Physics-Based Machine Learning for Fiber-Optic Communication Systems

Christian Häger

Department of Electrical Engineering, Chalmers University of Technology, Sweden

Workshop on Machine Learning and Optical Systems
(Boston Chapter of the IEEE Photonics Society)
October 28, 2020

FORCE
FIBER-OPTIC COMMUNICATIONS
RESEARCH CENTER

AVANCEZ 1829
CHALMERS

Thank You!

Henry D. Pfister
Duke

Christoffer Fougstedt
Chalmers (now: Ericsson)

Lars Svensson
Chalmers

Per Larsson-Edefors
Chalmers

Rick M. Bütler
TU/e (now: TU Delft)

Gabriele Liga
TU/e

Alex Alvarado
TU/e

Vinícius Oliari
TU/e

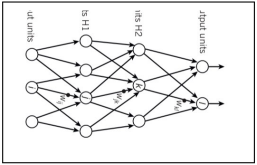
Sebastiaan Goossens
TU/e

Menno van den Hout
TU/e

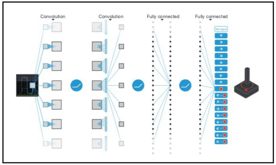
Sjoerd van der Heide
TU/e

Chigo Okonkwo
TU/e

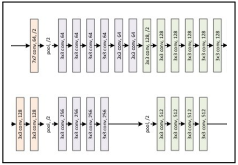
Deep Learning [LeCun et al., 2015]



Deep Q-Learning [Mnih et al., 2015]



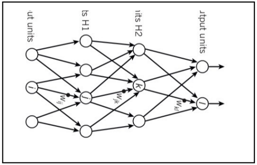
ResNet [He et al., 2015]



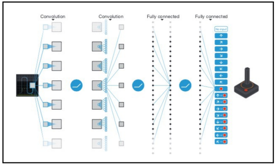
...

Multi-layer neural networks: impressive performance, countless applications

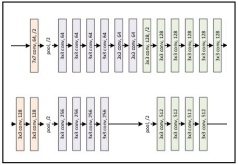
Deep Learning [LeCun et al., 2015]



Deep Q-Learning [Mnih et al., 2015]

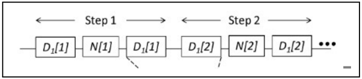


ResNet [He et al., 2015]

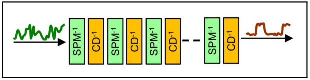


. . .

Multi-layer neural networks: impressive performance, countless applications



[Du and Lowery, 2010]



[Nakashima et al., 2017]

Split-step methods for solving the propagation equation in fiber-optics

Agenda

In this talk, we . . .

Agenda

In this talk, we . . .

1. show that multi-layer neural networks and the split-step method have the same functional form: both alternate linear and pointwise nonlinear steps

Agenda

In this talk, we ...

1. show that multi-layer neural networks and the split-step method have the same functional form: both alternate linear and pointwise nonlinear steps
2. propose a physics-based machine-learning approach based on parameterizing the split-step method (no black-box neural networks)

Agenda

In this talk, we . . .

1. show that multi-layer neural networks and the split-step method have the same functional form: both alternate linear and pointwise nonlinear steps
2. propose a physics-based machine-learning approach based on parameterizing the split-step method (no black-box neural networks)
3. revisit hardware-efficient nonlinear equalization via learned digital backpropagation

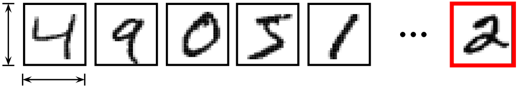
Outline

Outline

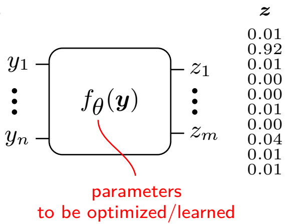
1. Machine Learning and Neural Networks for Communications
2. Physics-Based Machine Learning for Fiber-Optic Communications
3. Learned Digital Backpropagation
4. Polarization-Dependent Effects
5. Wideband Signals
6. Conclusions

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

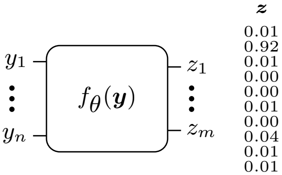


28×28 pixels $\implies n = 784$

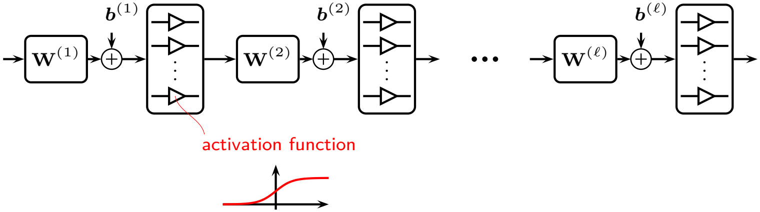


Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

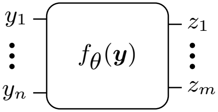


How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



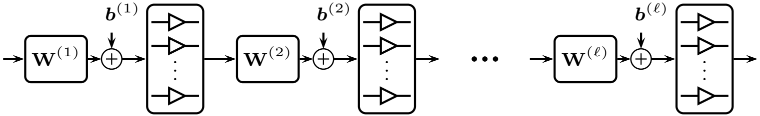
Supervised Learning

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

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

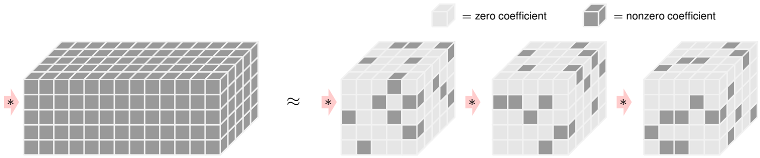
mean squared error, cross-entropy, ...

stochastic gradient descent, RMSProp, Adam, ...

Why Deep Models?

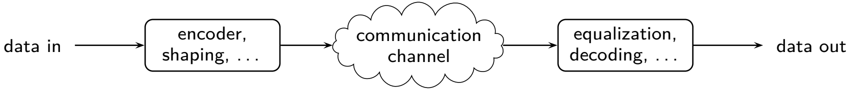
Many possible answers

One advantage is complexity: deep computation graphs tend to be more parameter efficient than shallow graphs [Lin et al., 2017]

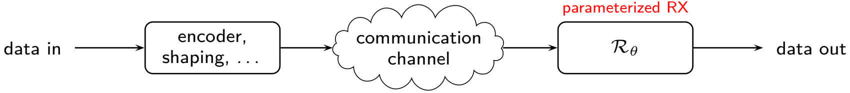


- Sparsity can emerge due to (approximate) factorization (even for linear models, e.g., FFT)
- Deep computation graphs allow for very simple elementary steps
- Deep models typically have many "good" parameter configurations that are close to each other $\Longrightarrow$ robustness to, e.g., quantization noise

Machine Learning for Physical-Layer Communications

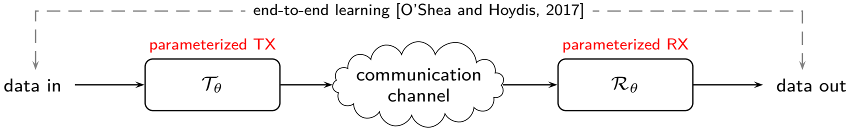


Machine Learning for Physical-Layer Communications



[Shen and Lau, 2011], Fiber nonlinearity compensation using extreme learning machine for DSP-based . . . , (*OECC*)
[Giacoumidis et al., 2015], Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based . . . , (*Opt. Lett.*)
[Zibar et al., 2016], Machine learning techniques in optical communication, (*J. Lightw. Technol.*)
[Kamalov et al., 2018], Evolution from 8qam live traffic to ps 64-qam with neural-network based nonlinearity compensation . . . , (*OFC*)
. . .

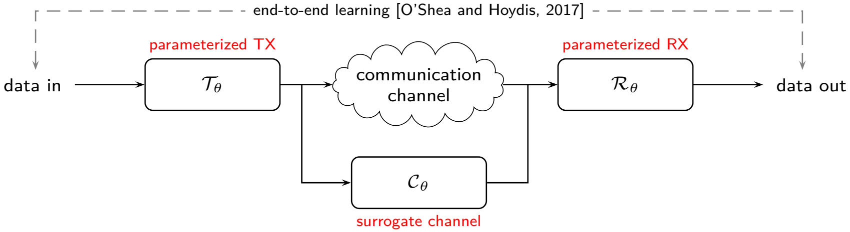
Machine Learning for Physical-Layer Communications



[Shen and Lau, 2011], Fiber nonlinearity compensation using extreme learning machine for DSP-based . . . , (*OECC*)
[Giacoumidis et al., 2015], Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based . . . , (*Opt. Lett.*)
[Zibar et al., 2016], Machine learning techniques in optical communication, (*J. Lightw. Technol.*)
[Kamalov et al., 2018], Evolution from 8qam live traffic to ps 64-qam with neural-network based nonlinearity compensation . . . , (*OFC*)
. . .

[O'Shea and Hoydis, 2017], An introduction to deep learning for the physical layer, (*IEEE Trans. Cogn. Commun. Netw.*)
[Karanov et al., 2018], End-to-end deep learning of optical fiber communications (*J. Lightw. Technol.*)
[Jones et al., 2018], Deep learning of geometric constellation shaping including fiber nonlinearities, (*ECOC*)
[Li et al., 2018], Achievable information rates for nonlinear fiber communication via end-to-end autoencoder learning, (*ECOC*)
. . .

Machine Learning for Physical-Layer Communications

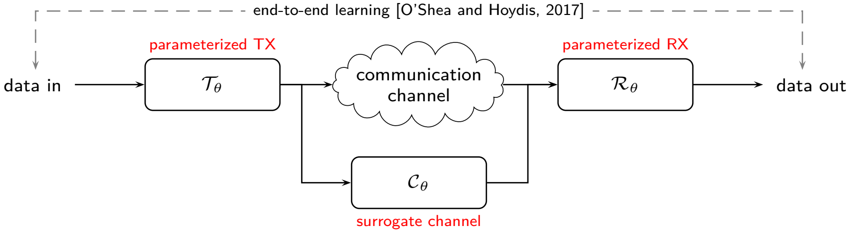


[Shen and Lau, 2011], Fiber nonlinearity compensation using extreme learning machine for DSP-based . . . , (*OECC*)
[Giacoumidis et al., 2015], Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based . . . , (*Opt. Lett.*)
[Zibar et al., 2016], Machine learning techniques in optical communication, (*J. Lightw. Technol.*)
[Kamalov et al., 2018], Evolution from 8qam live traffic to ps 64-qam with neural-network based nonlinearity compensation . . . , (*OFC*)
. . .

[O'Shea and Hoydis, 2017], An introduction to deep learning for the physical layer, (*IEEE Trans. Cogn. Commun. Netw.*)
[Karanov et al., 2018], End-to-end deep learning of optical fiber communications (*J. Lightw. Technol.*)
[Jones et al., 2018], Deep learning of geometric constellation shaping including fiber nonlinearities, (*ECOC*)
[Li et al., 2018], Achievable information rates for nonlinear fiber communication via end-to-end autoencoder learning, (*ECOC*)
. . .

[O'Shea et al., 2018], Approximating the void: Learning stochastic channel models from observation with variational GANs, (*arXiv*)
[Ye et al., 2018], Channel agnostic end-to-end learning based communication systems with conditional GAN, (*arXiv*)
. . .

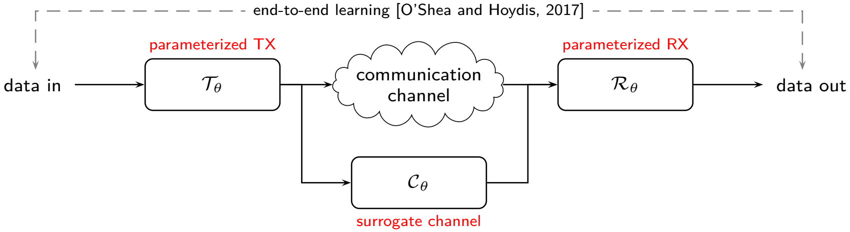
Machine Learning for Physical-Layer Communications



Using (deep) neural networks for $\mathcal{T}_\theta, \mathcal{R}_\theta, \mathcal{C}_\theta$

- How to choose network architecture (#layers, activation function)?
- How to initialize parameters?
- How to interpret solutions? Any insight gained?
- . . .

Machine Learning for Physical-Layer Communications



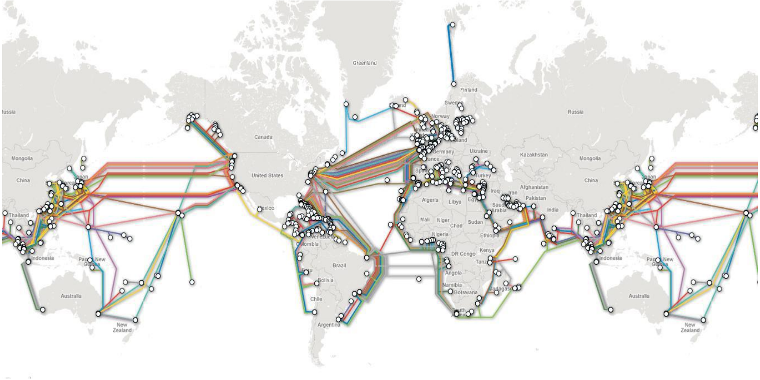
Using (deep) neural networks for $\mathcal{T}_\theta, \mathcal{R}_\theta, \mathcal{C}_\theta$

- How to choose network architecture (#layers, activation function)? ✗
- How to initialize parameters? ✗
- How to interpret solutions? Any insight gained? ✗
- . . .

Model-based learning: sparse signal recovery [Gregor and Lecun, 2010], [Borgerding and Schniter, 2016], neural belief propagation [Nachmani et al., 2016], radio transformer networks [O'Shea and Hoydis, 2017], . . .

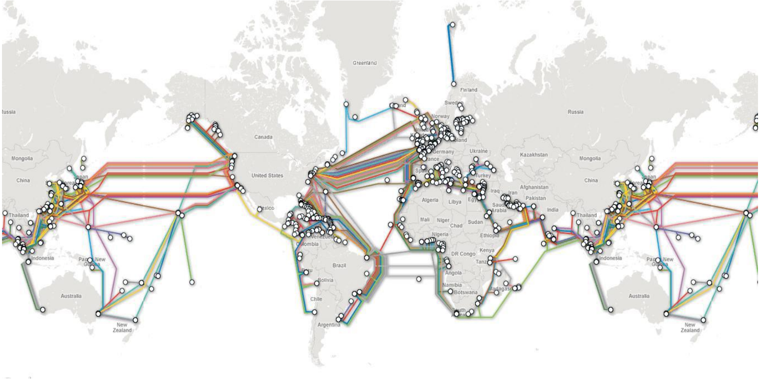
Outline

Fiber-Optic Communications



Fiber-optic systems enable data traffic over very long distances connecting cities, countries, and continents.

Fiber-Optic Communications

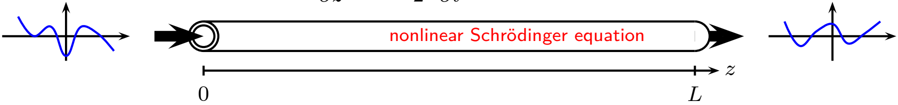


Fiber-optic systems enable data traffic over very long distances connecting cities, countries, and continents.

- Dispersion: different wavelengths travel at different speeds (linear)
- Kerr effect: refractive index changes with signal intensity (nonlinear)

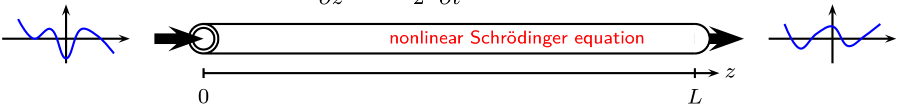
Channel Modeling

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



Channel Modeling

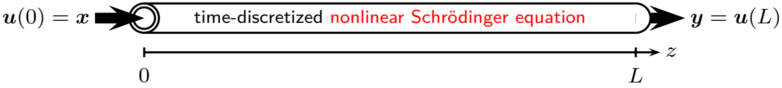
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$

Channel Modeling

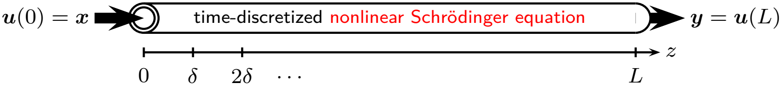
$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \mathbf{A}\boldsymbol{u}(z) + \jmath\gamma\boldsymbol{\rho}(\boldsymbol{u}(z))$$



- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$

Channel Modeling

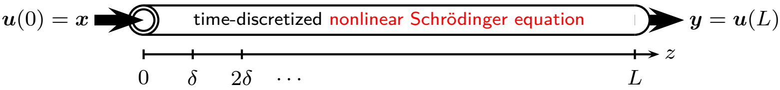
$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \mathbf{A}\boldsymbol{u}(z) + \jmath\gamma\boldsymbol{\rho}(\boldsymbol{u}(z))$$



- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$
- Split-step method with M steps ($\delta = L/M$):

Channel Modeling

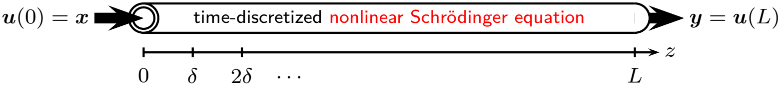
$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \mathbf{A}\boldsymbol{u}(z)$$



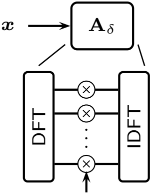
- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$
- Split-step method with M steps ($\delta = L/M$):

Channel Modeling

$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \mathbf{A}\boldsymbol{u}(z)$$



- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$
- Split-step method with M steps ($\delta = L/M$):

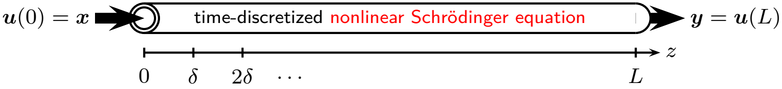


$H_k = e^{\jmath \frac{\beta_2}{2} \delta \omega_k^2}$ group velocity dispersion (all-pass filter)

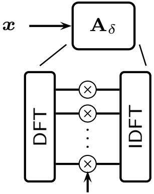
Channel Modeling

$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \qquad + \jmath\gamma\boldsymbol{\rho}(\boldsymbol{u}(z))$$

$\rho(x) = |x|^2 x$ element-wise



- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$
- Split-step method with M steps ($\delta = L/M$):

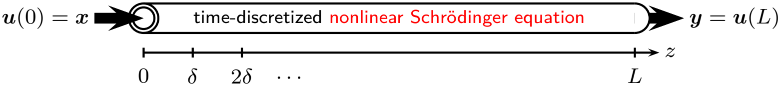


$H_k = e^{\jmath \frac{\beta_2}{2} \delta \omega_k^2}$ group velocity dispersion (all-pass filter)

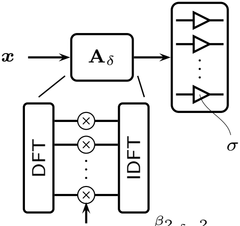
Channel Modeling

$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \quad + \jmath\gamma\boldsymbol{\rho}(\boldsymbol{u}(z))$$

$\rho(x) = |x|^2 x$ element-wise



- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$
- Split-step method with M steps ($\delta = L/M$):



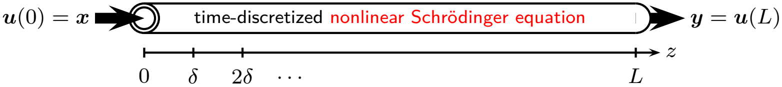
$\sigma_\delta(x) = x e^{\jmath\gamma\delta|x|^2}$ Kerr effect

$H_k = e^{\jmath\frac{\beta_2}{2}\delta\omega_k^2}$ group velocity dispersion (all-pass filter)

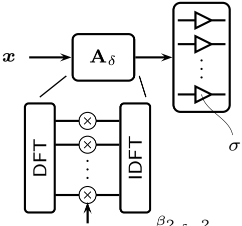
Channel Modeling

$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \mathbf{A}\boldsymbol{u}(z) + \jmath\gamma\boldsymbol{\rho}(\boldsymbol{u}(z))$$

$\rho(x) = |x|^2 x$ element-wise



- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$
- Split-step method with M steps ($\delta = L/M$):



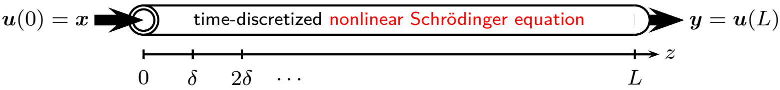
$\sigma_\delta(x) = xe^{\jmath\gamma\delta|x|^2}$ Kerr effect

$H_k = e^{\jmath\frac{\beta_2}{2}\delta\omega_k^2}$ group velocity dispersion (all-pass filter)

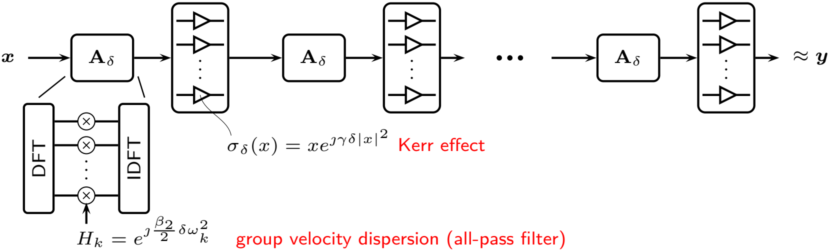
Channel Modeling

$$\frac{\mathrm{d}\boldsymbol{u}(z)}{\mathrm{d}z} = \mathbf{A}\boldsymbol{u}(z) + \jmath\gamma\boldsymbol{\rho}(\boldsymbol{u}(z))$$

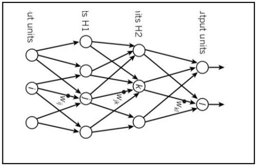
$\rho(x) = |x|^2 x$ element-wise



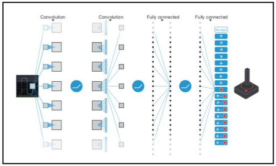
- Sampling over a fixed time interval $\implies \mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$
- Split-step method with M steps ($\delta = L/M$):



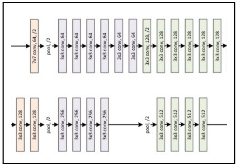
Deep Learning [LeCun et al., 2015]



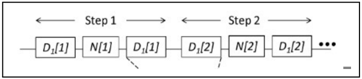
Deep Q-Learning [Mnih et al., 2015]



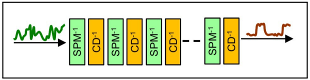
ResNet [He et al., 2015]



. . .



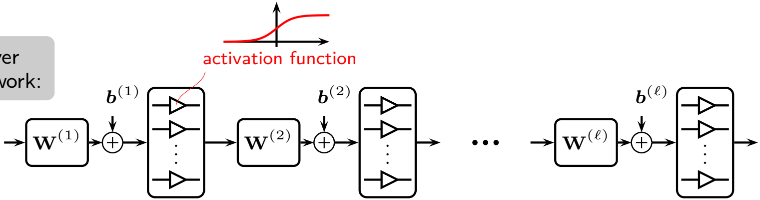
[Du and Lowery, 2010]



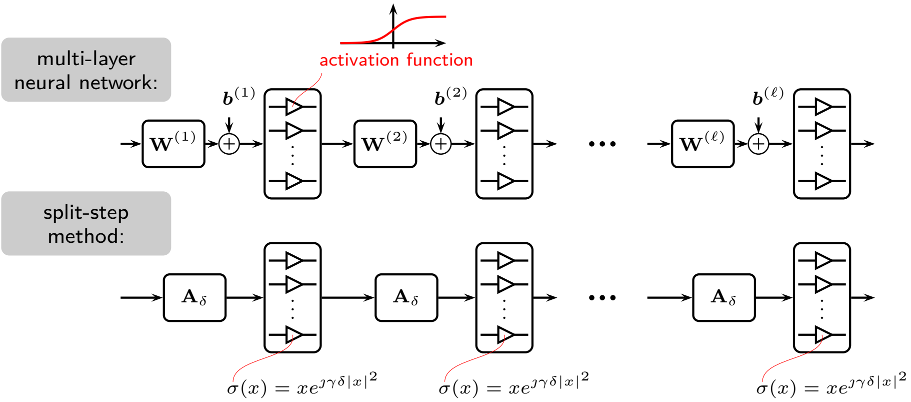
[Nakashima et al., 2017]

Parameterizing the Split-Step Method

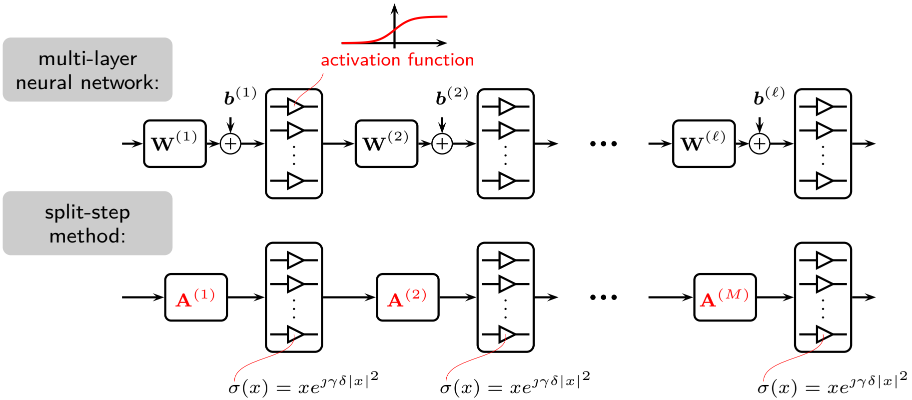
multi-layer neural network:



Parameterizing the Split-Step Method

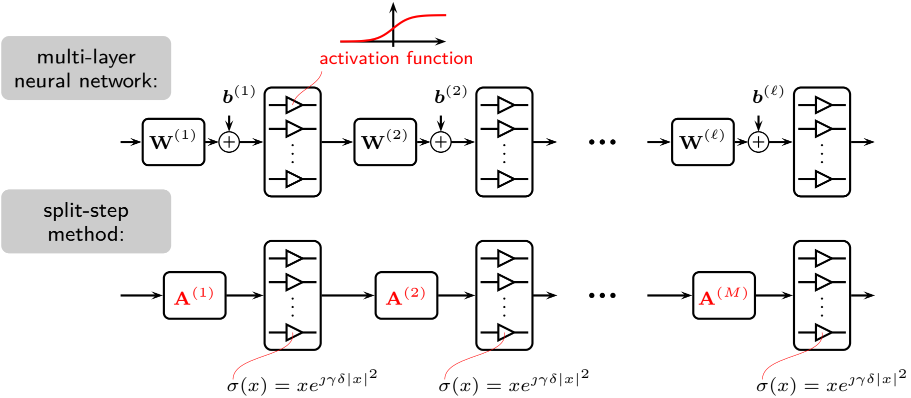


Parameterizing the Split-Step Method



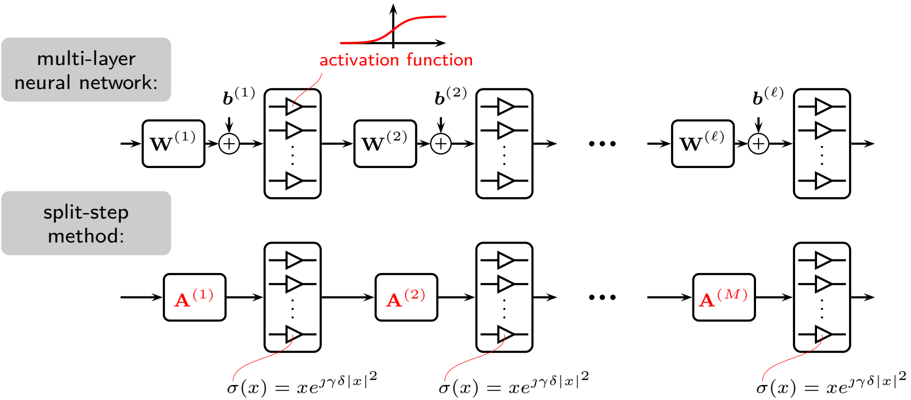
[Häger & Pfister, 2018], Nonlinear Interference Mitigation via Deep Neural Networks, *(OFC)*
[Häger & Pfister, 2018], Deep Learning of the Nonlinear Schrödinger Equation in Fiber-Optic Communications, *(ISIT)*

Parameterizing the Split-Step Method



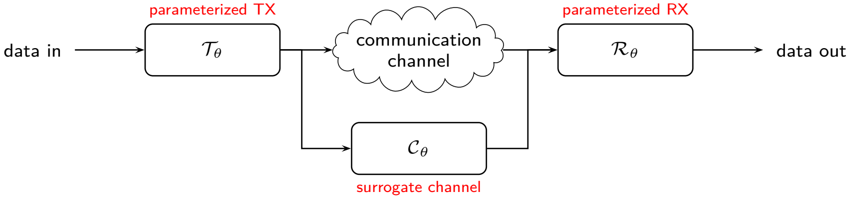
- Parameterized model f_θ with $\theta = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(M)}\}$

Parameterizing the Split-Step Method

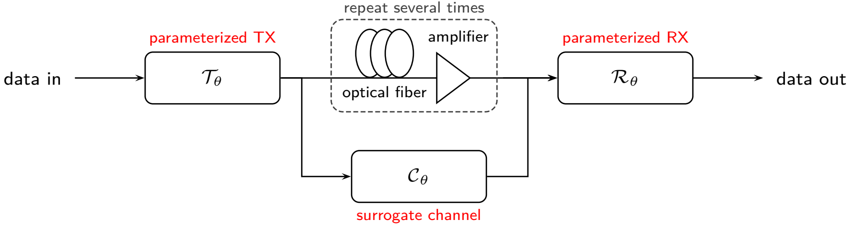


- Parameterized model f_θ with $\theta = \{\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(M)}\}$
- Includes as special cases: step-size optimization, "placement" of nonlinear operator, higher-order dispersion, matched filtering ...

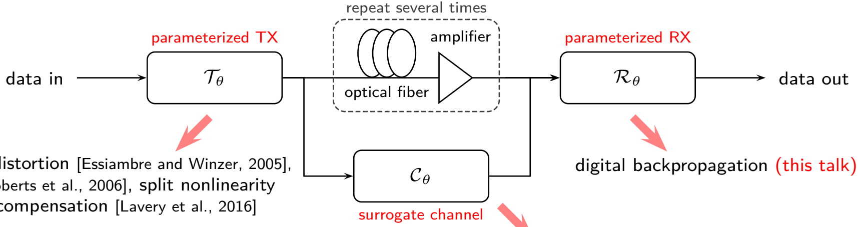
Possible Applications



Possible Applications



Possible Applications

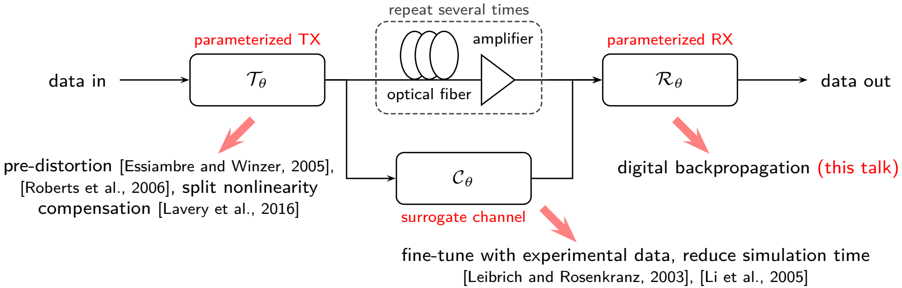


pre-distortion [Essiambre and Winzer, 2005], [Roberts et al., 2006], split nonlinearity compensation [Lavery et al., 2016]

digital backpropagation (this talk)

fine-tune with experimental data, reduce simulation time [Leibrich and Rosenkranz, 2003], [Li et al., 2005]

Possible Applications

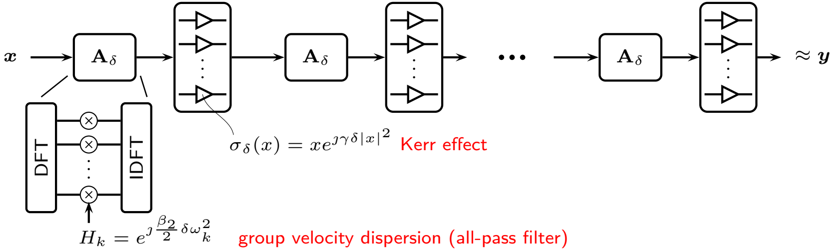


Physics/model-based learning approaches

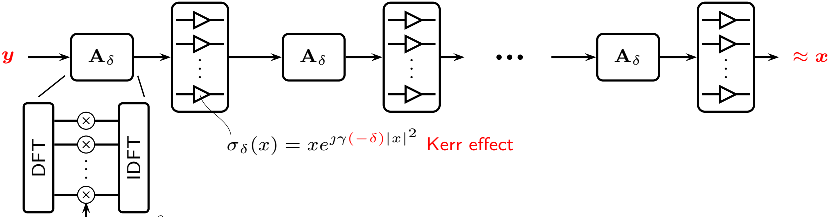
- How to choose network architecture (#layers, activation function)? ✓
- How to initialize parameters? ✓
- How to interpret solutions? Any insight gained? ✓

Outline

Digital Backpropagation



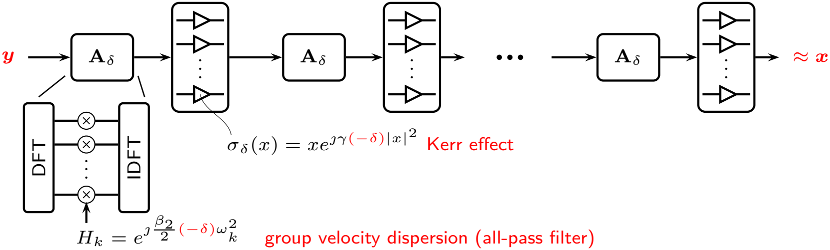
Digital Backpropagation



$\sigma_\delta(x) = x e^{\jmath \gamma (-\delta) |x|^2}$ Kerr effect

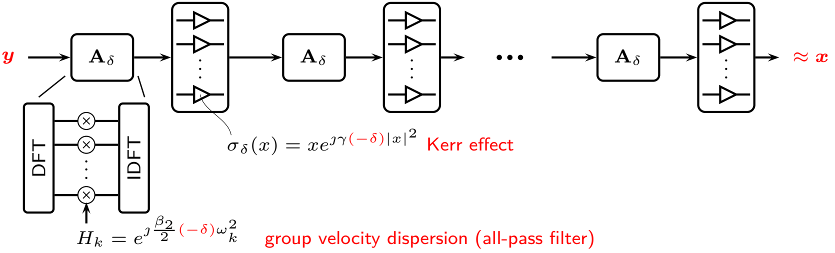
$H_k = e^{\jmath \frac{\beta_2}{2} (-\delta) \omega_k^2}$ group velocity dispersion (all-pass filter)

Digital Backpropagation



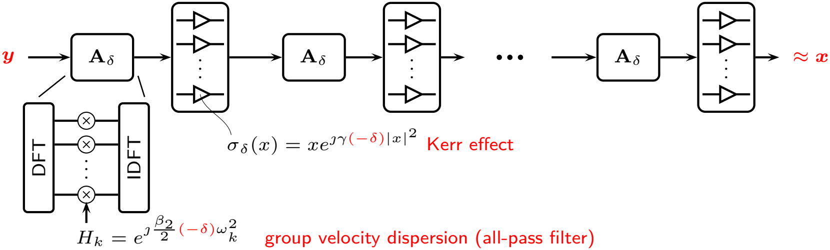
- Fiber with negated parameters ($\beta_2 \to -\beta_2$, $\gamma \to -\gamma$) would perform perfect channel inversion [Paré et al., 1996] (ignoring attenuation)

Digital Backpropagation



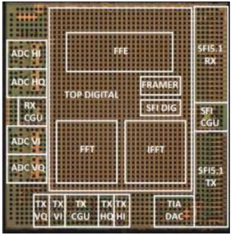
- Fiber with negated parameters ($\beta_2 \to -\beta_2$, $\gamma \to -\gamma$) would perform perfect channel inversion [Paré et al., 1996] (ignoring attenuation)
- Digital backpropagation: invert a partial differential equation in real time
  [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]

Digital Backpropagation



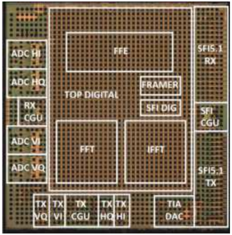
- Fiber with negated parameters ($\beta_2 \to -\beta_2$, $\gamma \to -\gamma$) would perform perfect channel inversion [Paré et al., 1996] (ignoring attenuation)
- Digital backpropagation: invert a partial differential equation in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]
- 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

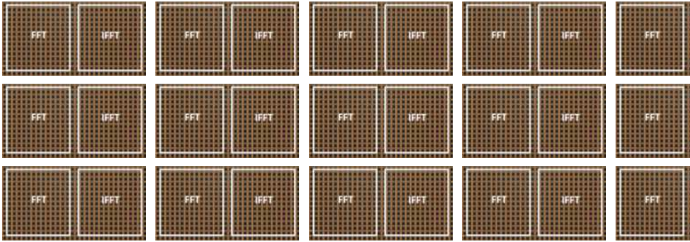


[Crivelli et al., 2014]

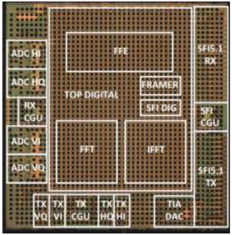
Real-Time Digital Backpropagation



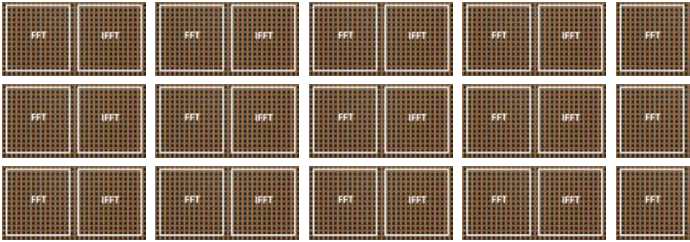
[Crivelli et al., 2014]



Real-Time Digital Backpropagation

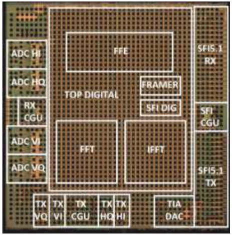


[Crivelli et al., 2014]

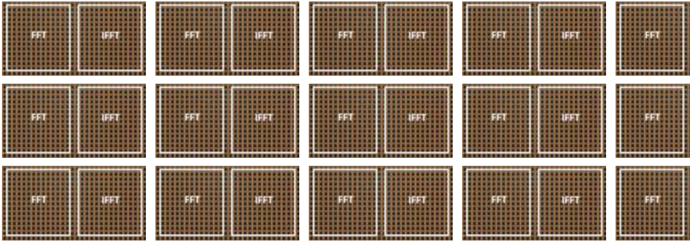


- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)

Real-Time Digital Backpropagation

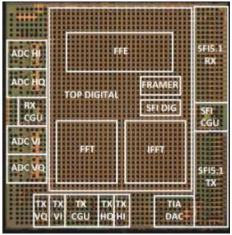


[Crivelli et al., 2014]

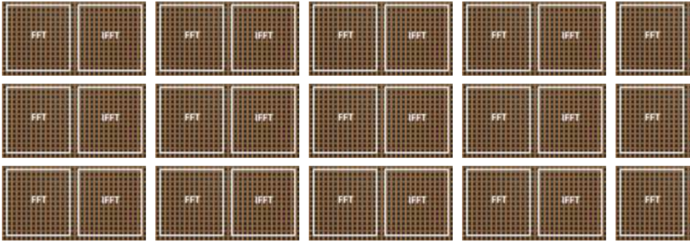


- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], ...)
- Intuitive, but ...

Real-Time Digital Backpropagation

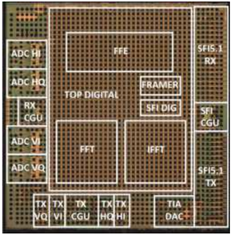


[Crivelli et al., 2014]

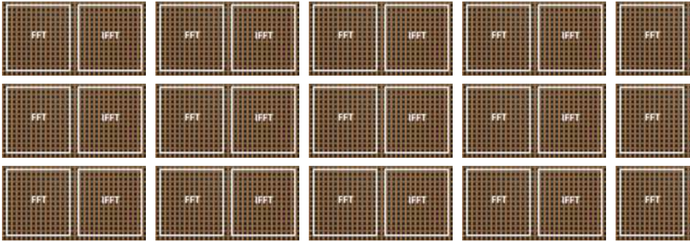


- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], ...)
- Intuitive, but ... this flattens a deep (multi-layer) computation graph

Real-Time Digital Backpropagation



[Crivelli et al., 2014]



- 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

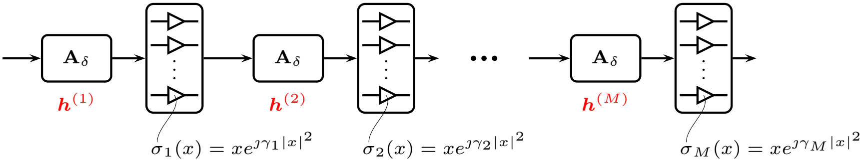
Our approach: physics-based deep learning and model compression

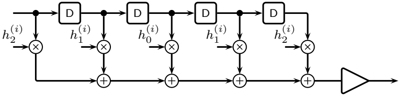
Joint optimization, pruning, and quantization of all linear steps $\implies$ hardware-efficient digital backpropagation

Learned Digital Backpropagation

Learned Digital Backpropagation

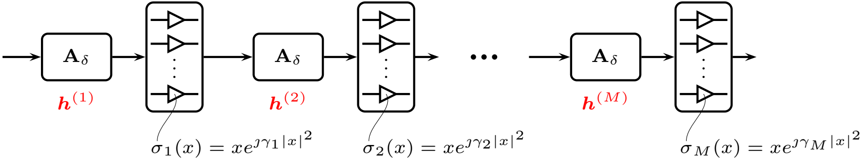
TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:





Learned Digital Backpropagation

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



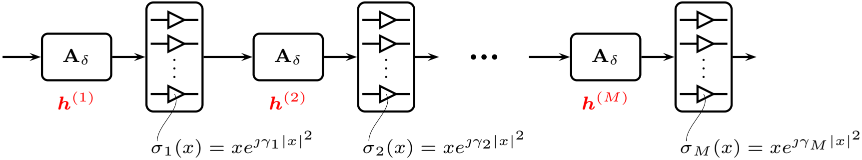
Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \ldots, \boldsymbol{h}^{(M)}\}$:

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

mean squared error — Adam optimizer, fixed learning rate

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

Iteratively prune (set to 0) outermost filter taps during gradient descent

Iterative Filter Tap Pruning

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} \\ \boldsymbol{h}^{(2)} \\ \vdots \\ \boldsymbol{h}^{(M)} \end{cases}$$

Iterative Filter Tap Pruning

starting length $2K' + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; h^{(1)}_{K'} \; \cdots \; h^{(1)}_{K} \; \cdots \; h^{(1)}_{1} \; h^{(1)}_{0} \; h^{(1)}_{1} \; \cdots \; h^{(1)}_{K} \; \cdots \; h^{(1)}_{K'} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; h^{(2)}_{K'} \; \cdots \; h^{(2)}_{K} \; \cdots \; h^{(2)}_{1} \; h^{(2)}_{0} \; h^{(2)}_{1} \; \cdots \; h^{(2)}_{K} \; \cdots \; h^{(2)}_{K'} \;) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ \boldsymbol{h}^{(M)} = (\; h^{(M)}_{K'} \; \cdots \; h^{(M)}_{K} \; \cdots \; h^{(M)}_{1} \; h^{(M)}_{0} \; h^{(M)}_{1} \; \cdots \; h^{(M)}_{K} \; \cdots \; h^{(M)}_{K'} \;) & \text{step } M \end{cases}$$

Iterative Filter Tap Pruning

$$\longleftarrow \text{starting length } 2K'+1 \longrightarrow$$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; h_{K'}^{(1)} \; \cdots \; h_{K}^{(1)} \; \cdots \; h_{1}^{(1)} \; h_{0}^{(1)} \; h_{1}^{(1)} \; \cdots \; h_{K}^{(1)} \; \cdots \; h_{K'}^{(1)} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; h_{K'}^{(2)} \; \cdots \; h_{K}^{(2)} \; \cdots \; h_{1}^{(2)} \; h_{0}^{(2)} \; h_{1}^{(2)} \; \cdots \; h_{K}^{(2)} \; \cdots \; h_{K'}^{(2)} \;) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ \boldsymbol{h}^{(M)} = (\; h_{K'}^{(M)} \; \cdots \; h_{K}^{(M)} \; \cdots \; h_{1}^{(M)} \; h_{0}^{(M)} \; h_{1}^{(M)} \; \cdots \; h_{K}^{(M)} \; \cdots \; h_{K'}^{(M)} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; h_{K'}^{(1)} \; \cdots \; h_K^{(1)} \; \cdots \; h_1^{(1)} \; h_0^{(1)} \; h_1^{(1)} \; \cdots \; h_K^{(1)} \; \cdots \; h_{K'}^{(1)} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; h_{K'}^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; h_1^{(2)} \; h_0^{(2)} \; h_1^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; h_{K'}^{(2)} \;) & \text{step } 2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots & \\ \boldsymbol{h}^{(M)} = (\; h_{K'}^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_1^{(M)} \; h_0^{(M)} \; h_1^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_{K'}^{(M)} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; \cancel{h_{K'}^{(1)}} \; \cdots \; h_K^{(1)} \; \cdots \; h_1^{(1)} \; h_0^{(1)} \; h_1^{(1)} \; \cdots \; h_K^{(1)} \; \cdots \; \cancel{h_{K'}^{(1)}} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; h_{K'}^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; h_1^{(2)} \; h_0^{(2)} \; h_1^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; h_{K'}^{(2)} \;) & \text{step } 2 \\ \vdots \\ \boldsymbol{h}^{(M)} = (\; h_{K'}^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_1^{(M)} \; h_0^{(M)} \; h_1^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_{K'}^{(M)} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; \cancel{h_{K'}^{(1)}} \; \cdots \; h_K^{(1)} \; \cdots \; h_1^{(1)} \; h_0^{(1)} \; h_1^{(1)} \; \cdots \; h_K^{(1)} \; \cdots \; \cancel{h_{K'}^{(1)}} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; \cancel{h_{K'}^{(2)}} \; \cdots \; h_K^{(2)} \; \cdots \; h_1^{(2)} \; h_0^{(2)} \; h_1^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; \cancel{h_{K'}^{(2)}} \;) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ \boldsymbol{h}^{(M)} = (\; h_{K'}^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_1^{(M)} \; h_0^{(M)} \; h_1^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_{K'}^{(M)} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

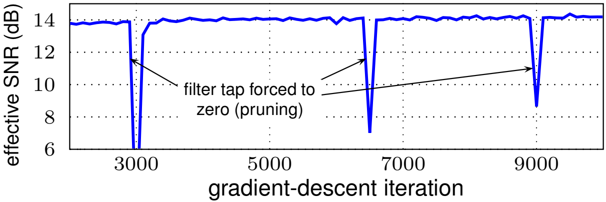
Iterative Filter Tap Pruning

starting length $2K' + 1$

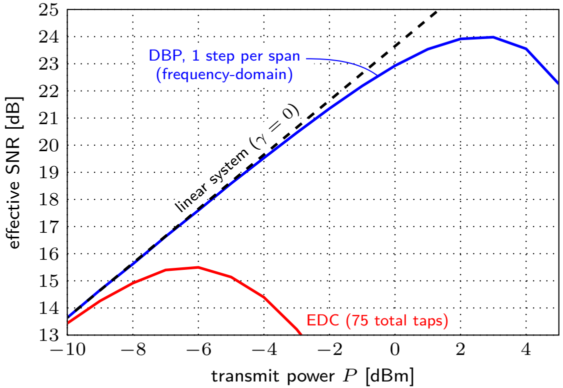
target length $2K + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\cancel{h_{K'}^{(1)}} \; \cdots \; h_K^{(1)} \; \cdots \; h_1^{(1)} \; h_0^{(1)} \; h_1^{(1)} \; \cdots \; h_K^{(1)} \; \cdots \; \cancel{h_{K'}^{(1)}} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\cancel{h_{K'}^{(2)}} \; \cdots \; h_K^{(2)} \; \cdots \; h_1^{(2)} \; h_0^{(2)} \; h_1^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; \cancel{h_{K'}^{(2)}} \;) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ \boldsymbol{h}^{(M)} = (h_{K'}^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_1^{(M)} \; h_0^{(M)} \; h_1^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_{K'}^{(M)} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]
- Typical learning curve:



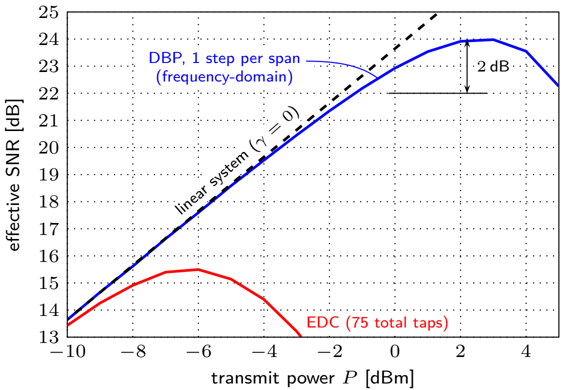
Revisiting Ip and Kahn (2008)



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.

Revisiting Ip and Kahn (2008)

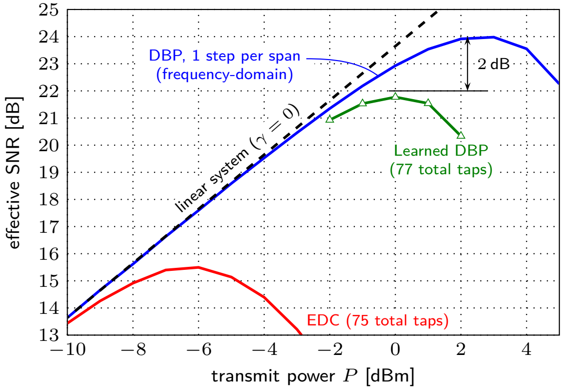


Parameters similar to [Ip and Kahn, 2008]:

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

- $\gg 1000$ total taps (70 taps/step) $\implies$ $> 100\times$ complexity of EDC

Revisiting Ip and Kahn (2008)

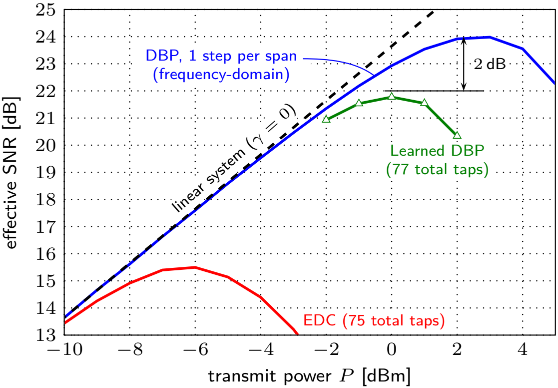


Parameters similar to [Ip and Kahn, 2008]:

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

- $\gg 1000$ total taps (70 taps/step) $\implies$ $> 100\times$ complexity of EDC
- Learned approach uses only 77 total taps: alternate 5 and 3 taps/step and use different filter coefficients in all steps [Häger and Pfister, 2018a]

Revisiting Ip and Kahn (2008)

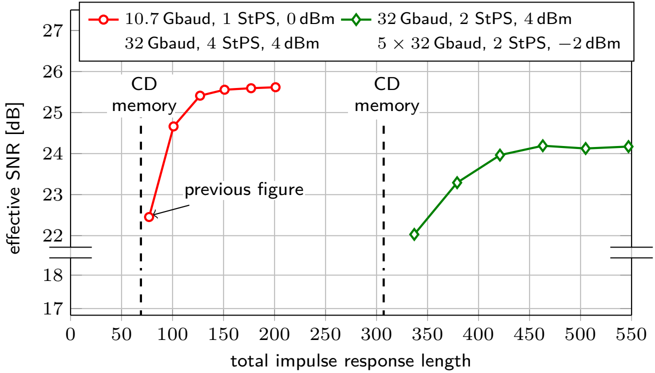


Parameters similar to [Ip and Kahn, 2008]:

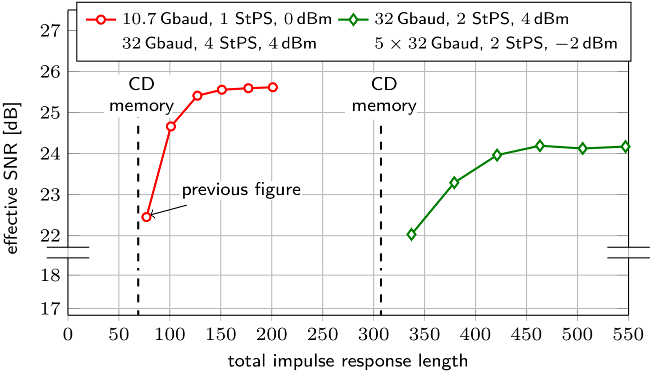
- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

- $\gg 1000$ total taps (70 taps/step) $\implies$ $> 100\times$ complexity of EDC
- Learned approach uses only 77 total taps: alternate 5 and 3 taps/step and use different filter coefficients in all steps [Häger and Pfister, 2018a]
- Can outperform "ideal DBP" in the nonlinear regime [Häger and Pfister, 2018b]

Performance–Complexity Trade-off



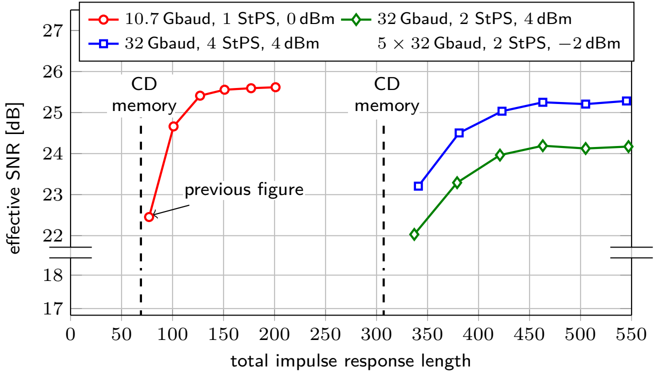
Performance–Complexity Trade-off



Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity $\overset{?}{\approx}$ Number of Steps

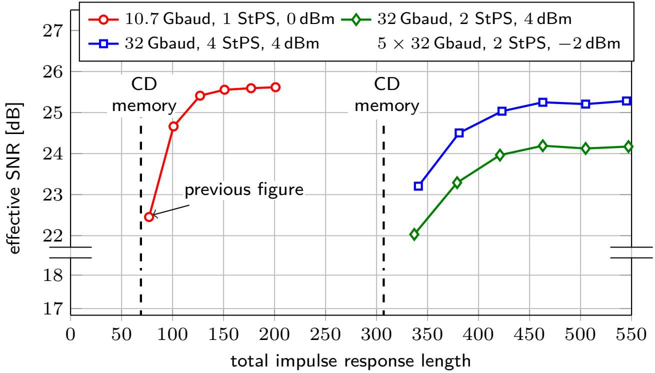
Performance–Complexity Trade-off



Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity $\overset{?}{\approx}$ Number of Steps

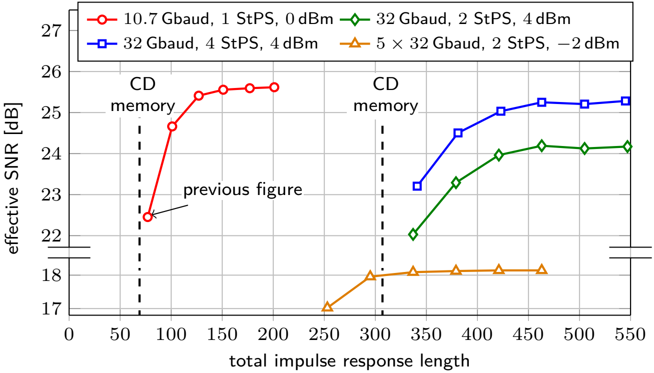
Performance–Complexity Trade-off



Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity = Number of Steps × Complexity per Step

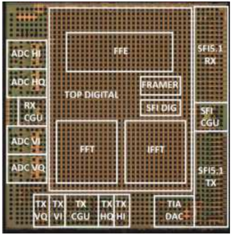
Performance–Complexity Trade-off



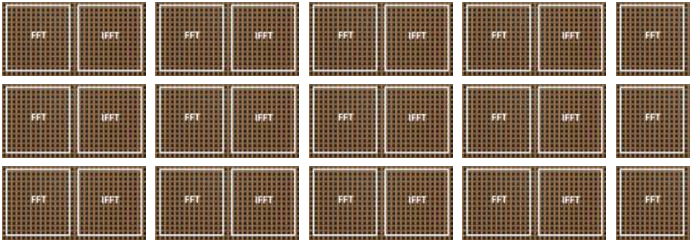
Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity	=	Number of Steps	×	Complexity per Step

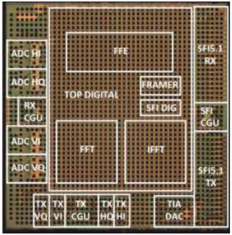
Real-Time ASIC Implementation



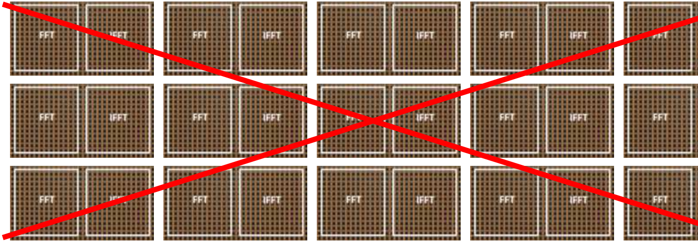
[Crivelli et al., 2014]



Real-Time ASIC Implementation

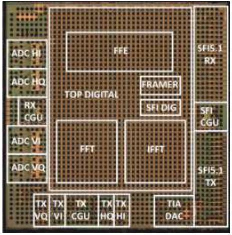


[Crivelli et al., 2014]

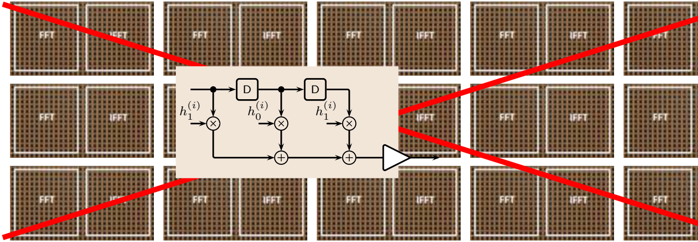


[Fougstedt et al., 2017], Time-domain digital back propagation: Algorithm and finite-precision implementation aspects, (*OFC*)
[Fougstedt et al., 2018], ASIC implementation of time-domain digital back propagation for coherent receivers, *(PTL)*
[Sherborne et al., 2018], On the impact of fixed point hardware for optical fiber nonlinearity compensation algorithms, (*JLT*)

Real-Time ASIC Implementation

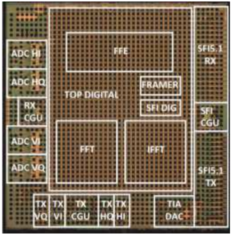


[Crivelli et al., 2014]

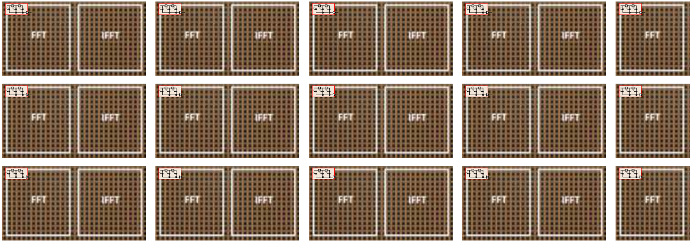


- Our linear steps are very short symmetric FIR filters (as few as 3 taps)

Real-Time ASIC Implementation



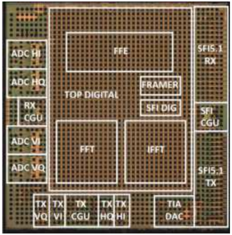
[Crivelli et al., 2014]



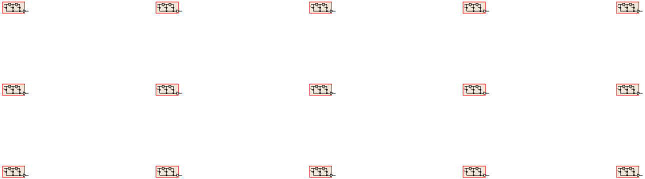
- Our linear steps are very short symmetric FIR filters (as few as 3 taps)
- 28-nm ASIC at 416.7 MHz clock speed (40 GHz signal)
  - Only 5-6 bit filter coefficients via learned quantization
  - Hardware-friendly nonlinear steps (Taylor expansion)
  - All FIR filters are fully reconfigurable

[Fougstedt et al., 2018], ASIC implementation of time-domain digital backpropagation with deep-learned chromatic dispersion filters, *(ECOC)*

Real-Time ASIC Implementation



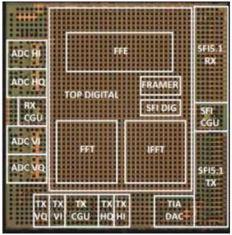
[Crivelli et al., 2014]



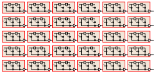
- Our linear steps are very short symmetric FIR filters (as few as 3 taps)
- 28-nm ASIC at 416.7 MHz clock speed (40 GHz signal)
  - Only 5-6 bit filter coefficients via learned quantization
  - Hardware-friendly nonlinear steps (Taylor expansion)
  - All FIR filters are fully reconfigurable

[Fougstedt et al., 2018], ASIC implementation of time-domain digital backpropagation with deep-learned chromatic dispersion filters, *(ECOC)*

Real-Time ASIC Implementation



[Crivelli et al., 2014]



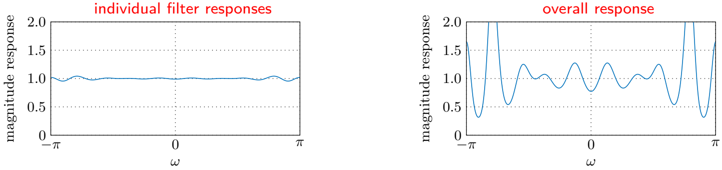
- Our linear steps are very short symmetric FIR filters (as few as 3 taps)
- 28-nm ASIC at 416.7 MHz clock speed (40 GHz signal)
  - Only 5-6 bit filter coefficients via learned quantization
  - Hardware-friendly nonlinear steps (Taylor expansion)
  - All FIR filters are fully reconfigurable
- $< 2\times$ power compared to EDC [Crivelli et al., 2014, Pillai et al., 2014]

[Fougstedt et al., 2018], ASIC implementation of time-domain digital backpropagation with deep-learned chromatic dispersion filters, *(ECOC)*

Why Does The Learning Approach Work?

Previous work: design a single filter or filter pair and use it repeatedly.

$\Longrightarrow$ Good overall response only possible with very long filters.



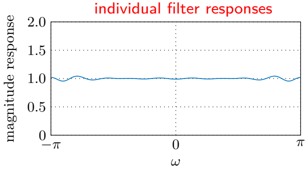
From [Ip and Kahn, 2009]:

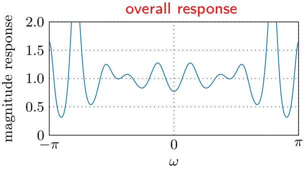
- “We also note that [. . .] 70 taps, is much larger than expected”
- “This is due to amplitude ringing in the frequency domain”
- “Since backpropagation requires multiple iterations of the linear filter, amplitude distortion due to ringing accumulates (Goldfarb & Li, 2009)”

Why Does The Learning Approach Work?

Previous work: design a single filter or filter pair and use it repeatedly.

$\Longrightarrow$ Good overall response only possible with very long filters.





From [Ip and Kahn, 2009]:

- "We also note that [...] 70 taps, is much larger than expected"
- "This is due to amplitude ringing in the frequency domain"
- "Since backpropagation requires multiple iterations of the linear filter, amplitude distortion due to ringing accumulates (Goldfarb & Li, 2009)"

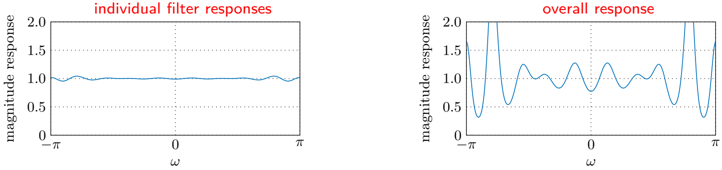
The learning approach uncovered that there is no such requirement!

[Lian, Häger, Pfister, 2018], What can machine learning teach us about communications? (*ITW*)

Why Does The Learning Approach Work?

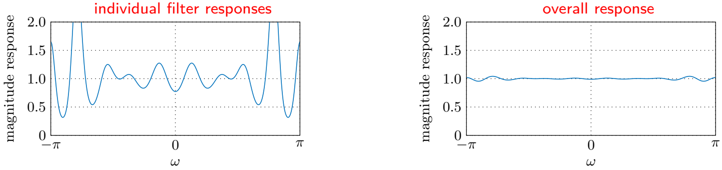
Previous work: design a single filter or filter pair and use it repeatedly.

$\Longrightarrow$ Good overall response only possible with very long filters.

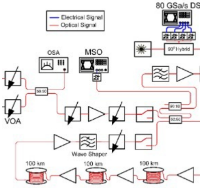


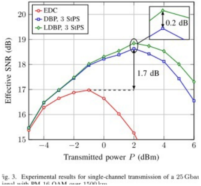
Sacrifice individual filter accuracy, but different response per step.

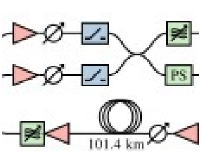
$\Longrightarrow$ Good overall response even with very short filters by joint optimization.

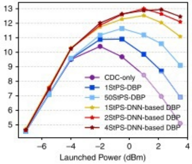


Experimental Investigations









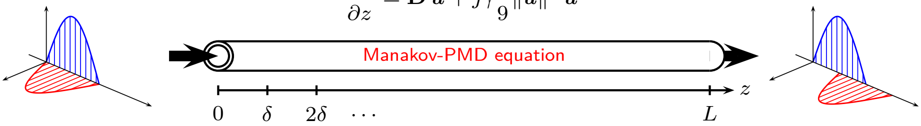
Training with real-world data sets including presence of various hardware impairments (phase noise, timing error, frequency offset, etc.)

- [Oliari et al., 2020], Revisiting Efficient Multi-step Nonlinearity Compensation with Machine Learning: An Experimental Demonstration, *(J. Lightw. Technol.)*
- [Sillekens et al., 2020], Experimental Demonstration of Learned Time-domain Digital Back-propagation, *(Proc. IEEE Workshop on Signal Processing Systems)*
- [Fan et al., 2020], Advancing Theoretical Understanding and Practical Performance of Signal Processing for Nonlinear Optical Communications through Machine Learning, *(Nat. Commun.)*
- [Bitachon et al., 2020], Deep learning based Digital Back Propagation Demonstrating SNR gain at Low Complexity in a 1200 km Transmission Link, *(Opt. Express)*

Outline

Evolution of Polarization-Multiplexed Signals

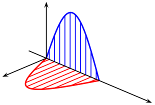
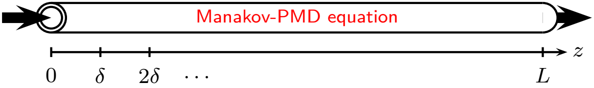
$$\frac{\partial \boldsymbol{u}}{\partial z} = \hat{\mathbf{D}}\,\boldsymbol{u} + \jmath\gamma\frac{8}{9}\|\boldsymbol{u}\|^2\,\boldsymbol{u}$$

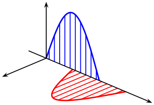


- Jones vector $\boldsymbol{u} \triangleq (u_1(t,z), u_2(t,z))^\top$ with complex baseband signals
- linear operator $\hat{\mathbf{D}}$: attentuation, chromatic & polarization mode dispersion

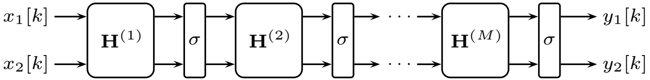
Evolution of Polarization-Multiplexed Signals

$$\frac{\partial \boldsymbol{u}}{\partial z} = \hat{\mathbf{D}}\,\boldsymbol{u} + \jmath\gamma\frac{8}{9}\|\boldsymbol{u}\|^2\,\boldsymbol{u}$$

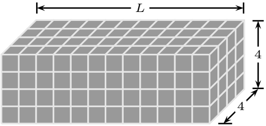





- Jones vector $\boldsymbol{u} \triangleq (u_1(t,z), u_2(t,z))^\top$ with complex baseband signals
- linear operator $\hat{\mathbf{D}}$: attentuation, chromatic & polarization mode dispersion
- Split-step method: alternate linear and nonlinear steps $\sigma(\boldsymbol{x}) = \boldsymbol{x}e^{\jmath\gamma\frac{8}{9}\delta\|\boldsymbol{x}\|^2}$

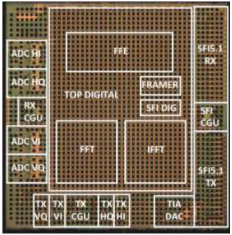


$e^{\delta\hat{\mathbf{D}}} \approx$

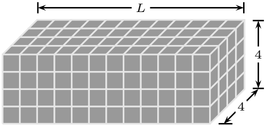


2×2 complex or 4×4 real MIMO filters $\Longrightarrow$ complexity!

Real-Time Compensation of Polarization Impairments

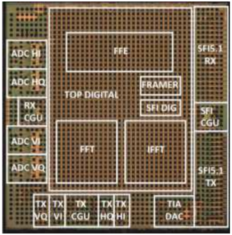


[Crivelli et al., 2014]

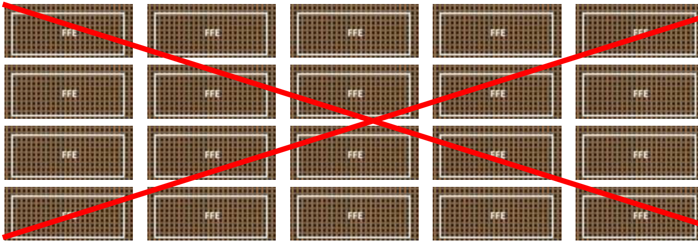


- time-varying effects (e.g., drifts) & apriori unknown realizations
- $\implies$ adaptive filtering (via stochastic gradient descent) required

Real-Time Compensation of Polarization Impairments



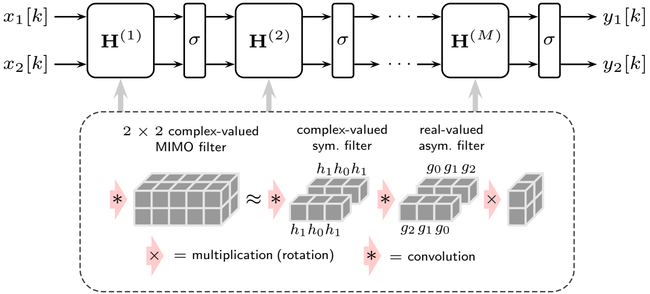
[Crivelli et al., 2014]



- time-varying effects (e.g., drifts) & apriori unknown realizations
- $\Longrightarrow$ adaptive filtering (via stochastic gradient descent) required

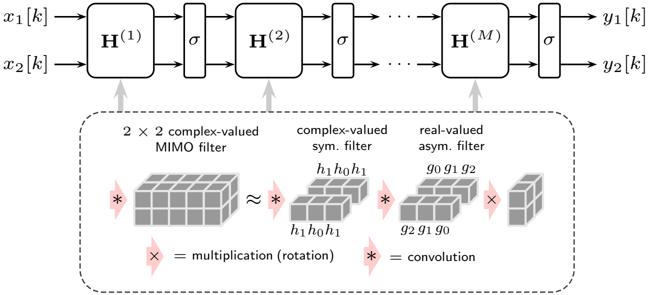
Using (and updating) full MIMO filters in each step is not feasible.

Our approach: Factorize each MIMO Filter



- 5-tap real-valued filters to approximate first-order PMD (DGD)
- Memoryless rotations $\begin{pmatrix} a & -b^* \\ b & a^* \end{pmatrix}$, where $a, b \in \mathbb{C}$ (4 real parameters)

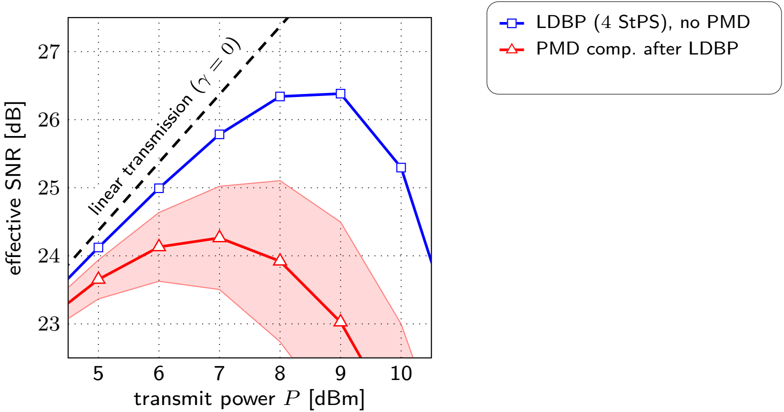
Our approach: Factorize each MIMO Filter



- 5-tap real-valued filters to approximate first-order PMD (DGD)
- Memoryless rotations $\begin{pmatrix} a & -b^* \\ b & a^* \end{pmatrix}$, where $a, b \in \mathbb{C}$ (4 real parameters)
- Assumes no knowledge about PMD realizations or accumulated PMD
- FIR-filter based! Avoids frequency-domain (FFT-based) filtering

[Goroshko et al., 2016], Overcoming performance limitations of digital back propagation due to polarization mode dispersion, *(CTON)*
[Czegledi et al., 2017], Digital backpropagation accounting for polarization-mode dispersion, *(Opt. Express)*
[Liga et al., 2018], A PMD-adaptive DBP receiver based on SNR optimization, *(OFC)*

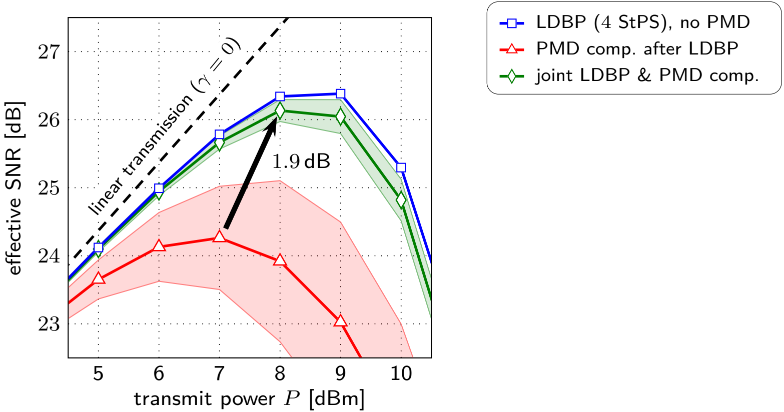
Results (32 Gbaud, 10×100 km, 0.2 ps/$\sqrt{\text{km}}$ PMD)



- Similar parameters & simulation setup compared to [Czegledi et al., 2016], results averaged over 40 PMD realizations

[Bütler et al., 2020], Model-based Machine Learning for Joint Digital Backpropagation and PMD Compensation, *(J. Lightw. Technol.)*, see arXiv:2010.12313

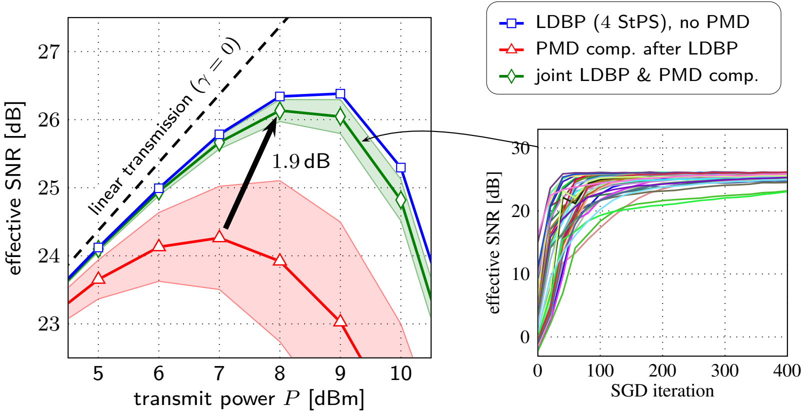
Results (32 Gbaud, 10×100 km, 0.2 ps/$\sqrt{\text{km}}$ PMD)



- Similar parameters & simulation setup compared to [Czegledi et al., 2016], results averaged over 40 PMD realizations

[Bütler et al., 2020], Model-based Machine Learning for Joint Digital Backpropagation and PMD Compensation, *(J. Lightw. Technol.)*, see arXiv:2010.12313

Results (32 Gbaud, 10×100 km, 0.2 ps/$\sqrt{\text{km}}$ PMD)



- Similar parameters & simulation setup compared to [Czegledi et al., 2016], results averaged over 40 PMD realizations
- Reliable convergence "from scratch" + only 9 real parameters per step

[Bütler et al., 2020], Model-based Machine Learning for Joint Digital Backpropagation and PMD Compensation, *(J. Lightw. Technol.)*, see arXiv:2010.12313

Outline

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

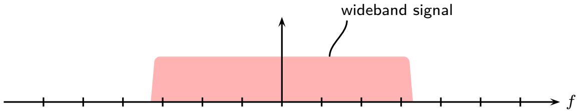
Consider a 96-Gbaud signal, where delay spread is 125 symbol periods per 100 km (alternatively: superchannel or multiple WDM channels).

- 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)

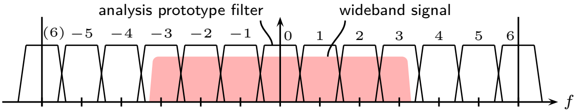
Question

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

Wideband Signals and Subband Processing



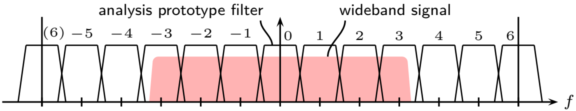
Wideband Signals and Subband Processing



- Subband processing: split received signal into N parallel signals

[Taylor, 2008], Compact digital dispersion compensation algorithms, *(OFC)*
[Ho, 2009], Subband equaliser for chromatic dispersion of optical fibre, *(Electronics Lett.)*
[Slim et al., 2013], Delayed single-tap frequency-domain chromatic-dispersion compensation, *(PTL)*
[Nazarathy and Tolmachev, 2014], Subbanded DSP architectures based on underdecimated filter banks . . . , *(Signal Proc. Mag.)*
[Mateo et al., 2010], Efficient compensation of inter-channel nonlinear effects via digital backward . . . , *(Opt. Express)*
[Ip et al., 2011], Complexity versus performance tradeoff for fiber nonlinearity compensation . . . *(OFC)*
[Oyama et al., 2015], Complexity reduction of perturbation-based nonlinear compensator by sub-band processing, *(OFC)*
. . .

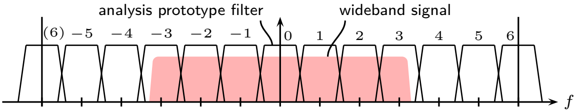
Wideband Signals and Subband Processing



- Subband processing: split received signal into N parallel signals
- Similar structure as popular convolutional neural networks (alternating filter banks and nonlinearities)

[Taylor, 2008], Compact digital dispersion compensation algorithms, *(OFC)*
[Ho, 2009], Subband equaliser for chromatic dispersion of optical fibre, *(Electronics Lett.)*
[Slim et al., 2013], Delayed single-tap frequency-domain chromatic-dispersion compensation, *(PTL)*
[Nazarathy and Tolmachev, 2014], Subbanded DSP architectures based on underdecimated filter banks . . . , *(Signal Proc. Mag.)*
[Mateo et al., 2010], Efficient compensation of inter-channel nonlinear effects via digital backward . . . , *(Opt. Express)*
[Ip et al., 2011], Complexity versus performance tradeoff for fiber nonlinearity compensation . . . *(OFC)*
[Oyama et al., 2015], Complexity reduction of perturbation-based nonlinear compensator by sub-band processing, *(OFC)*
. . .

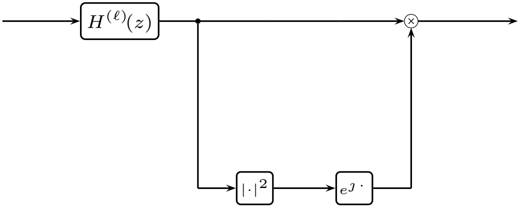
Wideband Signals and Subband Processing



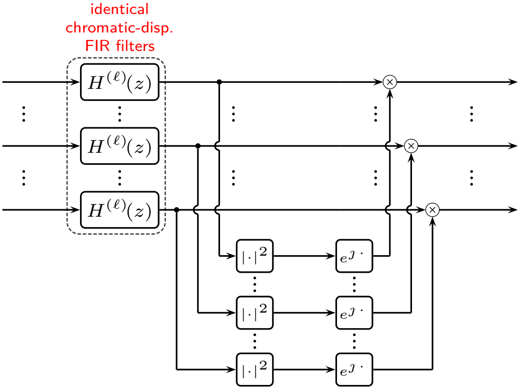
- Subband processing: split received signal into N parallel signals
- Similar structure as popular convolutional neural networks (alternating filter banks and nonlinearities)
- MIMO filter accounts for cross-phase modulation (XPM) between subbands [Leibrich and Rosenkranz, 2003]

[Taylor, 2008], Compact digital dispersion compensation algorithms, *(OFC)*
[Ho, 2009], Subband equaliser for chromatic dispersion of optical fibre, *(Electronics Lett.)*
[Slim et al., 2013], Delayed single-tap frequency-domain chromatic-dispersion compensation, *(PTL)*
[Nazarathy and Tolmachev, 2014], Subbanded DSP architectures based on underdecimated filter banks . . . , *(Signal Proc. Mag.)*
[Mateo et al., 2010], Efficient compensation of inter-channel nonlinear effects via digital backward . . . , *(Opt. Express)*
[Ip et al., 2011], Complexity versus performance tradeoff for fiber nonlinearity compensation . . . *(OFC)*
[Oyama et al., 2015], Complexity reduction of perturbation-based nonlinear compensator by sub-band processing, *(OFC)*
. . .

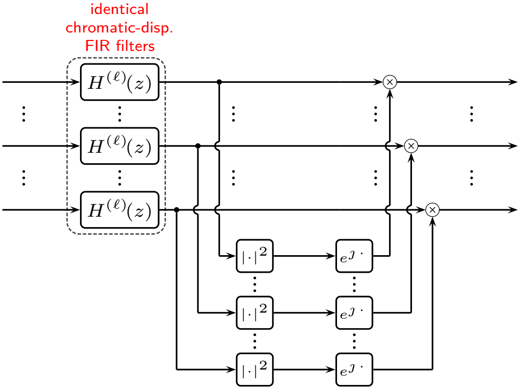
Proposed DSP Architecture (ℓ-th Step)



Proposed DSP Architecture (ℓ-th Step)

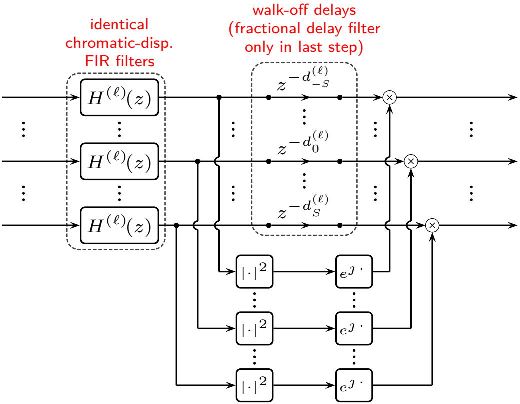


Proposed DSP Architecture (ℓ-th Step)



subband group delay differences depend linearly on propagation distance
$\implies$ choose step size such that delays are integer multiples of sampling interval

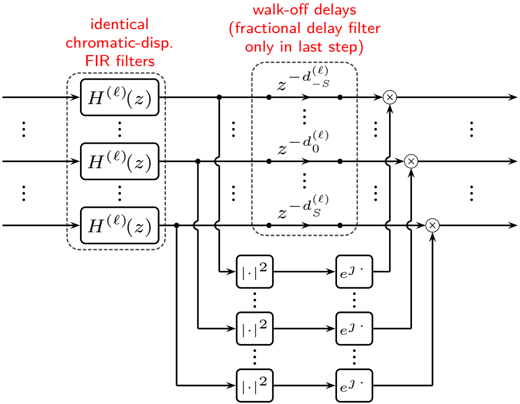
Proposed DSP Architecture (ℓ-th Step)



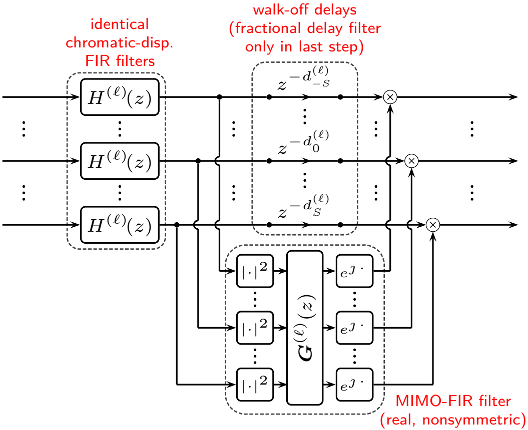
subband group delay differences depend linearly on propagation distance

$\Longrightarrow$ choose step size such that delays are integer multiples of sampling interval

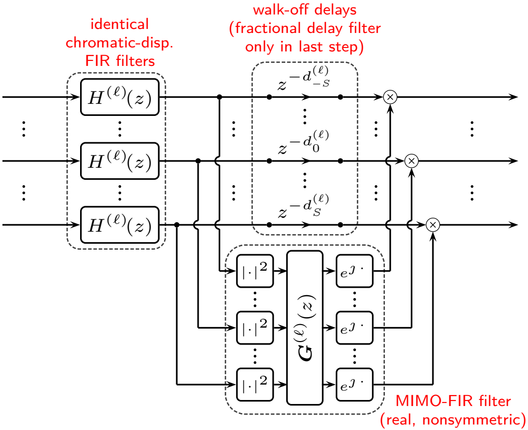
Proposed DSP Architecture (ℓ-th Step)



Proposed DSP Architecture (ℓ-th Step)

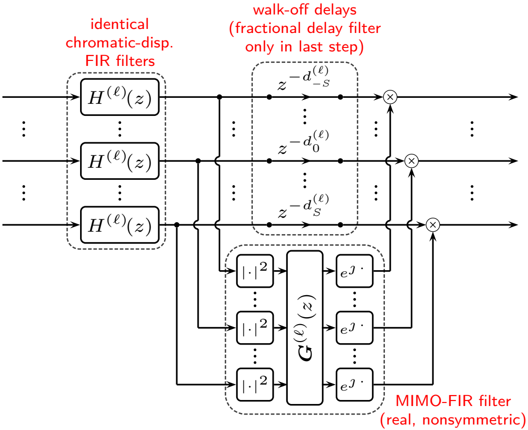


Proposed DSP Architecture (ℓ-th Step)

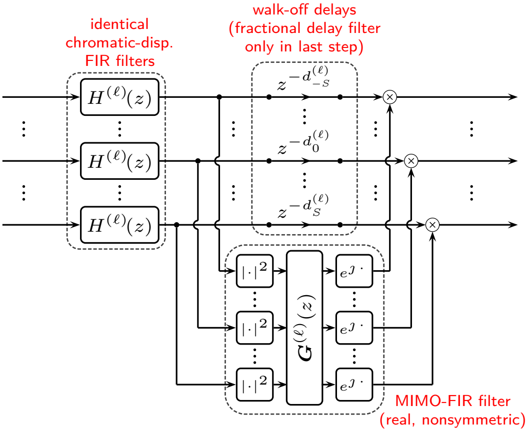


- Hardware-efficient implementation (no FFT/IFFT) of split-step method for coupled NLSEs [Leibrich and Rosenkranz, 2003], see also [Mateo et al., 2010]
- Only accounts for XPM between subbands, but not FWM

Proposed DSP Architecture (ℓ-th Step)

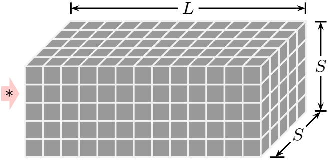


Proposed DSP Architecture (ℓ-th Step)



- "Unrolling" all steps gives a deep, multi-layer computation graph
- Deep learning to jointly optimize filters $H^{(\ell)}(z)$, $\mathbf{G}^{(\ell)}(z)$ in all steps by maximizing effective SNR based on stochastic gradient descent
- Iteratively prune (set to 0) the outermost taps to get very short filters

Wideband Signals and Subband Processing

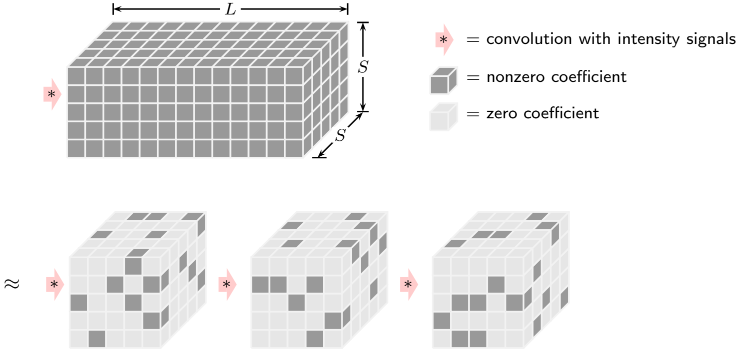


* = convolution with intensity signals

= nonzero coefficient

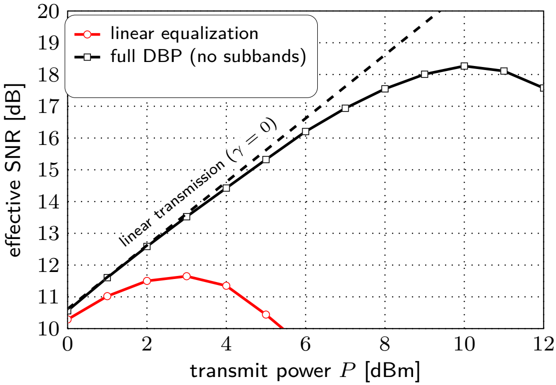
= zero coefficient

Wideband Signals and Subband Processing



- L_1-norm regularization applied to filter coefficients during gradient descent
- $\implies$ 92% of coefficients are zero with little performance penality

Results (12 Subbands for 96 Gbaud)

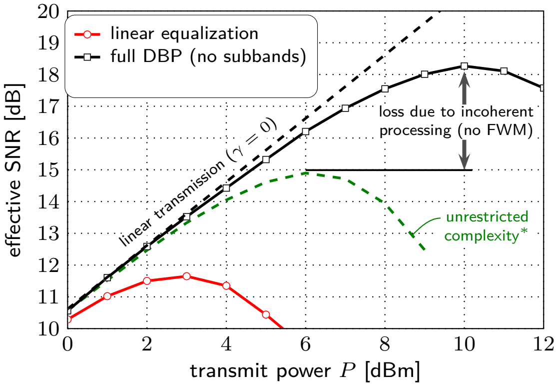


System parameters:

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

[Häger and Pfister, 2018], Wideband time-domain digital backpropagation via subband processing and deep learning, *(ECOC)*

Results (12 Subbands for 96 Gbaud)



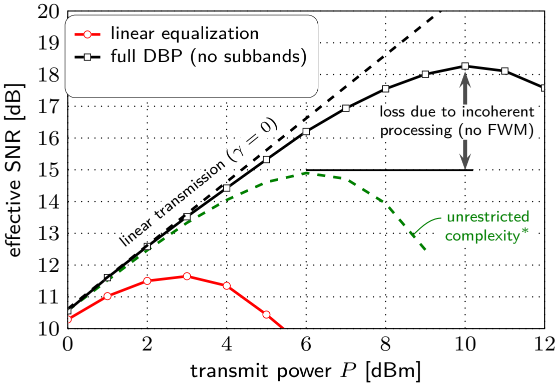
System parameters:

- 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

[Häger and Pfister, 2018], Wideband time-domain digital backpropagation via subband processing and deep learning, *(ECOC)*

Results (12 Subbands for 96 Gbaud)



System parameters:

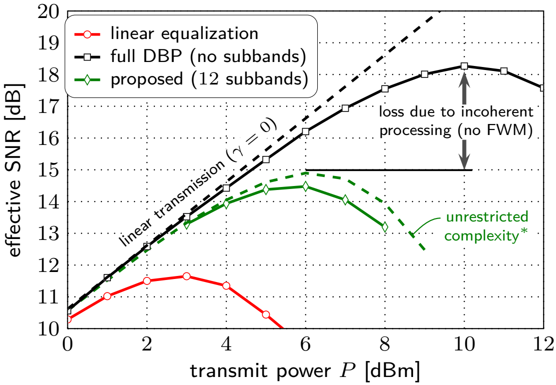
- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

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

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

[Häger and Pfister, 2018], Wideband time-domain digital backpropagation via subband processing and deep learning, *(ECOC)*

Results (12 Subbands for 96 Gbaud)



System parameters:

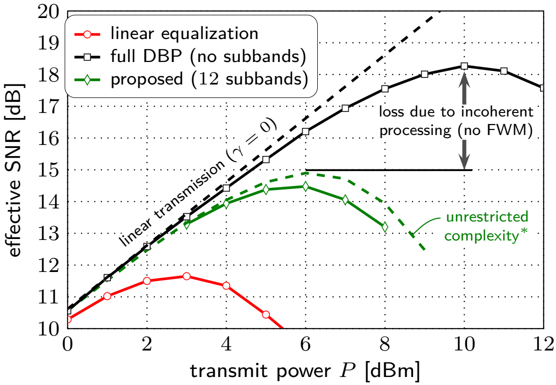
- 25×100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

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

- 1.5 subband oversampling, 38.2 km step size ($\approx$ 2.6 steps/span)

[Häger and Pfister, 2018], Wideband time-domain digital backpropagation via subband processing and deep learning, *(ECOC)*

Results (12 Subbands for 96 Gbaud)



System parameters:

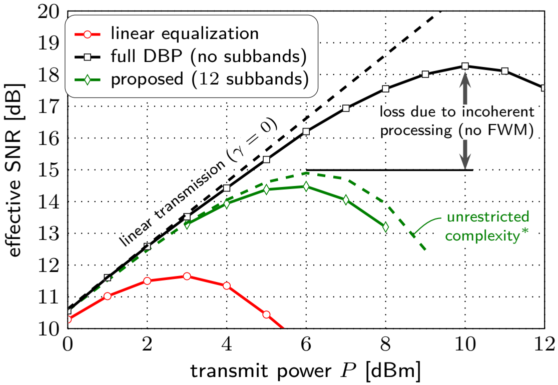
- 25 × 100 km fiber
- 96 Gbaud
- Gaussian symbols, single pol.
- RRC pulses (0.1 roll-off)

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

- 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.)

[Häger and Pfister, 2018], Wideband time-domain digital backpropagation via subband processing and deep learning, *(ECOC)*

Results (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

- 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.)
- $> 4\times$ less real multipl.compared to FFT/IFFT [Mateo et al., 2010]
- $\approx 2 - 3\times$ less complexity compared to full DBP (estimated)

[Häger and Pfister, 2018], Wideband time-domain digital backpropagation via subband processing and deep learning, *(ECOC)*

Outline

1. Machine Learning and Neural Networks for Communications
2. Physics-Based Machine Learning for Fiber-Optic Communications
3. Learned Digital Backpropagation
4. Polarization-Dependent Effects
5. Wideband Signals
6. Conclusions

The Bigger Picture

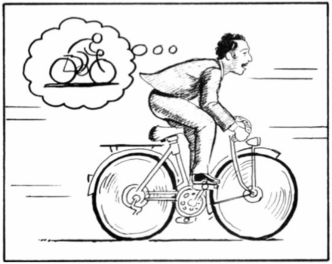
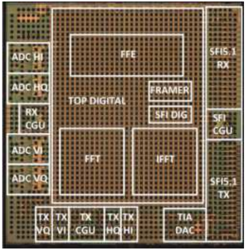


Figure 1. A World Model, from Scott McCloud's *Understanding Comics.* (McCloud, 1993; E, 2012)



[Crivelli et al., 2014]

- Optical receivers build models of their "environment"

[Ha & Schmidhuber, 2018], "World Models", arXiv:1803.10122 [cs.LG]

The Bigger Picture

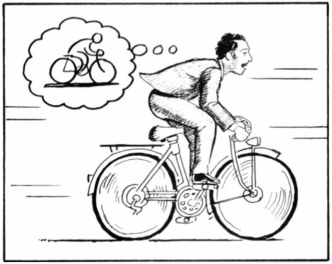
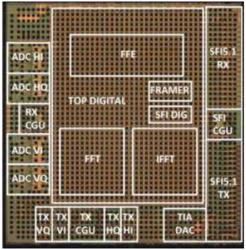


Figure 1. A World Model, from Scott McCloud's *Understanding Comics.* (McCloud, 1993; E, 2012)



[Crivelli et al., 2014]

- Optical receivers build models of their "environment"
- Currently these models are linear and/or rigid (non-adaptive)
- Interpretable physics-based "multi-layer" models for machine learning can be obtained by exploiting our existing domain knowledge

[Ha & Schmidhuber, 2018], "World Models", arXiv:1803.10122 [cs.LG]

Conclusions

Conclusions

neural-network-based ML
universal function approximators
good designs require experience and fine-tuning
black boxes, difficult to "open"

Conclusions

neural-network-based ML	**model-based ML**
universal function approximators	application-tailored
good designs require experience and fine-tuning	relies on domain knowledge (algorithms, physics, . . .)
black boxes, difficult to "open"	familiar building blocks (e.g., FIR filters) can enable interpretability

Conclusions

neural-network-based ML	**model-based ML**
universal function approximators	application-tailored
good designs require experience and fine-tuning	relies on domain knowledge (algorithms, physics, . . .)
black boxes, difficult to "open"	familiar building blocks (e.g., FIR filters) can enable interpretability

[Häger & Pfister, 2020], "Physics-Based Deep Learning for Fiber-Optic Communication Systems", in *IEEE J. Sel. Areas Commun.* (to appear), see `https://arxiv.org/abs/2010.14258`

Code: `https://github.com/chaeger/LDBP`

Conclusions

neural-network-based ML	**model-based ML**
universal function approximators	application-tailored
good designs require experience and fine-tuning	relies on domain knowledge (algorithms, physics, . . .)
black boxes, difficult to "open"	familiar building blocks (e.g., FIR filters) can enable interpretability

[Häger & Pfister, 2020], "Physics-Based Deep Learning for Fiber-Optic Communication Systems", in *IEEE J. Sel. Areas Commun.* (to appear), see `https://arxiv.org/abs/2010.14258`

Code: `https://github.com/chaeger/LDBP`

Thank you!

FORCE FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER Duke UNIVERSITY TU/e

References I

Borgerding, M. and Schniter, P. (2016).
Onsager-corrected deep learning for sparse linear inverse problems.
In *Proc. IEEE Global Conf. Signal and Information Processing (GlobalSIP)*, Washington, DC.

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.

Czegledi, C. B., Liga, G., Lavery, D., Karlsson, M., Agrell, E., Savory, S. J., and Bayvel, P. (2016).
Polarization-mode dispersion aware digital backpropagation.
In *Proc. European Conf. Optical Communication (ECOC)*, Düsseldorf, Germany.

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.

Gregor, K. and Lecun, Y. (2010).
Learning fast approximations of sparse coding.
In *Proc. Int. Conf. Mach. Learning (ICML)*, Haifa, Israel.

References II

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.

He, K., Zhang, X., Ren, S., and Sun, J. (2015).
Deep residual learning for image recognition.

Ip, E. and Kahn, J. M. (2008).
Compensation of dispersion and nonlinear impairments using digital backpropagation.
J. Lightw. Technol., 26(20):3416–3425.

Ip, E. and Kahn, J. M. (2009).
Nonlinear impairment compensation using backpropagation.
Optical Fiber New Developments, Chapter 10.

Lavery, D., Ives, D., Liga, G., Alvarado, A., Savory, S. J., and Bayvel, P. (2016).
The benefit of split nonlinearity compensation for single-channel optical fiber communications.
IEEE Photon. Technol. Lett., 28(17):1803–1806.

LeCun, Y., Bengio, Y., and Hinton, G. (2015).
Deep learning.
Nature, 521(7553):436–444.

References III

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)*, Los Angeles, CA.

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.

Li, Y., Ho, C. K., Wu, Y., and Sun, S. (2005).
Bit-to-symbol mapping in LDPC coded modulation.
In *Proc. Vehicular Technology Conf. (VTC)*, Stockholm, Sweden.

Lin, H. W., Tegmark, M., and Rolnick, D. (2017).
Why does deep and cheap learning work so well?
J. Stat. Phys., 168(6):1223–1247.

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):15144–15154.

References IV

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015).
Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533.

Nachmani, E., Be'ery, Y., and Burshtein, D. (2016).
Learning to decode linear codes using deep learning.
In *Proc. Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL.

Nakashima, H., Oyama, T., Ohshima, C., Akiyama, Y., Tao, Z., and Hoshida, T. (2017).
Digital nonlinear compensation technologies in coherent optical communication systems.
In *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA.

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):1351–1362.

O'Shea, T. and Hoydis, J. (2017).
An introduction to deep learning for the physical layer.
IEEE Trans. Cogn. Commun. Netw., 3(4):563–575.

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

References V

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.

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.