Digital Backpropagation with Deep-Learned Chromatic Dispersion Filters

Christian Häger[1,2]

Joint work with: Henry D. Pfister[2], Christoffer Fougstedt[3], Lars Svensson[3], and Per Larsson-Edefors[3]

[1]Department of Electrical Engineering, Chalmers University of Technology, Gothenburg
[2]Department of Electrical and Computer Engineering, Duke University, Durham
[3]Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg

UCL, London, August 21, 2018

FORCE FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

Duke UNIVERSITY

CHALMERS

Machine Learning and Fiber-Optic Communications

- What can machine learning contribute to the design of fiber-optic communication systems?

Machine Learning and Fiber-Optic Communications

- What can machine learning contribute to the design of fiber-optic communication systems? Many topics and references:
  - Equalization [Shen and Lau, 2011], [Jarajreh et al., 2015], [Giacoumidis et al., 2015], [Zibar et al., 2016], . . . ,
  - Performance monitoring [Xiaoxia et al., 2009], [Khan et al., 2012], [Tanimura et al., 2016], . . .
  - End-to-end autoencoder learning [Lee et al., 2018], [Karanov et al., 2018], [Jones et al., 2018], [Li et al., 2018], . . .
  - . . .

Machine Learning and Fiber-Optic Communications

- What can machine learning contribute to the design of fiber-optic communication systems? Many topics and references:
  - Equalization [Shen and Lau, 2011], [Jarajreh et al., 2015], [Giacoumidis et al., 2015], [Zibar et al., 2016], . . . ,
  - Performance monitoring [Xiaoxia et al., 2009], [Khan et al., 2012], [Tanimura et al., 2016], . . .
  - End-to-end autoencoder learning [Lee et al., 2018], [Karanov et al., 2018], [Jones et al., 2018], [Li et al., 2018], . . .
  - . . .
- We revisit efficient nonlinearity compensation via digital backpropagation

Machine Learning and Fiber-Optic Communications

- What can machine learning contribute to the design of fiber-optic communication systems? Many topics and references:
  - Equalization [Shen and Lau, 2011], [Jarajreh et al., 2015], [Giacoumidis et al., 2015], [Zibar et al., 2016], . . . ,
  - Performance monitoring [Xiaoxia et al., 2009], [Khan et al., 2012], [Tanimura et al., 2016], . . .
  - End-to-end autoencoder learning [Lee et al., 2018], [Karanov et al., 2018], [Jones et al., 2018], [Li et al., 2018], . . .
  - . . .
- We revisit efficient nonlinearity compensation via digital backpropagation

This talk

1. Is not about black-box neural networks . . .

Machine Learning and Fiber-Optic Communications

- What can machine learning contribute to the design of fiber-optic communication systems? Many topics and references:
  - Equalization [Shen and Lau, 2011], [Jarajreh et al., 2015], [Giacoumidis et al., 2015], [Zibar et al., 2016], . . . ,
  - Performance monitoring [Xiaoxia et al., 2009], [Khan et al., 2012], [Tanimura et al., 2016], . . .
  - End-to-end autoencoder learning [Lee et al., 2018], [Karanov et al., 2018], [Jones et al., 2018], [Li et al., 2018], . . .
  - . . .
- We revisit efficient nonlinearity compensation via digital backpropagation

This talk

1. Is not about black-box neural networks . . . but we uncover and exploit an interesting connection between neural networks and the split-step method

Machine Learning and Fiber-Optic Communications

- What can machine learning contribute to the design of fiber-optic communication systems? Many topics and references:
  - Equalization [Shen and Lau, 2011], [Jarajreh et al., 2015], [Giacoumidis et al., 2015], [Zibar et al., 2016], . . . ,
  - Performance monitoring [Xiaoxia et al., 2009], [Khan et al., 2012], [Tanimura et al., 2016], . . .
  - End-to-end autoencoder learning [Lee et al., 2018], [Karanov et al., 2018], [Jones et al., 2018], [Li et al., 2018], . . .
  - . . .
- We revisit efficient nonlinearity compensation via digital backpropagation

This talk

1. Is not about black-box neural networks . . . but we uncover and exploit an interesting connection between neural networks and the split-step method
2. We use deep learning to jointly optimize, prune, and quantize all linear substeps $\Longrightarrow$ ASIC power consumption becomes comparable to linear equalization, even with multiple steps per span

Machine Learning and Fiber-Optic Communications

- What can machine learning contribute to the design of fiber-optic communication systems? Many topics and references:
  - Equalization [Shen and Lau, 2011], [Jarajreh et al., 2015], [Giacoumidis et al., 2015], [Zibar et al., 2016], . . . ,
  - Performance monitoring [Xiaoxia et al., 2009], [Khan et al., 2012], [Tanimura et al., 2016], . . .
  - End-to-end autoencoder learning [Lee et al., 2018], [Karanov et al., 2018], [Jones et al., 2018], [Li et al., 2018], . . .
  - . . .
- We revisit efficient nonlinearity compensation via digital backpropagation

This talk

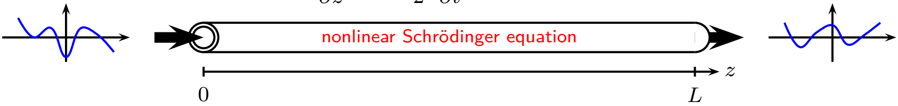
1. Is not about black-box neural networks . . . but we uncover and exploit an interesting connection between neural networks and the split-step method
2. We use deep learning to jointly optimize, prune, and quantize all linear substeps $\Longrightarrow$ ASIC power consumption becomes comparable to linear equalization, even with multiple steps per span
3. No step-reducing approaches: the spirit behind “deep learning” and “reducing steps” are fundamentally opposed

Outline

Outline

Digital Backpropagation

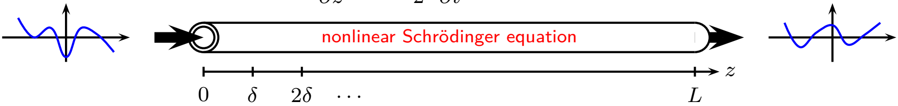
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



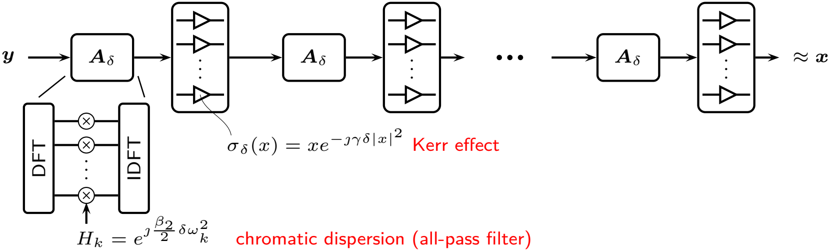
- Invert a partial differential equation in real time ([Paré et al., 1996], [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008])

Digital Backpropagation

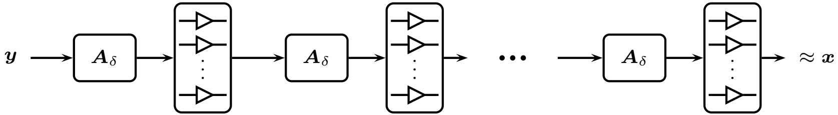
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



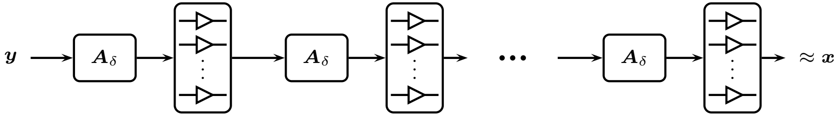
- Invert a partial differential equation in real time ([Paré et al., 1996], [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008])
- Split-step Fourier method with M steps ($\delta = L/M$):



Real-Time Digital Backpropagation

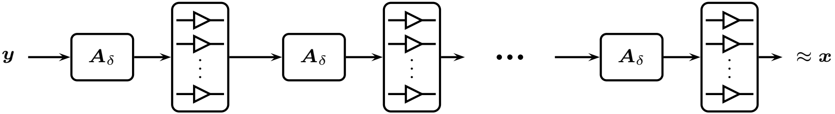


Real-Time Digital Backpropagation



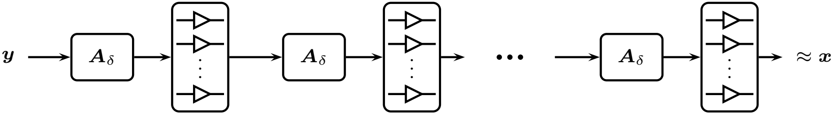
- Widely considered to be impractical (too complex): linear equalization is already one of the most power-hungry DSP blocks in coherent receivers

Real-Time Digital Backpropagation



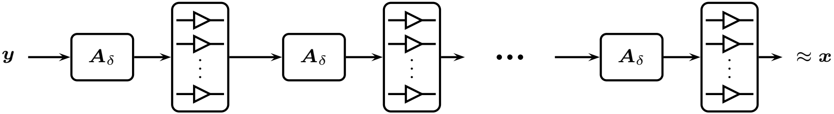
- Widely considered to be impractical (too complex): linear equalization is already one of the most power-hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], ...)

Real-Time Digital Backpropagation



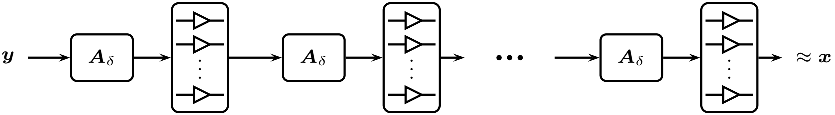
- Widely considered to be impractical (too complex): linear equalization is already one of the most power-hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)
- Intuitive, but . . .

Real-Time Digital Backpropagation



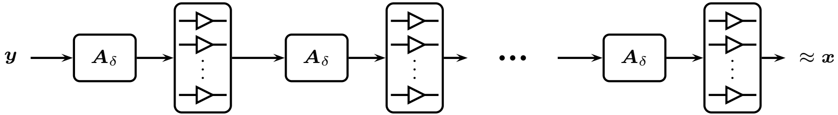
- Widely considered to be impractical (too complex): linear equalization is already one of the most power-hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], ...)
- Intuitive, but ... this flattens a deep (multi-layer) computation graph

Real-Time Digital Backpropagation



- Widely considered to be impractical (too complex): linear equalization is already one of the most power-hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)
- Intuitive, but . . . this flattens a deep (multi-layer) computation graph
- Machine learning: deep computation graphs tend to work better and can be more parameter efficient than shallow ones

Real-Time Digital Backpropagation



- Widely considered to be impractical (too complex): linear equalization is already one of the most power-hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)
- Intuitive, but . . . this flattens a deep (multi-layer) computation graph
- Machine learning: deep computation graphs tend to work better and can be more parameter efficient than shallow ones

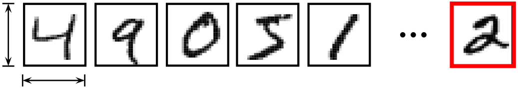
Main contribution

Joint optimization and sparsification of all linear substeps leads to efficient digital backpropagation, even with many steps.

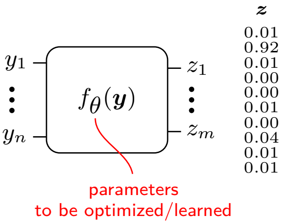
Outline

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

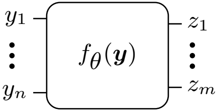


28×28 pixels $\Longrightarrow n = 784$



Supervised Learning

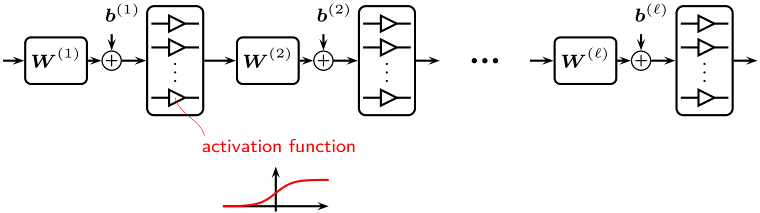
handwritten digit recognition (MNIST: 70,000 images)



$\boldsymbol{z}$

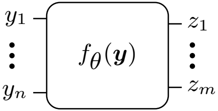
0.01
0.92
0.01
0.00
0.00
0.01
0.00
0.04
0.01
0.01

How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



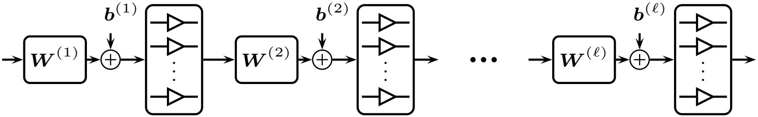
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks

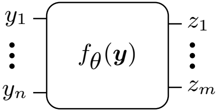


How to optimize $\theta = \{\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(1)}, \dots, \boldsymbol{b}^{(\ell)}\}$? Deep learning

$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k) \qquad (1)$$

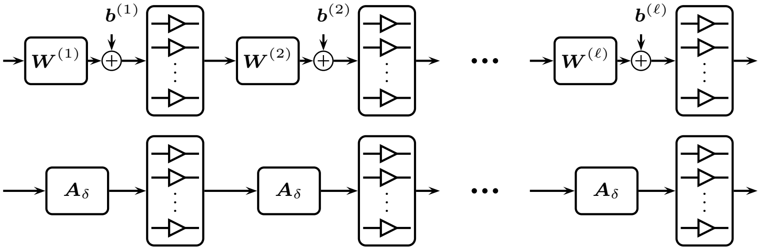
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



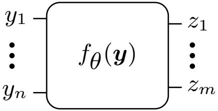
$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



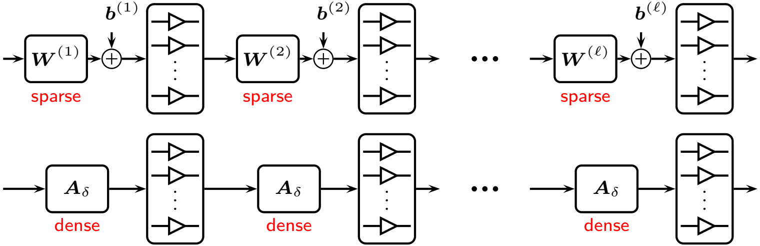
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



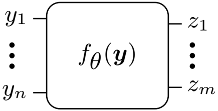
$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



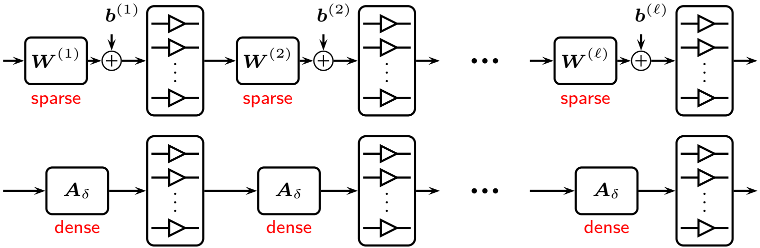
Supervised Learning

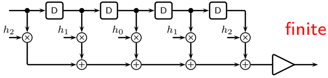
handwritten digit recognition (MNIST: 70,000 images)



$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

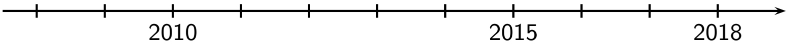
How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



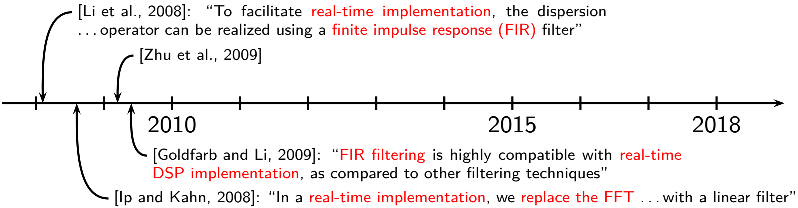


finite impulse response (FIR) filters ?

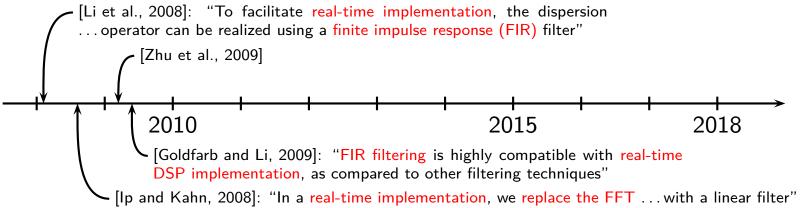
Time-Domain Digital Backpropagation: Literature



Time-Domain Digital Backpropagation: Literature



Time-Domain Digital Backpropagation: Literature

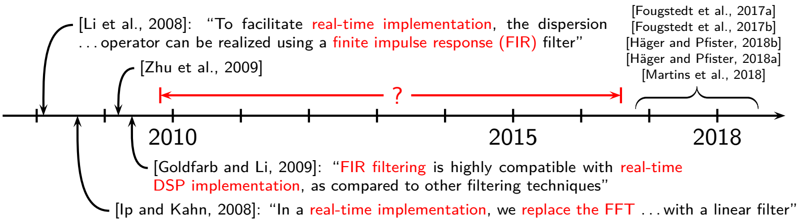


Nontrivial to achieve a good performance–complexity tradeoff!

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000\,\text{km}$ [Ip and Kahn, 2008]

> 1000 total taps required $\implies$ $100\times$ more operations than linear equalization

Time-Domain Digital Backpropagation: Literature

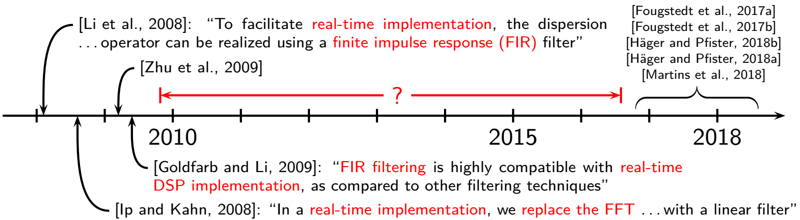


Nontrivial to achieve a good performance–complexity tradeoff!

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000\,$km [Ip and Kahn, 2008]

> 1000 total taps required $\implies$ $100\times$ more operations than linear equalization

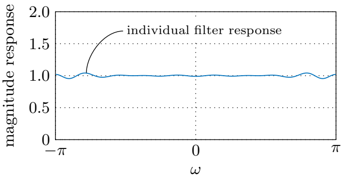
Time-Domain Digital Backpropagation: Literature

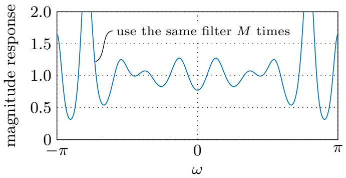


Nontrivial to achieve a good performance–complexity tradeoff!

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000$ km [Ip and Kahn, 2008]

> 1000 total taps required $\implies$ $100\times$ more operations than linear equalization

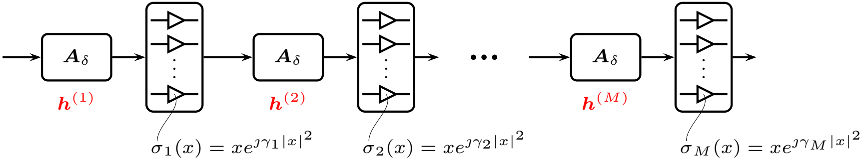




Learned Digital Backpropagation

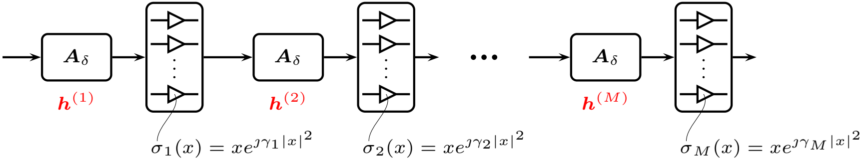
Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



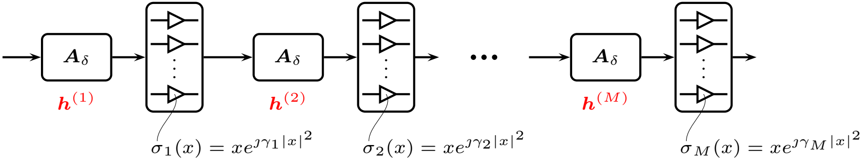
Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}\}$:

$$\min_\theta \sum_{i=1}^{N} \mathsf{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

mean squared error — Adam optimizer, fixed learning rate

Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}\}$:

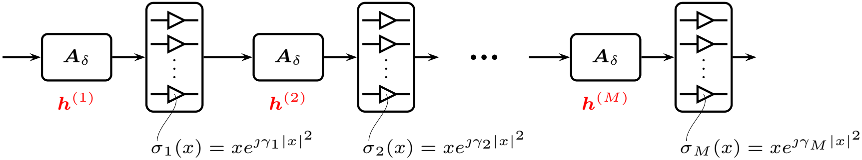
$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

mean squared error — Adam optimizer, fixed learning rate

- How to choose the starting point θ_0 and get short filters?

Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



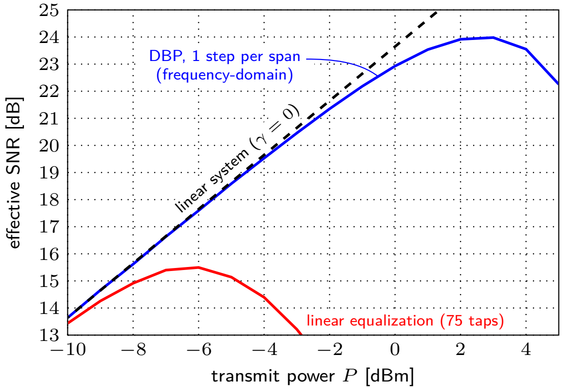
Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \ldots, \boldsymbol{h}^{(M)}\}$:

$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

mean squared error — Adam optimizer, fixed learning rate

- How to choose the starting point θ_0 and get short filters?
- Iteratively prune (set to 0) the outermost filter taps during gradient descent until a certain target filter length is reached

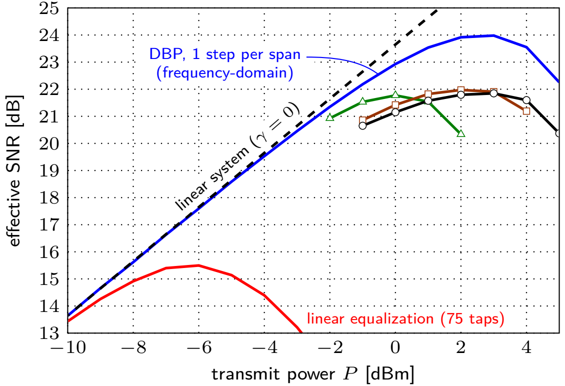
Optimization Results (10.7 Gbaud)



Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

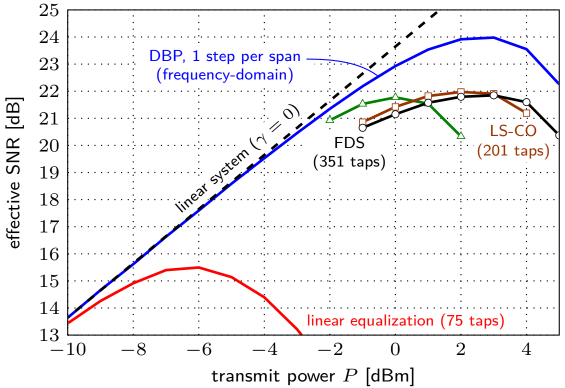
Optimization Results (10.7 Gbaud)



Parameters similar to [Ip and Kahn, 2008]:

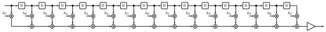
- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

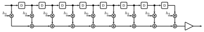
Optimization Results (10.7 Gbaud)



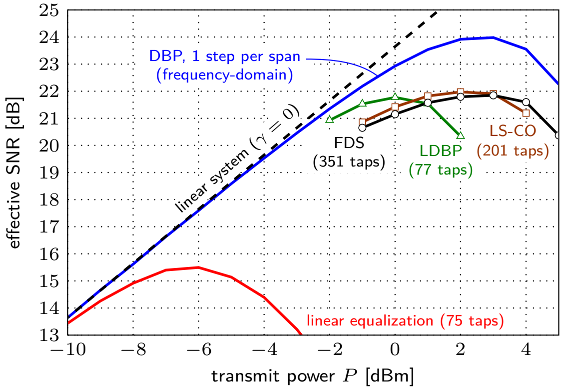
Parameters similar to [Ip and Kahn, 2008]:

- 25 × 80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

FDS  · · · frequency-domain sampling (15 taps per step)

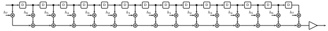
LS-CO  · · · constrained least-squares [Sheikh et al., 2016] (9 taps per step)

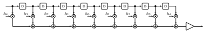
Optimization Results (10.7 Gbaud)

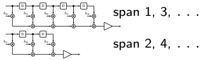


Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

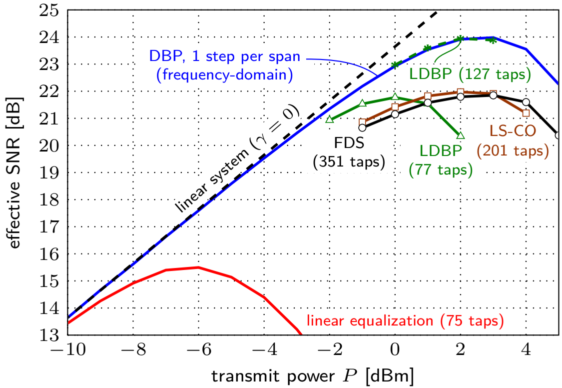
FDS  . . . frequency-domain sampling (15 taps per step)

LS-CO  . . . constrained least-squares [Sheikh et al., 2016] (9 taps per step)

LDBP 

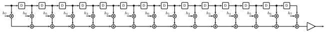
deep learning and pruning (alternate 5 and 3 taps and use different coefficients in all steps)

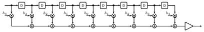
Optimization Results (10.7 Gbaud)

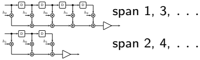


Parameters similar to [Ip and Kahn, 2008]:

- 25 × 80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

FDS  . . . frequency-domain sampling (15 taps per step)

LS-CO  . . . constrained least-squares [Sheikh et al., 2016] (9 taps per step)

LDBP 

deep learning and pruning (alternate 5 and 3 taps and use different coefficients in all steps)

Outline

Filter Coefficient Quantization

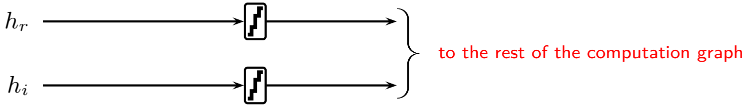
Filter Coefficient Quantization

DSP implementation requires quantized coefficients [Fougstedt et al., 2017a], [Fougstedt et al., 2017b], [Martins et al., 2018], [Sherborne et al., 2018].

Filter Coefficient Quantization

DSP implementation requires quantized coefficients [Fougstedt et al., 2017a], [Fougstedt et al., 2017b], [Martins et al., 2018], [Sherborne et al., 2018].

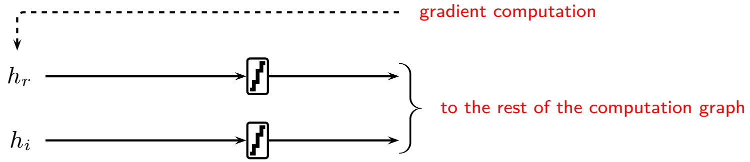
- Apply TensorFlow's "fake quantization" to each filter coefficient variable:



Filter Coefficient Quantization

DSP implementation requires quantized coefficients [Fougstedt et al., 2017a], [Fougstedt et al., 2017b], [Martins et al., 2018], [Sherborne et al., 2018].

- Apply TensorFlow's "fake quantization" to each filter coefficient variable:

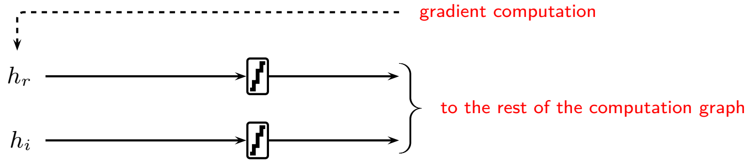


- Fake quantization: gradient computation and parameter updates are still performed in floating point

Filter Coefficient Quantization

DSP implementation requires quantized coefficients [Fougstedt et al., 2017a], [Fougstedt et al., 2017b], [Martins et al., 2018], [Sherborne et al., 2018].

- Apply TensorFlow's "fake quantization" to each filter coefficient variable:

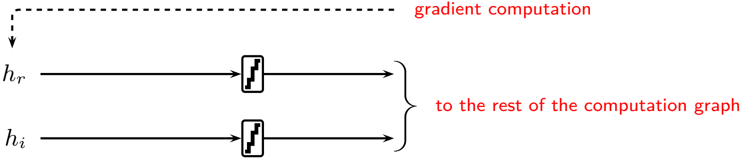


- Fake quantization: gradient computation and parameter updates are still performed in floating point
- Activate after the (floating-point) optimization has converged and continue training for few more iterations

Filter Coefficient Quantization

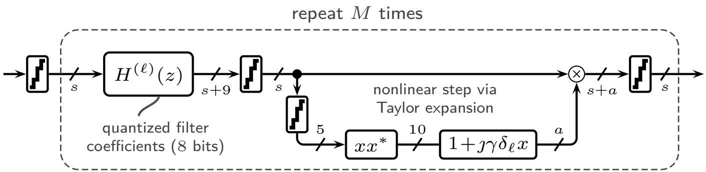
DSP implementation requires quantized coefficients [Fougstedt et al., 2017a], [Fougstedt et al., 2017b], [Martins et al., 2018], [Sherborne et al., 2018].

- Apply TensorFlow's "fake quantization" to each filter coefficient variable:

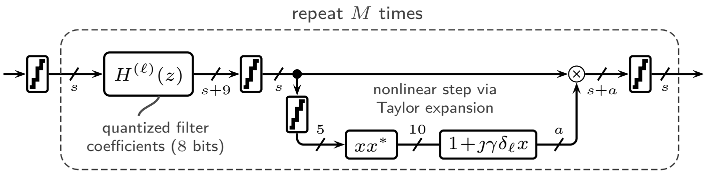


- Fake quantization: gradient computation and parameter updates are still performed in floating point
- Activate after the (floating-point) optimization has converged and continue training for few more iterations
- Joint optimization of quantized impulse responses $\implies$ partial cancellation of quantization-induced frequency-response errors

Hardware Model and Circuit Implementation

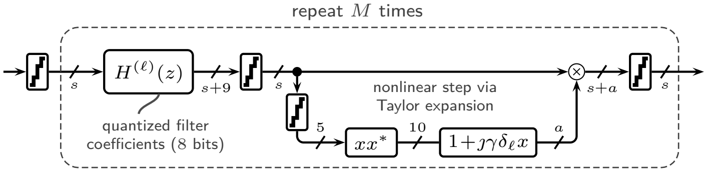


Hardware Model and Circuit Implementation



- Signal requantization to s bits after each FIR filter and nonlinear step

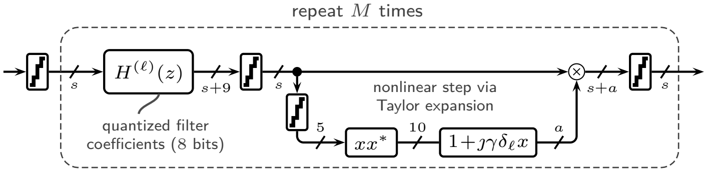
Hardware Model and Circuit Implementation



- Signal requantization to s bits after each FIR filter and nonlinear step
- Nonlinear steps via first-order Taylor expansion [Fougstedt et al., 2017a]:

$$xe^{\jmath\gamma\delta_\ell|x|^2} \approx x(1 + \jmath\gamma\delta_\ell|x|^2)$$

Hardware Model and Circuit Implementation

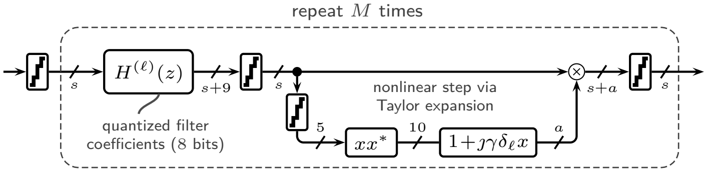


- Signal requantization to s bits after each FIR filter and nonlinear step
- Nonlinear steps via first-order Taylor expansion [Fougstedt et al., 2017a]:

$$xe^{\jmath\gamma\delta_\ell|x|^2} \approx x(1+\jmath\gamma\delta_\ell|x|^2)$$

- 96-parallel VHDL implementation at 416.7 MHz clock speed (40 GHz RX signal), synthesized using a low-power 28-nm CMOS library
- All FIR filters are fully reconfigurable

Hardware Model and Circuit Implementation

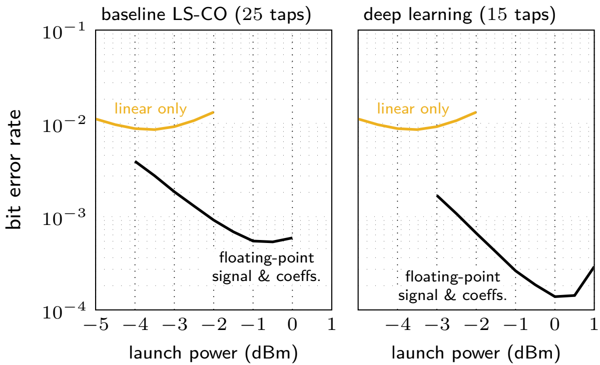


- Signal requantization to s bits after each FIR filter and nonlinear step
- Nonlinear steps via first-order Taylor expansion [Fougstedt et al., 2017a]:

$$xe^{\jmath\gamma\delta_\ell|x|^2} \approx x(1+\jmath\gamma\delta_\ell|x|^2)$$

- 96-parallel VHDL implementation at 416.7 MHz clock speed (40 GHz RX signal), synthesized using a low-power 28-nm CMOS library
- All FIR filters are fully reconfigurable
- Power estimation based on simulation of internal circuit switching statistics

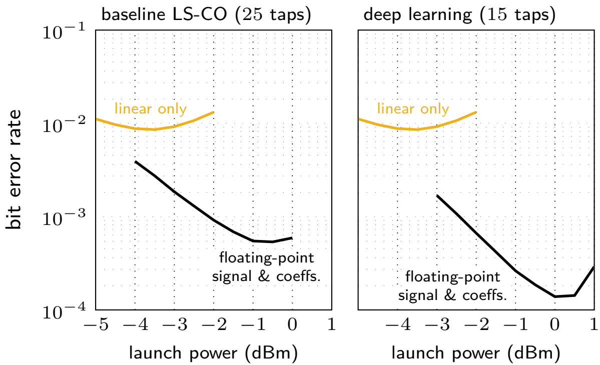
Performance and Power Consumption (20 Gbaud)



System parameters:

- 32×100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

Performance and Power Consumption (20 Gbaud)

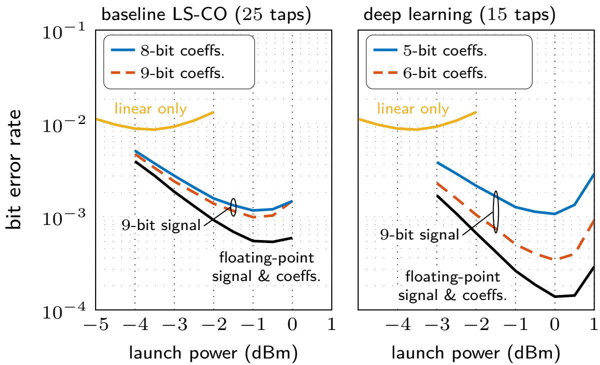


System parameters:

- 32×100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

- Deep learning gives 15-tap filters with better performance

Performance and Power Consumption (20 Gbaud)

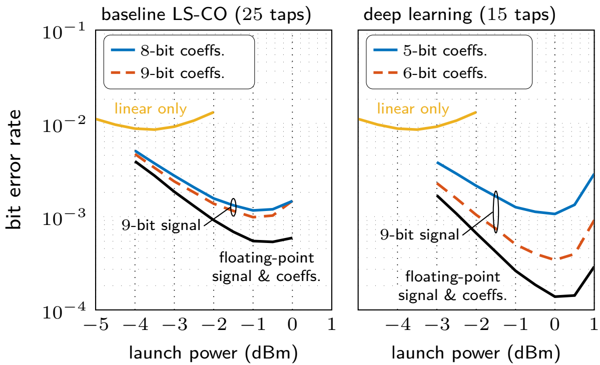


System parameters:

- 32 × 100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

- Deep learning gives 15-tap filters with better performance
- 8–9 signal bits required in both cases, depending on performance
- Deep learning leads to significantly fewer bits for the filter taps

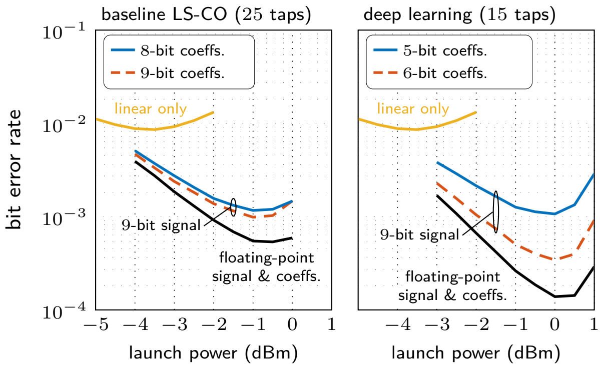
Performance and Power Consumption (20 Gbaud)



System parameters:

- 32 × 100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

Performance and Power Consumption (20 Gbaud)



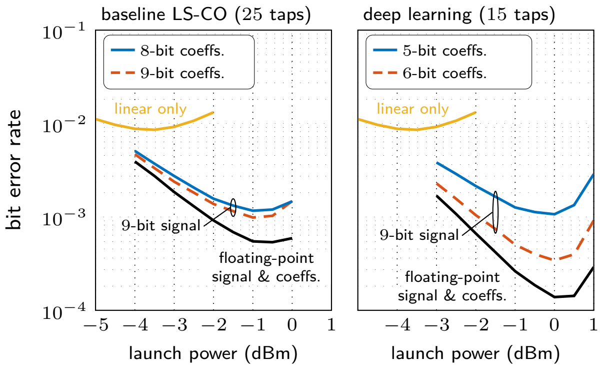
System parameters:

- 32 × 100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

ASIC power per step (100 km)	0.37 W (9-bit coeffs.)	0.18 W (5-bit coeffs.)

> 50% reduction due to fewer taps and bits

Performance and Power Consumption (20 Gbaud)



System parameters:

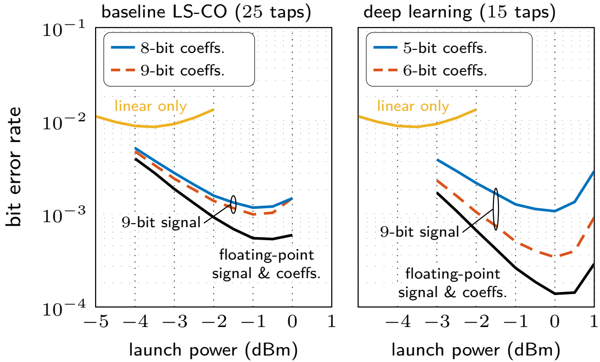
- 32 × 100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

ASIC power per step (100 km)	0.37 W (9-bit coeffs.)	0.18 W (5-bit coeffs.)

> 50% reduction due to fewer taps and bits

[Crivelli et al., 2014]	3500 km	50 Gb/s (12.5 Gbaud)	7 W
[Pillai et al., 2014]	2400 km	100 Gb/s (30 Gbaud)	11 W

Performance and Power Consumption (20 Gbaud)



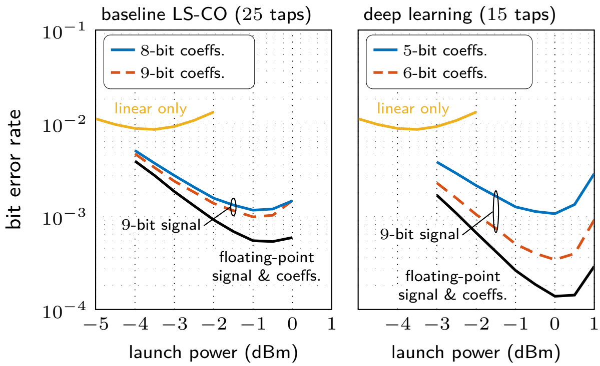
System parameters:

- 32 × 100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

ASIC power per step (100 km)	0.37 W (9-bit coeffs.)	0.18 W (5-bit coeffs.)	> 50% reduction due to fewer taps and bits

[Crivelli et al., 2014]	3500 km	50 Gb/s (12.5 Gbaud)	7 W	$2 \times 35 \times 0.18$ W $\approx$ 13 W
[Pillai et al., 2014]	2400 km	100 Gb/s (30 Gbaud)	11 W	

Performance and Power Consumption (20 Gbaud)

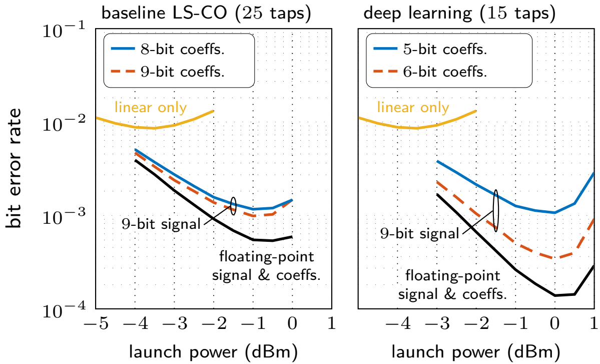


System parameters:

- 32 × 100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

ASIC power per step (100 km)	0.37 W (9-bit coeffs.)		0.18 W (5-bit coeffs.)	> 50% reduction due to fewer taps and bits
[Crivelli et al., 2014]	3500 km	50 Gb/s (12.5 Gbaud)	7 W	2 × 35 × 0.18 W ≈ 13 W
[Pillai et al., 2014]	2400 km	100 Gb/s (30 Gbaud)	11 W	2 × 24 × 0.18 W ≈ 9 W (20 W adjusted for baudrate)

Performance and Power Consumption (20 Gbaud)



System parameters:

- 32×100 km fiber
- 1 step per span
- 16-QAM single pol.
- RRC pulses (0.1 roll-off)
- 20 Gbaud
- 2 samples/symbol
- single channel

ASIC power per step (100 km)	0.37 W (9-bit coeffs.)		0.18 W (5-bit coeffs.)	> 50% reduction due to fewer taps and bits
[Crivelli et al., 2014]	3500 km	50 Gb/s (12.5 Gbaud)	7 W	$2 \times 35 \times 0.18$ W ≈ 13 W
[Pillai et al., 2014]	2400 km	100 Gb/s (30 Gbaud)	11 W	$2 \times 24 \times 0.18$ W ≈ 9 W (20 W adjusted for baudrate)

within factor 2 of published results for static chromatic dispersion compensation

Outline

1. Introduction

2. Connection between Deep Learning and Digital Backpropagation

3. ASIC Implementation Aspects

4. Wideband Digital Backpropagation via Subband Processing

5. Conclusions

Wideband Time-Domain Backpropagation

Wideband Time-Domain Backpropagation

- Single-channel DBP gives limited gain in a WDM scenario
  - Option 1: Consider the time-varying ISI channel
  - Option 2: Increase backpropagation bandwidth (here)

Wideband Time-Domain Backpropagation

- Single-channel DBP gives limited gain in a WDM scenario
  - Option 1: Consider the time-varying ISI channel
  - Option 2: Increase backpropagation bandwidth (here)
- Problem: quadratic dependence of overall system memory on the backpropagated bandwidth

Wideband Time-Domain Backpropagation

- Single-channel DBP gives limited gain in a WDM scenario
  - Option 1: Consider the time-varying ISI channel
  - Option 2: Increase backpropagation bandwidth (here)
- Problem: quadratic dependence of overall system memory on the backpropagated bandwidth

Example

Consider a 96-Gbaud signal, where delay spread is 125 symbol periods per 100 km (alternatively: superchannel or multiple WDM channels).

- Power estimate for 1500 km and 20 Gbaud: $2 \times 15 \times 0.18$ W $= 5.4$ W
- Quadratic scaling: $\approx 25 \times 5.4$ W $= 135$ W (full DBP)
- Linear scaling: $\approx 5 \times 5.4$ W $= 27$ W (5 independent receivers)

Wideband Time-Domain Backpropagation

- Single-channel DBP gives limited gain in a WDM scenario
  - Option 1: Consider the time-varying ISI channel
  - Option 2: Increase backpropagation bandwidth (here)
- Problem: quadratic dependence of overall system memory on the backpropagated bandwidth

Example

Consider a 96-Gbaud signal, where delay spread is 125 symbol periods per 100 km (alternatively: superchannel or multiple WDM channels).

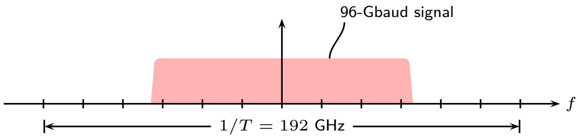
- Power estimate for 1500 km and 20 Gbaud: $2 \times 15 \times 0.18$ W $= 5.4$ W
- Quadratic scaling: $\approx 25 \times 5.4$ W $= 135$ W (full DBP)
- Linear scaling: $\approx 5 \times 5.4$ W $= 27$ W (5 independent receivers)

Question

Is it possible to scale the time-domain / deep learning approach gracefully to larger bandwidths?

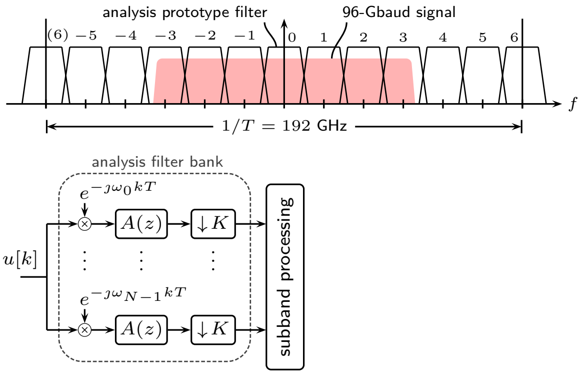
Subband Processing via Filter Banks

See, e.g., [Taylor, 2008], [Ho, 2009], [Slim et al., 2013], [Nazarathy and Tolmachev, 2014] (linear comp.) and [Mateo et al., 2010], [Ip et al., 2011], [Oyama et al., 2015] (nonlinear comp.)



Subband Processing via Filter Banks

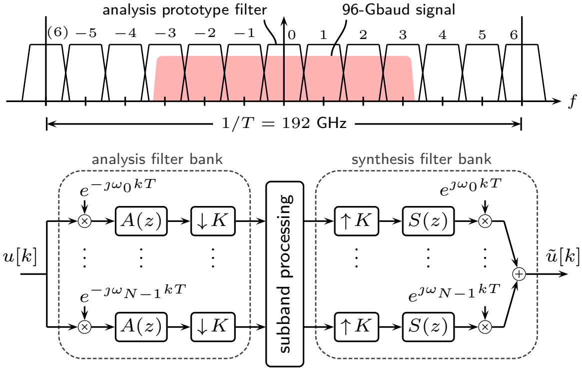
See, e.g., [Taylor, 2008], [Ho, 2009], [Slim et al., 2013], [Nazarathy and Tolmachev, 2014] (linear comp.) and [Mateo et al., 2010], [Ip et al., 2011], [Oyama et al., 2015] (nonlinear comp.)



- Split received signal into N parallel signals, then downsample by K

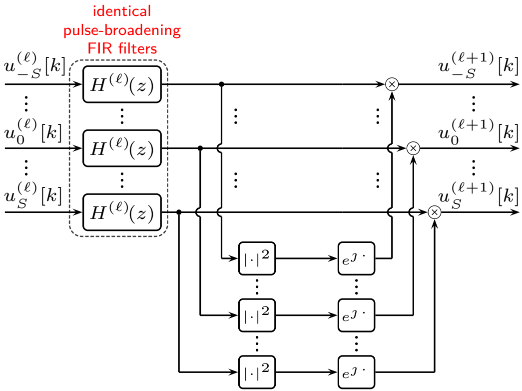
Subband Processing via Filter Banks

See, e.g., [Taylor, 2008], [Ho, 2009], [Slim et al., 2013], [Nazarathy and Tolmachev, 2014] (linear comp.) and [Mateo et al., 2010], [Ip et al., 2011], [Oyama et al., 2015] (nonlinear comp.)

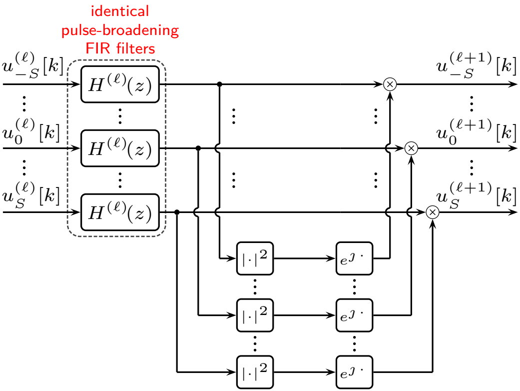


- Split received signal into N parallel signals, then downsample by K
- Synthesis filter bank reassembles the signal after processing

Proposed DSP Architecture

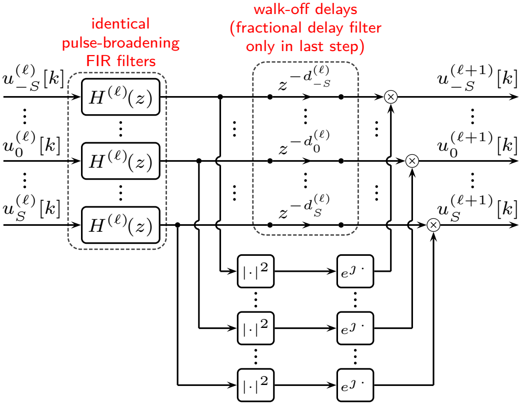


Proposed DSP Architecture



subband group delay differences depend linearly on propagation distance
$\Longrightarrow$ choose step size such that walk-off can be compensated with delay elements

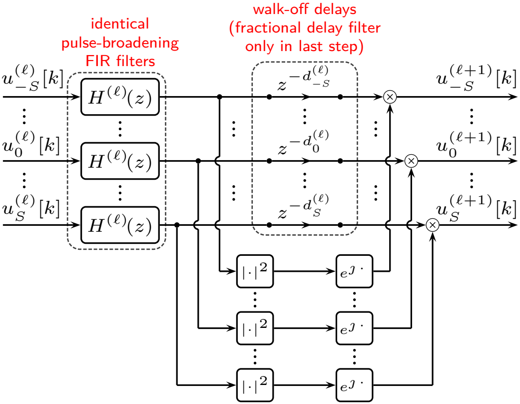
Proposed DSP Architecture



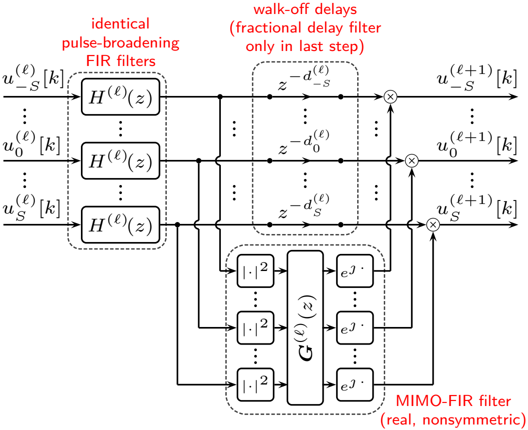
subband group delay differences depend linearly on propagation distance

$\Longrightarrow$ choose step size such that walk-off can be compensated with delay elements

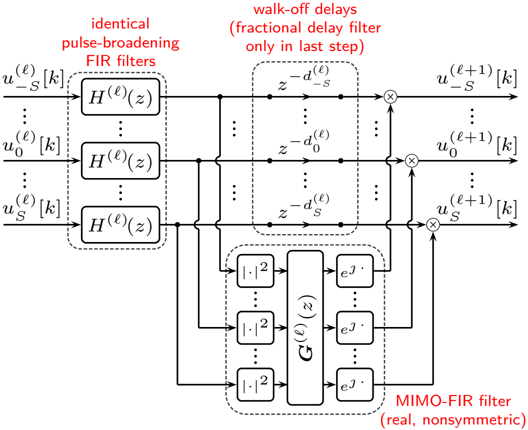
Proposed DSP Architecture



Proposed DSP Architecture

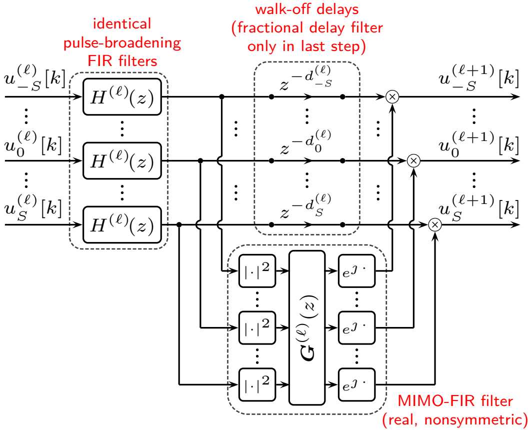


Proposed DSP Architecture



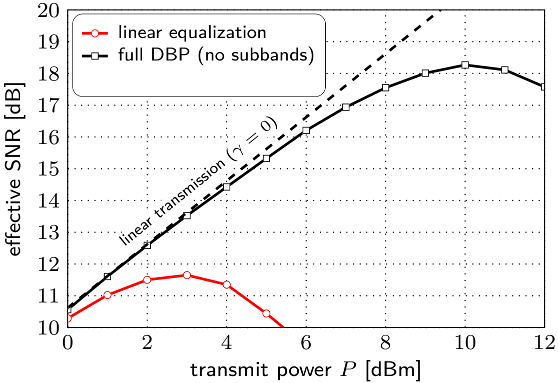
- Hardware-efficient implementation (no FFT/IFFT) of split-step method for coupled NLSEs [Leibrich and Rosenkranz, 2003], see also [Mateo et al., 2010]
- Only accounts for XPM between subbands, but not FWM

Proposed DSP Architecture



- Hardware-efficient implementation (no FFT/IFFT) of split-step method for coupled NLSEs [Leibrich and Rosenkranz, 2003], see also [Mateo et al., 2010]
- Only accounts for XPM between subbands, but not FWM
- Deep learning to optimize all filters $H^{(\ell)}(z)$, $\boldsymbol{G}^{(\ell)}(z)$ for $\ell = 1, \ldots, M$

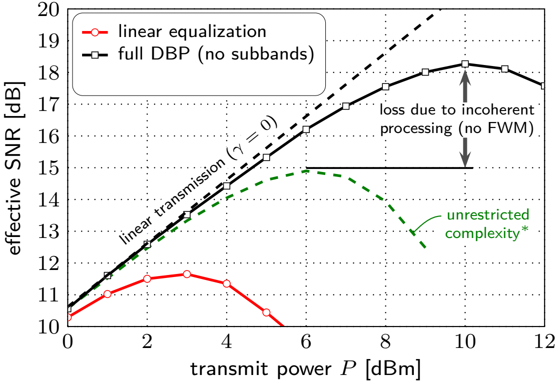
Results ($N = 12$ Subbands for 96 Gbaud)



System parameters:

- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

Results ($N = 12$ Subbands for 96 Gbaud)

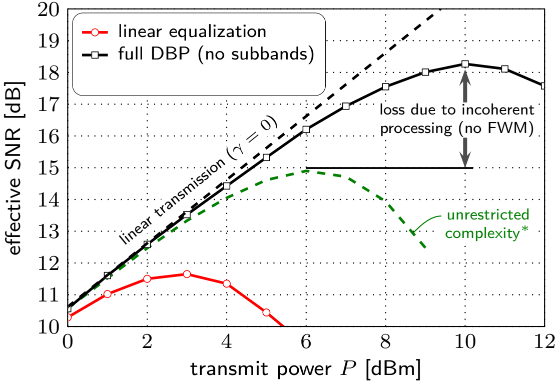


System parameters:

- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

* [Leibrich and Rosenkranz, 2003] FFT/IFFT, no downsampling, 1000 steps/span

Results ($N = 12$ Subbands for 96 Gbaud)



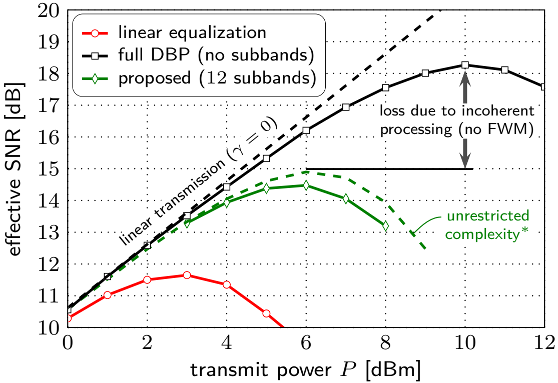
System parameters:

- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

* [Leibrich and Rosenkranz, 2003] FFT/IFFT, no downsampling, 1000 steps/span

- $K = 8$ (1.5 subband oversampling), 38.2 km step size (≈ 2.6 steps/span)

Results ($N = 12$ Subbands for 96 Gbaud)



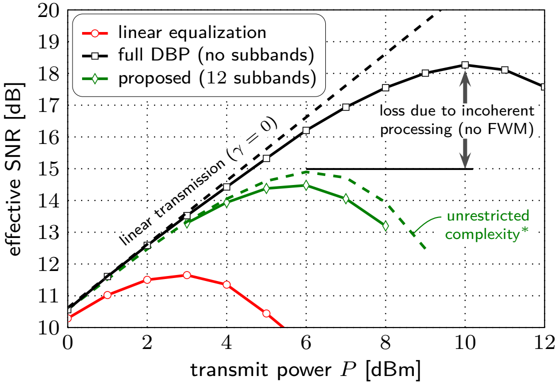
System parameters:

- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

* [Leibrich and Rosenkranz, 2003]
FFT/IFFT, no downsampling,
1000 steps/span

- $K = 8$ (1.5 subband oversampling), 38.2 km step size (≈ 2.6 steps/span)

Results ($N = 12$ Subbands for 96 Gbaud)



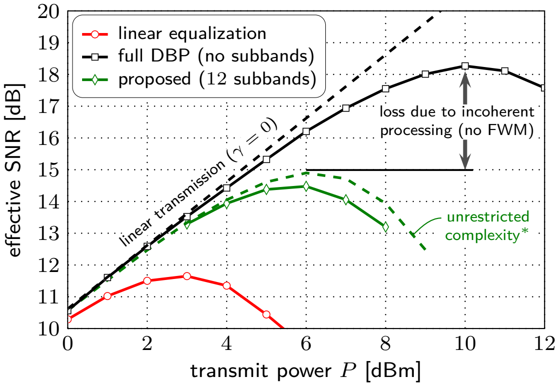
System parameters:

- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

* [Leibrich and Rosenkranz, 2003] FFT/IFFT, no downsampling, 1000 steps/span

- $K = 8$ (1.5 subband oversampling), 38.2 km step size (≈ 2.6 steps/span)
- 7-tap learned filters (16 real multpl.), sparse MIMO filters (8 real multpl.)

Results ($N = 12$ Subbands for 96 Gbaud)



System parameters:

- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

* [Leibrich and Rosenkranz, 2003] FFT/IFFT, no downsampling, 1000 steps/span

- $K = 8$ (1.5 subband oversampling), 38.2 km step size ($\approx$ 2.6 steps/span)
- 7-tap learned filters (16 real multpl.), sparse MIMO filters (8 real multpl.)
- $> 4\times$ less real multipl. compared to FFT/IFFT [Mateo et al., 2010]
- $\approx 2 - 3\times$ less complexity compared to full DBP (estimated)

Conclusions and Future Work

Conclusions and Future Work

- Split-step digital backpropagation appears feasible for real-time DSP implementation using a time-domain approach for the linear steps
- Deep learning can be used to
  - jointly optimize all chromatic dispersion filters
  - prune filter taps to get very short filters
  - jointly quantize all filter coefficients
- Wideband compensation can be efficient using subband processing

Conclusions and Future Work

- Split-step digital backpropagation appears feasible for real-time DSP implementation using a time-domain approach for the linear steps
- Deep learning can be used to
  - jointly optimize all chromatic dispersion filters
  - prune filter taps to get very short filters
  - jointly quantize all filter coefficients
- Wideband compensation can be efficient using subband processing

- ASIC implementation of subband procesing & experimental demonstration
- Can we do efficient FWM compensation with subbands?

Conclusions and Future Work

- Split-step digital backpropagation appears feasible for real-time DSP implementation using a time-domain approach for the linear steps
- Deep learning can be used to
  - jointly optimize all chromatic dispersion filters
  - prune filter taps to get very short filters
  - jointly quantize all filter coefficients
- Wideband compensation can be efficient using subband processing

- ASIC implementation of subband procesing & experimental demonstration
- Can we do efficient FWM compensation with subbands?

Thank you!

FORCE FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

Duke UNIVERSITY

References I

Crivelli, D. E., Hueda, M. R., Carrer, H. S., Del Barco, M., López, R. R., Gianni, P., Finochietto, J., Swenson, N., Voois, P., and Agazzi, O. E. (2014).
Architecture of a single-chip 50 Gb/s DP-QPSK/BPSK transceiver with electronic dispersion compensation for coherent optical channels.
IEEE Trans. Circuits Syst. I: Reg. Papers, 61(4):1012–1025.

Du, L. B. and Lowery, A. J. (2010).
Improved single channel backpropagation for intra-channel fiber nonlinearity compensation in long-haul optical communication systems.
Opt. Express, 18(16):17075–17088.

Essiambre, R.-J. and Winzer, P. J. (2005).
Fibre nonlinearities in electronically pre-distorted transmission.
In *Proc. European Conf. Optical Communication (ECOC)*, Glasgow, UK.

Fougstedt, C., Mazur, M., Svensson, L., Eliasson, H., Karlsson, M., and Larsson-Edefors, P. (2017a).
Time-domain digital back propagation: Algorithm and finite-precision implementation aspects.
In *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA.

Fougstedt, C., Svensson, L., Mazur, M., Karlsson, M., and Larsson-Edefors, P. (2017b).
Finite-precision optimization of time-domain digital back propagation by inter-symbol interference minimization.
In *Proc. European Conf. Optical Communication*, Gothenburg, Sweden.

Giacoumidis, E., Le, S. T., Ghanbarisabagh, M., McCarthy, M., Aldaya, I., Mhatli, S., Jarajreh, M. A., Haigh, P. A., Doran, N. J., Ellis, A. D., and Eggleton, B. J. (2015).
Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based nonlinear equalization.
Opt. Lett., 40(21):5113–5116.

References II

Goldfarb, G. and Li, G. (2009).
Efficient backward-propagation using wavelet- based filtering for fiber backward-propagation.
Opt. Express, 17(11):814–816.

Häger, C. and Pfister, H. D. (2018a).
Deep learning of the nonlinear Schrödinger equation in fiber-optic communications.
In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Vail, CO.

Häger, C. and Pfister, H. D. (2018b).
Nonlinear interference mitigation via deep neural networks.
In *Proc. Optical Fiber Communication Conf. (OFC)*, San Diego, CA.

Ho, K.-P. (2009).
Subband equaliser for chromatic dispersion of optical fibre.
Electronics Lett., 45(24):1224–1226.

Ip, E., Bai, N., and Wang, T. (2011).
Complexity versus performance tradeoff for fiber nonlinearity compensation using frequency-shaped, multi-subband backpropagation.
In *Proc. Optical Fiber Communication Conf. (OFC)*, page OThF4, Los Angeles, CA.

Ip, E. and Kahn, J. M. (2008).
Compensation of dispersion and nonlinear impairments using digital backpropagation.
J. Lightw. Technol., 26:3416–3425.

Jarajreh, A. M., Giacoumidis, E., Aldaya, I., Le, S. T., Tsokanos, A., Ghassemlooy, Z., and Doran, N. J. (2015).
Artificial neural network nonlinear equalizer for coherent optical OFDM.
IEEE Photon. Technol. Lett., 27(4):387–390.

References III

Jones, R. T., Eriksson, T. A., Yankov, M. P., and Zibar, D. (2018).
Deep learning of geometric constellation shaping including fiber nonlinearities.
(1):3–5.

Karanov, B., Chagnon, M., Thouin, F., Eriksson, T. A., Bülow, H., Lavery, D., Bayvel, P., and Schmalen, L. (2018).
End-to-end deep learning of optical fiber communications.

Khan, F. N., Zhou, Y., Lau, A. P. T., and Lu, C. (2012).
Modulation format identification in heterogeneous fiber-optic networks using artificial neural networks.
Opt. Express, 20(11):12422.

Lee, H., Lee, I., and Lee, S. H. (2018).
Deep learning based transceiver design for multi-colored vlc systems.
Opt. Express, 26(5):6222–6238.

Leibrich, J. and Rosenkranz, W. (2003).
Efficient numerical simulation of multichannel WDM transmission systems limited by XPM.
IEEE Photon. Technol. Lett., 15(3):395–397.

Li, L., Tao, Z., Dou, L., Yan, W., Oda, S., Tanimura, T., Hoshida, T., and Rasmussen, J. C. (2011).
Implementation efficient nonlinear equalizer based on correlated digital backpropagation.
In *Proc. Optical Fiber Communication Conf. (OFC)*, page OWW3, Los Angeles, CA.

Li, S., Häger, C., Garcia, N., and Wymeersch, H. (2018).
Achievable information rates for nonlinear fiber communication via end-to-end autoencoder learning.
In *Proc. European Conf. Optical Communication (ECOC)*, Rome, Italy.

References IV

Li, X., Chen, X., Goldfarb, G., Mateo, E., Kim, I., Yaman, F., and Li, G. (2008).
Electronic post-compensation of WDM transmission impairments using coherent detection and digital signal processing.
Opt. Express, 16(2):880–888.

Martins, C. S., Bertignono, L., Nespola, A., Carena, A., Guiomar, F. P., and Pinto, A. N. (2018).
Efficient time-domain DBP using random step-size and multi-band quantization.
In *Proc. Optical Fiber Communication Conf. (OFC)*, San Diego, CA.

Mateo, E. F., Yaman, F., and Li, G. (2010).
Efficient compensation of inter-channel nonlinear effects via digital backward propagation in WDM optical transmission.
Opt. Express, 18(14):15144.

Napoli, A., Maalej, Z., Sleiffer, V. A. J. M., Kuschnerov, M., Rafique, D., Timmers, E., Spinnler, B., Rahman, T., Coelho, L. D., and Hanik, N. (2014).
Reduced complexity digital back-propagation methods for optical communication systems.
J. Lightw. Technol., 32(7).

Nazarathy, M. and Tolmachev, A. (2014).
Subbanded DSP architectures based on underdecimated filter banks for coherent OFDM receivers: Overview and recent advances.
IEEE Signal Processing Mag., 31(2):70–81.

Oyama, T., Nakashima, H., Hoshida, T., Tanimura, T., Akiyama, Y., Tao, Z., and Rasmussen, J. C. (2015).
Complexity reduction of perturbation-based nonlinear compensator by sub-band processing.
In *Proc. Optical Fiber Communication Conf. (OFC)*.

References V

Paré, C., Villeneuve, A., Bélanger, P.-A. A., and Doran, N. J. (1996).
Compensating for dispersion and the nonlinear Kerr effect without phase conjugation.
Optics Letters, 21(7):459–461.

Pillai, B. S. G., Sedighi, B., Guan, K., Anthapadmanabhan, N. P., Shieh, W., Hinton, K. J., and Tucker, R. S. (2014).
End-to-end energy modeling and analysis of long-haul coherent transmission systems.
J. Lightw. Technol., 32(18):3093–3111.

Rafique, D., Zhao, J., and Ellis, A. D. (2011).
Digital back-propagation for spectrally efficient wdm 112 gbit/s pm m-ary qam transmission.
Opt. Express, 19(6):5219–5224.

Roberts, K., Li, C., Strawczynski, L., O'Sullivan, M., and Hardcastle, I. (2006).
Electronic precompensation of optical nonlinearity.
IEEE Photon. Technol. Lett., 18(2):403–405.

Secondini, M., Rommel, S., Meloni, G., Fresi, F., Forestieri, E., and Poti, L. (2016).
Single-step digital backpropagation for nonlinearity mitigation.
Photon. Netw. Commun., 31(3):493–502.

Sheikh, A., Fougstedt, C., Graell i Amat, A., Johannisson, P., Larsson-Edefors, P., and Karlsson, M. (2016).
Dispersion compensation FIR filter with improved robustness to coefficient quantization errors.
J. Lightw. Technol., 34(22):5110–5117.

Shen, T. S. R. and Lau, A. P. T. (2011).
Fiber nonlinearity compensation using extreme learning machine for DSP-based coherent communication systems.
In *Proc. Optoelectronics and Communications Conf. (OECC)*, Kaohsiung, Taiwan.

References VI

Sherborne, T., Banks, B., Semrau, D., Killey, R. I., Bayvel, P., and Lavery, D. (2018).
On the impact of fixed point hardware for optical fiber nonlinearity compensation algorithms.
arXiv:1804.08545 [eess.SP].

Slim, I., Mezghani, A., Baltar, L. G., Qi, J., Hauske, F. N., and Nossek, J. A. (2013).
Delayed single-tap frequency-domain chromatic-dispersion compensation.
IEEE Photon. Technol. Lett., 25(2):167–170.

Tanimura, T., Hoshida, T., Hoshida, T., Kato, T., Watanabe, S., Rasmussen, J. C., Suzuki, M., and Morikawa, H. (2016).
Deep learning based OSNR monitoring independent of modulation format, symbol rate and chromatic dispersion.
In *Proc. European Conf. Optical Communication (ECOC)*, Düsseldorf, Germany.

Taylor, M. G. (2008).
Compact digital dispersion compensation algorithms.
In *Proc. Optical Fiber Communication Conf. (OFC)*, San Diego, CA.

Xiaoxia, W., Jargon, J., Skoog, R., Paraschis, L., and Willner, A. (2009).
Applications of artificial neural networks in optical performance monitoring.
J. Lightw. Technol., 27(16):3580–3589.

Yan, W., Tao, Z., Dou, L., Li, L., Oda, S., Tanimura, T., Hoshida, T., and Rasmussen, J. C. (2011).
Low complexity digital perturbation back-propagation.
In *Proc. European Conf. Optical Communication (ECOC)*, page Tu.3.A.2, Geneva, Switzerland.

Zhu, L., Li, X., Mateo, E., and Li, G. (2009).
Complementary FIR filter pair for distributed impairment compensation of WDM fiber transmission.
IEEE Photon. Technol. Lett., 21(5):292–294.

References VII

Zibar, D., Piels, M., Jones, R., and Schaeffer, C. G. (2016).
Machine learning techniques in optical communication.
J. Lightw. Technol., 34(6):1442–1452.