE 2-D-LMMSE EQUALIZER

Let M denote the order of $\mathbf{f}_{1,j}$. The MSE, σ_e^2, between the ideal PR signal and the linear equalizer output is given by

$$\sigma_e^2[k] = \mathbb{E}\big[\mathbf{y}_L[k] - \big(\mathbf{f}_{1,L} * \mathbf{r}_L + \mathbf{f}_{1,C} * \mathbf{r}_C + \mathbf{f}_{1,R} * \mathbf{r}_R\big)[k]\big]^2. \tag{22}$$

We drop the subscript index 1 from $\mathbf{f}_{1,j}$, $j \in L, C, R$, for convenience of notation, since it is clear from context that we are solving for $\mathbf{f}_{1,j}$. Writing out the convolution and simplifying gives

$$\sigma_e^2[k] = \mathbb{E}\left[y_L[k] - \sum_j \sum_{m=0}^{M} f_j[m] r_j[k-m] \right]^2 \tag{23}$$

$$= \mathbb{E}[y_L[k]]^2 - 2\sum_j \sum_m f_j[m]\, \mathbb{E}\big[y_L[k] r_j[k-m]\big] + \cdots$$

$$\cdots + \sum_{j,j'} \sum_{m,m'} f_j[m] f_{j'}\big[m'\big]\, \mathbb{E}\big[r_j[k-m] r_{j'}\big[k-m'\big]\big]. \tag{24}$$

Define the auto-correlation vector $R_{j,j'}[m-m'] \triangleq \mathbb{E}[r_j[k-m]r_{j'}[k-m']]$ and the cross correlation vector $q_j[m] \triangleq \mathbb{E}[y_L[k]r_j[m-k]]$. Note that the definitions implicitly assume that $r_j[k]$'s are wide sense stationary (WSS) random processes. Taking the derivative of (24) with respect to $f_j[m]$ and setting the outcome to zero, we obtain the MSE minimizing $\mathbf{f}_j$ by solving

$$\sum_{j'} R_{j,j'}\big[m-m'\big] f_{j'}[m] = q_j[m]. \tag{25}$$

We can write the condition in (25) in matrix form. Let $\mathbf{R}_{j,j'}$ denote a Toeplitz matrix (of size $M+1 \times M+1$) constructed from the vector $[R_{j,j'}[0], R_{j,j'}[1], \ldots, R_{j,j'}[M]]$ and $\mathbf{q}_j$ denote the column vector $\mathrm{col}[q_j[0], q_j[1], \ldots, q_j[M]]$. If $\mathbf{f}_j$ is a column vector of length $M+1$, then the solution $\mathbf{f}_j$ solves the equation

$$\begin{bmatrix} \mathbf{R}_{L,L} & \mathbf{R}_{L,C} & \mathbf{R}_{L,R} \\ \mathbf{R}_{C,L} & \mathbf{R}_{C,C} & \mathbf{R}_{C,R} \\ \mathbf{R}_{R,L} & \mathbf{R}_{R,C} & \mathbf{R}_{R,L} \end{bmatrix} \begin{bmatrix} \mathbf{f}_L \\ \mathbf{f}_C \\ \mathbf{f}_R \end{bmatrix} = \begin{bmatrix} \mathbf{q}_L \\ \mathbf{q}_C \\ \mathbf{q}_R \end{bmatrix} \tag{26}$$

where the leftmost matrix is a doubly block Toeplitz matrix of size $3(M+1) \times 3(M+1)$, and the central and right-hand side column vectors both have length $3(M+1)$.

Note that, given observations of length N, the auto- and cross-correlations in (26) are estimated empirically as (for $m \geq 0$)

$$\hat{R}_{j,j'}[m] = \frac{1}{N} \sum_{n=0}^{N-m-1} r_j[n+m] r_{j'}[n] \tag{27}$$

$$\hat{q}_j[m] = \frac{1}{N} \sum_{n=0}^{N-m-1} y_L[n+m] r_j[n]. \tag{28}$$

ACKNOWLEDGMENT

This work was supported in part by the United States National Science Foundation under Grant CCF-1817083 and in part by a gift from the Advanced Storage Research Consortium.

REFERENCES

[1] R. Wood, M. Williams, A. Kavcic, and J. Miles, "The feasibility of magnetic recording at 10 terabits per square inch on conventional media," *IEEE Trans. Magn.*, vol. 45, no. 2, pp. 917–923, Feb. 2009.

[2] S. J. Greaves, K. S. Chan, and Y. Kanai, "Areal density capability of dual-structure media for microwave-assisted magnetic recording," *IEEE Trans. Magn.*, vol. 55, no. 12, Dec. 2019, Art. no. 6701509.

[3] S. J. Greaves, K. S. Chan, and Y. Kanai, "Optimization of dual-structure recording media for microwave-assisted magnetic recording," *IEEE Trans. Magn.*, vol. 55, no. 7, Jul. 2019, Art. no. 3001305.

[4] K. S. Chan, S. Greaves, and S. Rahardja, "Optimization of the 3-D-MAMR media stack," *IEEE Trans. Magn.*, vol. 55, no. 9, Sep. 2019, Art. no. 7204905.

[5] K. S. Chan, R. Wood, and S. Rahardja, "Maximum likelihood detection for 3-D-MAMR," *IEEE Trans. Magn.*, vol. 55, no. 12, Dec. 2019, Art. no. 3002609.

[6] K. S. Chan, A. Aboutaleb, K. Sivakumar, B. Belzer, R. Wood, and S. Rahardja, "Data recovery for multilayer magnetic recording," *IEEE Trans. Magn.*, vol. 55, no. 12, Dec. 2019, Art. no. 6701216.

[7] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. London, U.K.: Pearson, 1995.

[8] C. K. Matcha and S. G. Srinivasa, "Generalized partial response equalization and data-dependent noise predictive signal detection over media models for TDMR," *IEEE Trans. Magn.*, vol. 51, no. 10, Oct. 2015, Art. no. 3101215.

[9] A. Sayyafan, B. J. Belzer, K. Sivakumar, J. Shen, K. S. Chan, and A. James, "Deep neural network based media noise predictors for use in high-density magnetic recording turbo-detectors," *IEEE Trans. Magn.*, vol. 55, no. 12, Dec. 2019, Art. no. 6701406.

[10] S. K. Nair and J. Moon, "Data storage channel equalization using neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1037–1048, Sep. 1997.

[11] J. Shen and N. Nangare, "Nonlinear equalization for TDMR channels using neural networks," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2020, pp. 1–6.

[12] J. Shen, A. Aboutaleb, K. Sivakumar, B. J. Belzer, K. S. Chan, and A. James, "Deep neural network a posteriori probability detector for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 56, no. 6, Jun. 2020, Art. no. 3100212.

[13] K. Luo *et al.*, "A study on block-based neural network equalization in TDMR system with LDPC coding," *IEEE Trans. Magn.*, vol. 55, no. 11, Nov. 2019, Art. no. 6700605.

[14] Z.-M. Yuan, C. L. Ong, S. H. Leong, T. Zhou, and B. Liu, "3-D sensitivity function of shielded reader by reciprocity principle," *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 1929–1932, Jun. 2010.

[15] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[17] A. Heinecke, J. Ho, and W. Hwang, "Refinement and universal approximation via sparsely connected ReLU convolution nets," *IEEE Signal Process. Lett.*, vol. 27, pp. 1175–1179, 2020, doi: 10.1109/LSP.2020.3005051.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[21] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018, *arXiv:1811.03378*. [Online]. Available: http://arxiv.org/abs/1811.03378

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[24] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, no. 1, p. 3.

[25] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: http://arxiv.org/abs/1609.04747

[26] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," 2017, *arXiv:1711.00489*. [Online]. Available: http://arxiv.org/abs/1711.00489

[27] J. Moon and W. Zeng, "Equalization for maximum likelihood detectors," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083–1088, Mar. 1995.

[28] X. Sun *et al.*, "ISI/ITI turbo equalizer for TDMR using trained local area influence probabilistic model," *IEEE Trans. Magn.*, vol. 55, no. 4, Apr. 2019, Art. no. 3100515.