Learning to Communicate
From Theory to Practical Over-the-Air Transmission

S. Cammerer[1], S. Dörner[1], J. Hoydis[2], S. ten Brink[1]

[1] University of Stuttgart, Germany

[2] Nokia Bell Labs, France

JWCC 2019, Kühtai, Austria

March 11, 2019

University of Stuttgart
Institute of Telecommunications
Prof. Dr. Ing. Stephan ten Brink

NOKIA Bell Labs

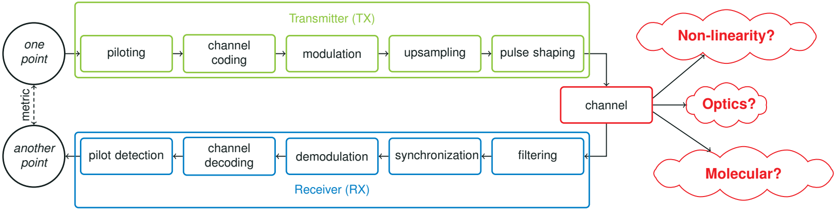
"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point." — Claude E. Shannon, 1948

Outline

Agenda

1. Motivation
2. Learning to Communicate: The Autoencoder
3. From Theory to Practical Over-the-Air Transmission
4. Results for Over-the-air Transmission
5. Conclusion

How it is done today?

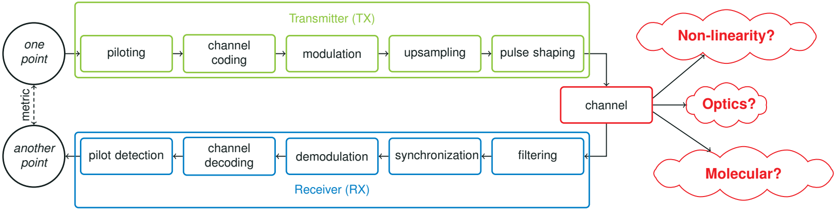


Block-based engineering design

- Independently optimized blocks for specific tasks

→ **Optimality requires many (strong) assumptions!**

How it is done today?



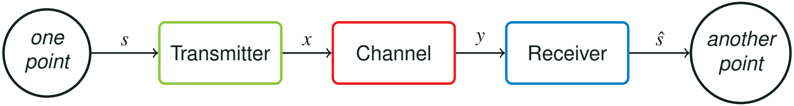
Block-based engineering design

- Independently optimized blocks for specific tasks

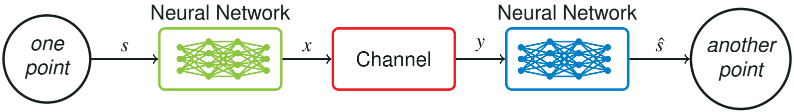
→ **Optimality requires many (strong) assumptions!**

Can we build a system that *learns* to communicate?

The End-to-End Learning Idea



The End-to-End Learning Idea

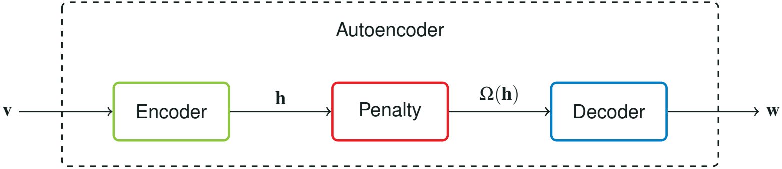


Replace transmitter and receiver by deep neural networks

- No underlying block structure
- All signal processing needs to be learned
- **End-to-end learning**
  - Train to minimize most relevant *end-to-end* metric (e.g., error rate or mutual information)

Agenda

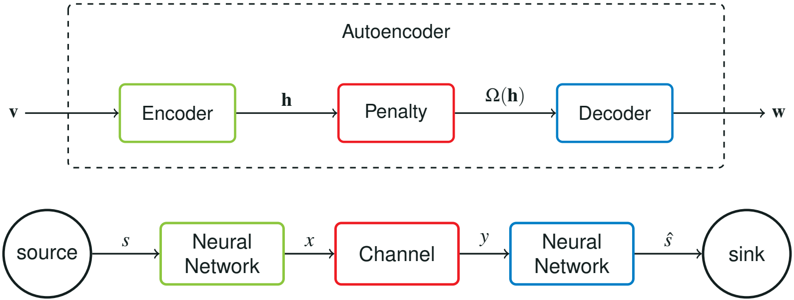
Autoencoder Neural Network



The autoencoder network as communication system

- Perfect fit: channel as penalty layer
- Transmitter is trained to find robust signal x
- Receiver must be able to decode signal y
- System can be trained end-to-end

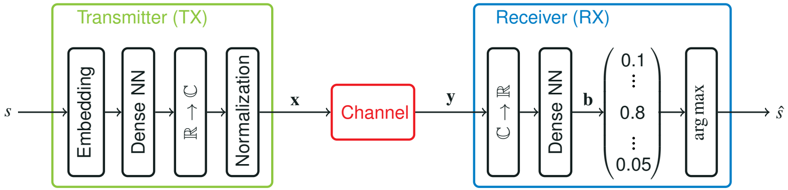
Autoencoder Neural Network



The autoencoder network as communication system

- Perfect fit: channel as penalty layer
- Transmitter is trained to find robust signal x
- Receiver must be able to decode signal y
- System can be trained end-to-end

Autoencoder Layer Structure



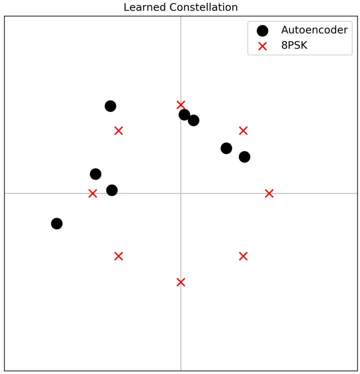
Transmitter Neural Network

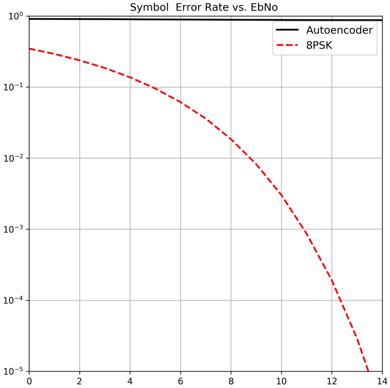
- Input: message $s \in \mathbb{M} = \{1, 2, \ldots, M\}$
- Output: normalized complex symbol sequence $\mathbf{x}$

Receiver Neural Network

- Input: complex sample sequence $\mathbf{y}$
- Output: message $\hat{s} \in \mathbb{M} = \{1, 2, \ldots, M\}$

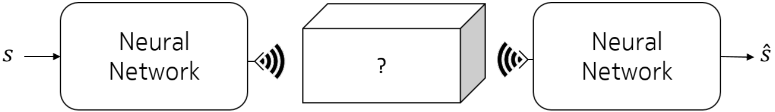
Learning 8-PSK





Agenda

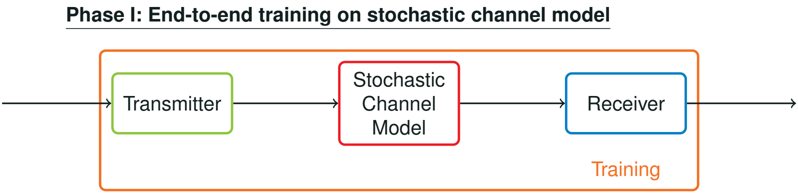
The Gap Between Theory and Practice



- For any real system, the channel is a black box
- Unknown transfer function $\rightarrow$ Gradient backpropagation impossible

$\rightarrow$ ***How can we apply the concept to a real system?***

Solution 1: Two-phase Training Strategy

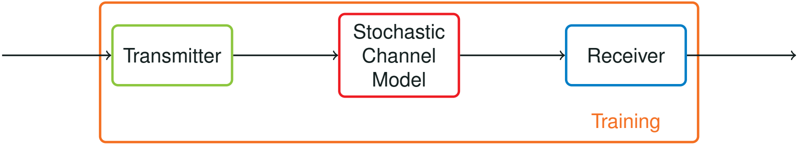


Using a stochastic channel model

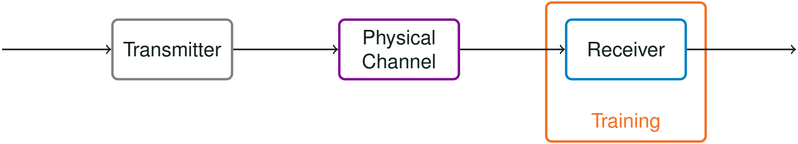
- Described by computable analytic functions
- Must be close to physical channel
- Simple LOS model: phase offset, CFO, SFO and AWGN
- Can be extended (e.g., channel tabs, hardware imperfections)

Solution 1: Two-phase Training Strategy

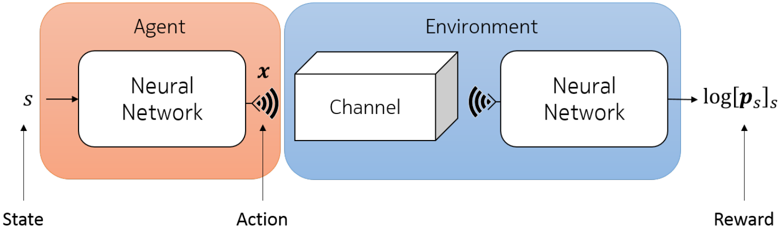
Phase I: End-to-end training on stochastic channel model



Phase II: Receiver finetuning on physical channel



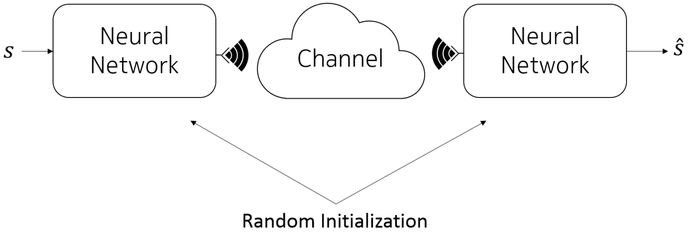
Solution 2: Reinforcement Learning[1]



- The transmitter observes the state $s \in \mathcal{M} = [1, \ldots, M]$,
- ...takes the action $\mathbf{x} = f_{\theta_t}(s)$,
- ...and observes the reward $\log\left[\mathbf{p}_s\right]_s \triangleq -l$
- Problem: $\text{argmax}_{\theta_t} \mathbb{E}\left[\log\left[\mathbf{p}_s\right]_s\right] = \text{argmin}_{\theta_t} \ \mathbb{E}\left[l\right]$

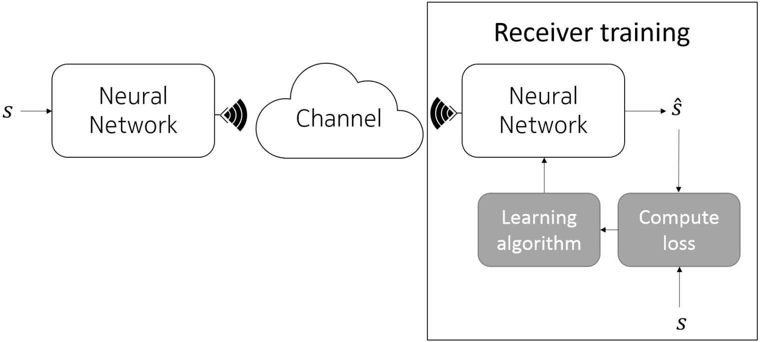
[1]F. Aoudia, J. Hoydis. "Model-free Training of End-to-end Communication Systems." arXiv preprint, 2018

Solution 2: Reinforcement Learning[2]



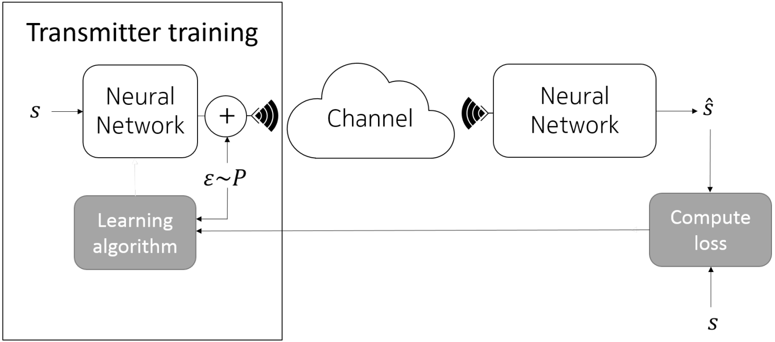
[2] F. Aoudia, J. Hoydis. "Model-free Training of End-to-end Communication Systems." arXiv preprint, 2018

Solution 2: Reinforcement Learning[2]



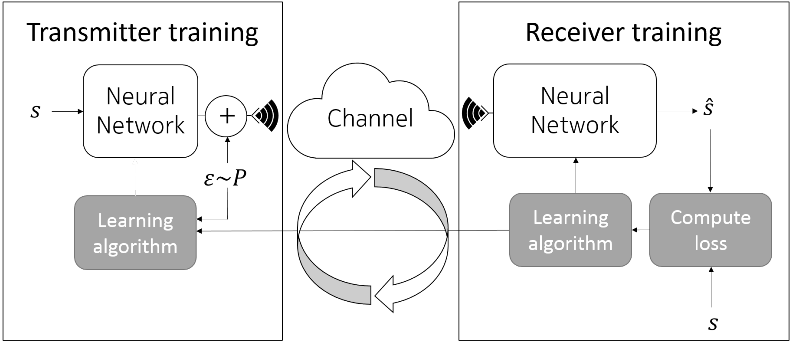
[2] F. Aoudia, J. Hoydis. "Model-free Training of End-to-end Communication Systems." arXiv preprint, 2018

Solution 2: Reinforcement Learning[2]



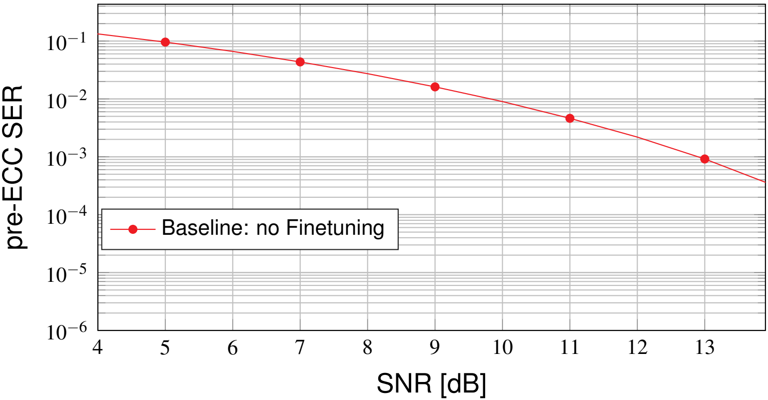
[2] F. Aoudia, J. Hoydis. "Model-free Training of End-to-end Communication Systems." arXiv preprint, 2018

Solution 2: Reinforcement Learning[2]

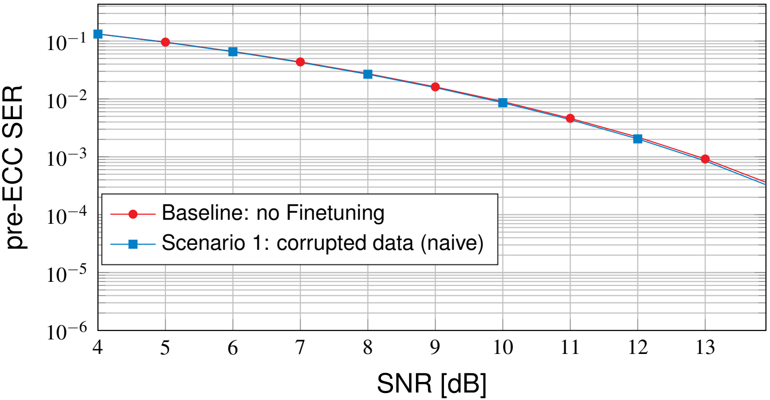


[2] F. Aoudia, J. Hoydis. "Model-free Training of End-to-end Communication Systems." arXiv preprint, 2018

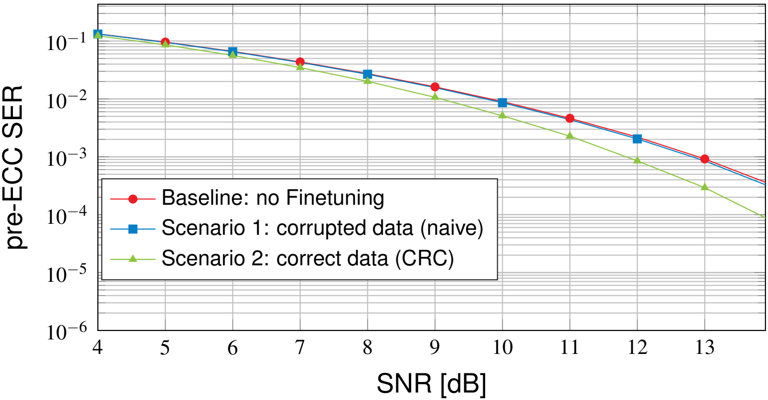
Adaptivity - How to Acquire Training Data?



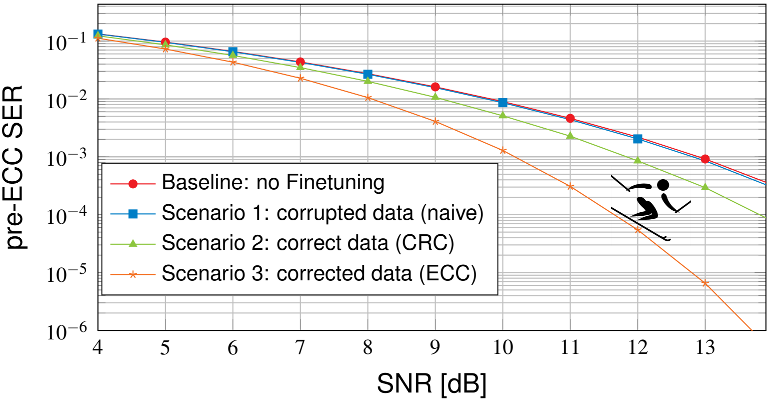
Adaptivity - How to Acquire Training Data?



Adaptivity - How to Acquire Training Data?

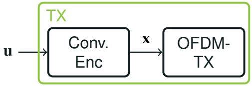


Adaptivity - How to Acquire Training Data?



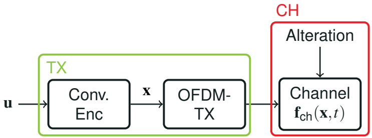
→ ***From errors one learns***

Label Recovery[3]



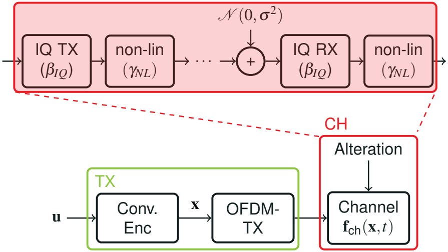
[3] S. Schibisch et al. "Online Label Recovery for Deep Learning-based Communication through error correcting codes", ISWCS, 2018

Label Recovery[3]



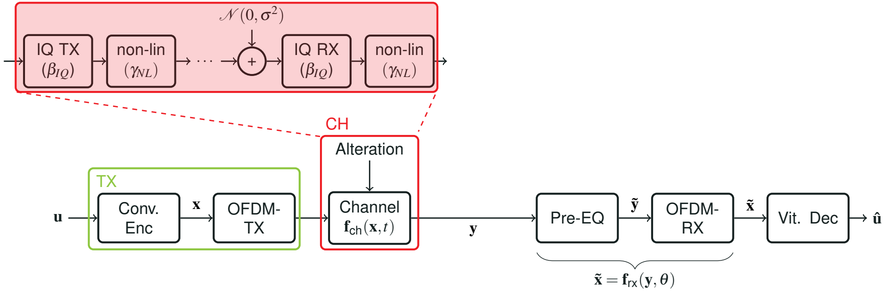
[3]S. Schibisch et al. "Online Label Recovery for Deep Learning-based Communication through error correcting codes", ISWCS, 2018

Label Recovery[3]



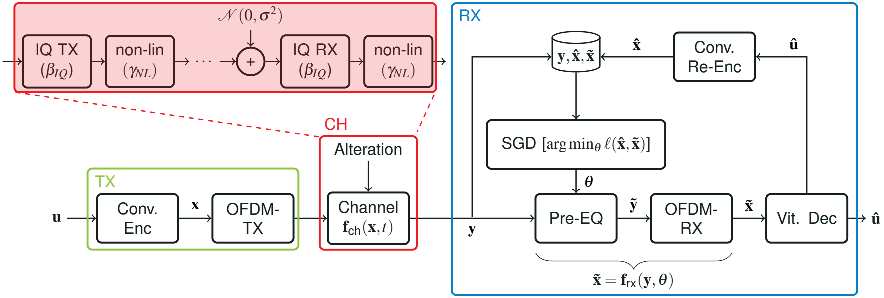
[3]S. Schibisch et al. "Online Label Recovery for Deep Learning-based Communication through error correcting codes", ISWCS, 2018

Label Recovery[3]



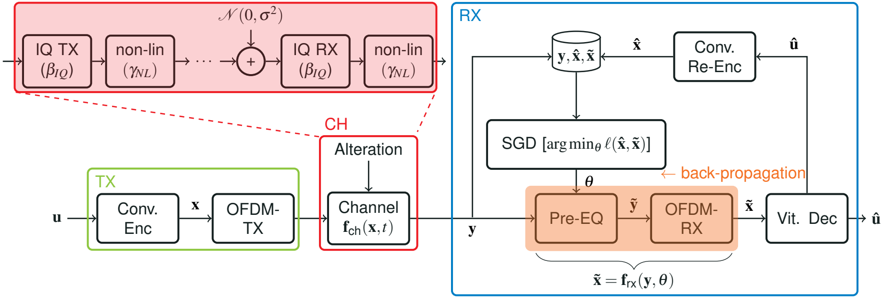
[3]S. Schibisch et al. "Online Label Recovery for Deep Learning-based Communication through error correcting codes", ISWCS, 2018

Label Recovery[3]



[3] S. Schibisch et al. "Online Label Recovery for Deep Learning-based Communication through error correcting codes", ISWCS, 2018

Label Recovery[3]



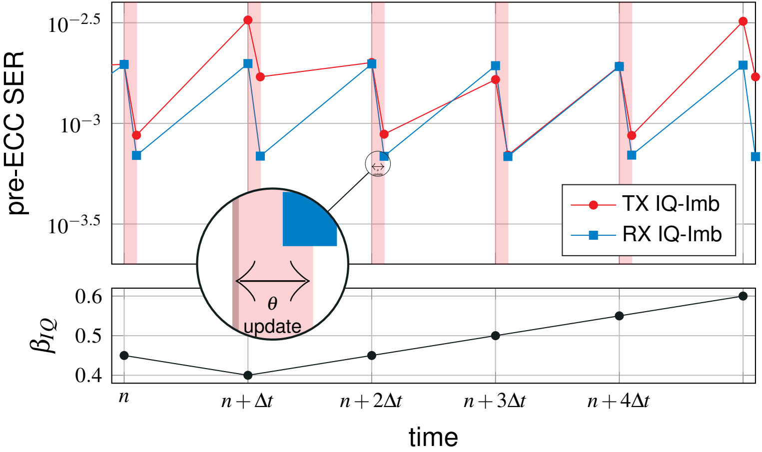
- Physical channel has no gradient (!)
  - → Only receiver can be finetuned

[3] S. Schibisch et al. "Online Label Recovery for Deep Learning-based Communication through error correcting codes", ISWCS, 2018

University of Stuttgart
Institute of Telecommunications

Adaptivity - IQ Imbalance

- SGD-based training is performed on-the-fly:
  - $N_{\Delta t} = 5{,}000$ OFDM symbols per time step Δt



Synchronization Problems [4]

Problem: SFO between Transmitter and Receiver:

- Problem for block-based system $\rightarrow$ messages skipped or repeated $\rightarrow$ insertion and deletion channel
- How to synchronize an AE symbol stream?

[4] S. Dörner et al. "Deep Learning-based Communication Over the Air", IEEE J-STSP, 2018

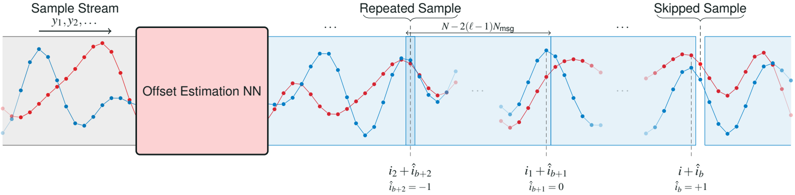
Synchronization Problems [4]

Problem: SFO between Transmitter and Receiver:

- Problem for block-based system → messages skipped or repeated → insertion and deletion channel
- How to synchronize an AE symbol stream?

Solution 1: Assisting Offset Estimation Neural Network

- Additional NN can skip or repeat samples



[4] S. Dörner et al. "Deep Learning-based Communication Over the Air", IEEE J-STSP, 2018

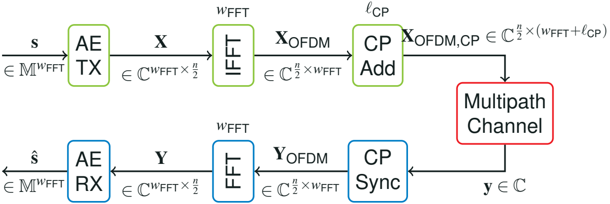
Synchronization Problems [4]

Problem: SFO between Transmitter and Receiver:

- Problem for block-based system $\rightarrow$ messages skipped or repeated $\rightarrow$ insertion and deletion channel
- How to synchronize an AE symbol stream?

Solution 2: OFDM with cyclic prefix (CP)

- CP allows easy synchronization

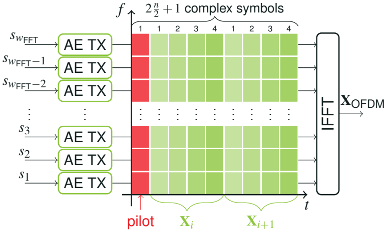


[4] S. Dörner et al. "Deep Learning-based Communication Over the Air", IEEE J-STSP, 2018

OFDM-Extensions [5]

Several advantages:

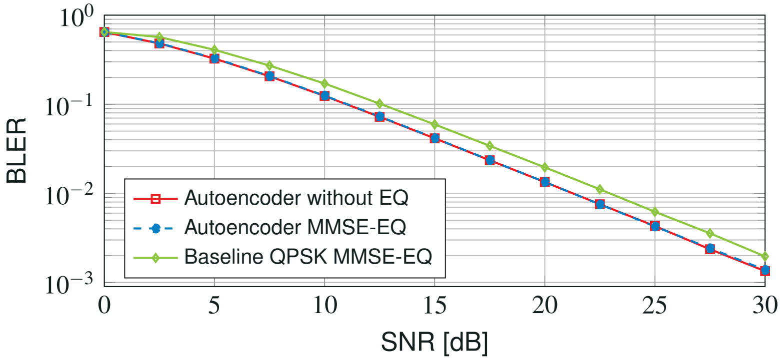
- Robustness against sampling synchronization errors
- Single-tap equalization
- Moderate training complexity (independent sub-carrier modulation)
- Can be embedded in exiting AE setup
- Full compatibility with existing schemes



[5] A. Felix et al. "OFDM-Autoencoder for End-to-End Learning of Communications Systems", SPAWC, 2018

Agenda

Simulated Performance

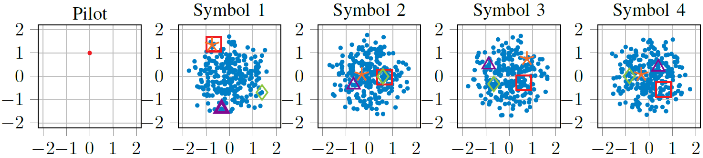


Transmission over 5-tap multi-path channel

- 1 pilot symbol per 8 OFDM symbols
- Autoencoder outperforms QPSK baseline
- Autoencoder *learns* MMSE equalization

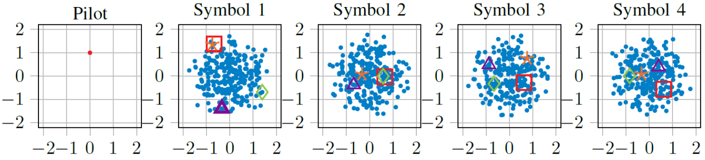
Learned IQ-Symbol Constellations

Average Power Normalization (with explicit pilot):

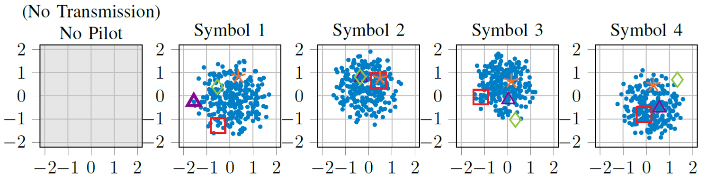


Learned IQ-Symbol Constellations

Average Power Normalization (with explicit pilot):

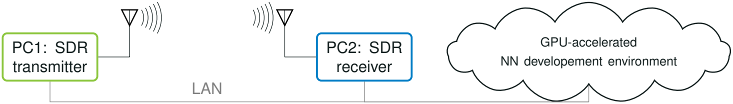


Average Power Normalization (without explicit pilot):



- $\rightarrow$ superimposed pilot

Measurement Setup



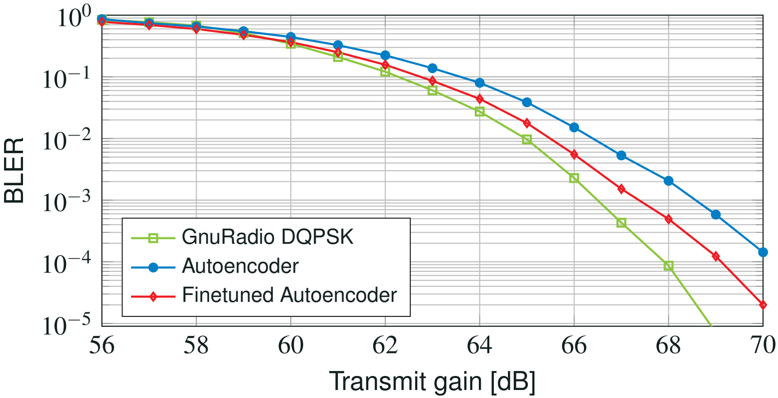
Transmitter and Receiver PCs

- Using USRP software defined radios with GnuRadio
- Connected via LAN to development server

Measurement Process

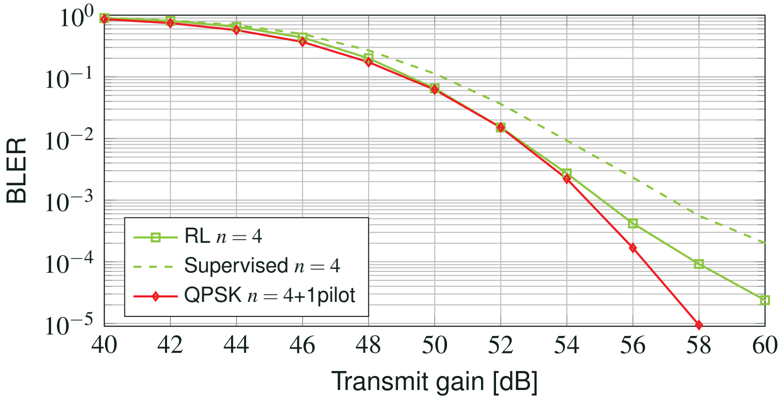
- Transmission of long message sequence by transmitter
- Recording of whole sequence by receiver
- Offline decoding by GPU-accelerated server

Over-the-Air Measurement Results



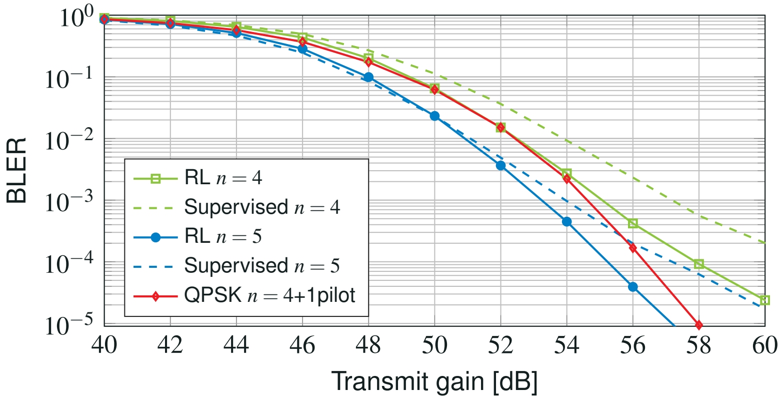
- 46m LOS channel, same information rate and same conditions for GnuRadio DQPSK and AE system
- The setup uses a single-carrier system (no OFDM)

Reinforcement Learning Over-the-Air



- 5m LOS channel, same information rate and same conditions for QPSK and AE system
- OFDM-based system

Reinforcement Learning Over-the-Air



- 5m LOS channel, same information rate and same conditions for QPSK and AE system
- OFDM-based system

Agenda

Conclusion

- World's first over-the-air communication system where signal processing is solely based on neural networks
- Proof of concept as SDR implementation
- Major challenges:
  - Unknown channel gradient → RX finetuning or RL training
  - Training complexity: enhanced network structure
  - How to synchronize? → OFDM-extensions
- Future steps:
  - Theoretical analysis: how *good* is the system really? Spectral efficiency?

Conclusion

- World's first over-the-air communication system where signal processing is solely based on neural networks
- Proof of concept as SDR implementation
- Major challenges:
  - Unknown channel gradient $\rightarrow$ RX finetuning or RL training
  - Training complexity: enhanced network structure
  - How to synchronize? $\rightarrow$ OFDM-extensions
- Future steps:
  - Theoretical analysis: how *good* is the system really? Spectral efficiency?

Conclusion

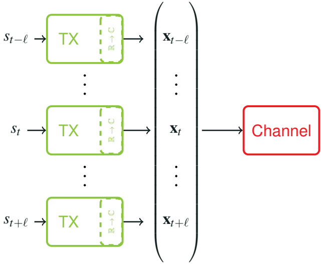
- World's first over-the-air communication system where signal processing is solely based on neural networks
- Proof of concept as SDR implementation
- Major challenges:
  - Unknown channel gradient $\rightarrow$ RX finetuning or RL training
  - Training complexity: enhanced network structure
  - How to synchronize? $\rightarrow$ OFDM-extensions
- Future steps:
  - Theoretical analysis: how *good* is the system really? Spectral efficiency?

Thank you for your attention!

For details:

- S. Dörner, S. Cammerer, J. Hoydis, S. ten Brink, "Deep Learning-based Communication Over the Air", arXiv:1707.03384, 2017
- Google Colab Notebook - Autoencoders - Learning to communicate

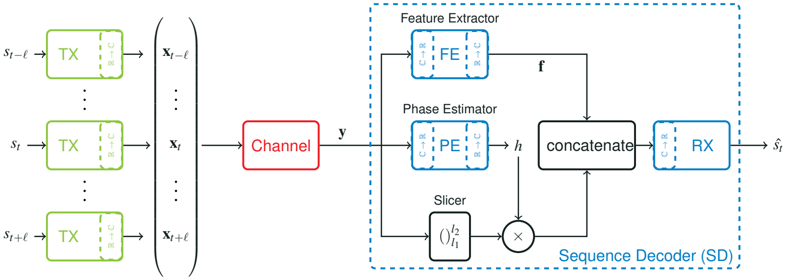
Upgrade to Sequence Autoencoder



Temporal Extension

- Problem: temporal dependencies (message s_t influenced by messages s_{t-1} and s_{t+1})
- Solution: multiple transmitter outputs concatenated

Upgrade to Sequence Autoencoder



Receiver Extension

- FE: extract features on sequence including multiple messages
- PE: estimate phase on sequence including multiple messages
- expert knowledge: complex phase shift by estimated phase h
- $\rightarrow$ same concept but more *sophisticated* receiver-NN

Dealing with Sample Frequency Offset

Problem: SFO between Transmitter and Receiver:

- more or less samples recorded than transmitted
- problem for block-based system → messages skipped or repeated → insertion and deletion channel

Solution: Assisting Offset Estimation Neural Network

- can skip or repeat samples

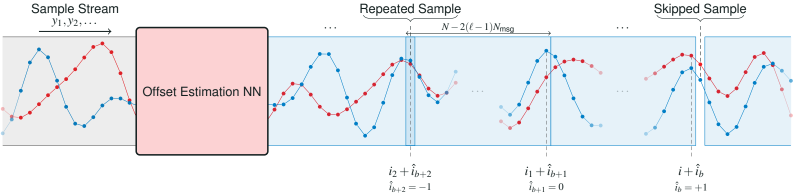
Dealing with Sample Frequency Offset

Problem: SFO between Transmitter and Receiver:

- more or less samples recorded than transmitted
- problem for block-based system → messages skipped or repeated → insertion and deletion channel

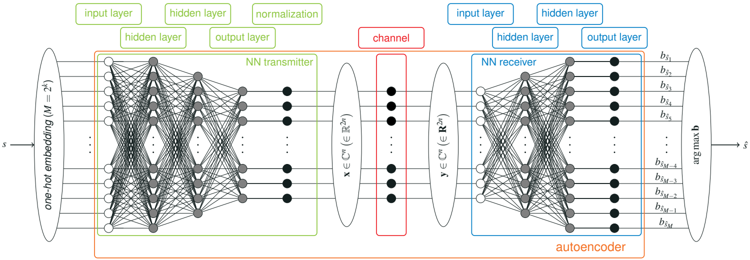
Solution: Assisting Offset Estimation Neural Network

- can skip or repeat samples



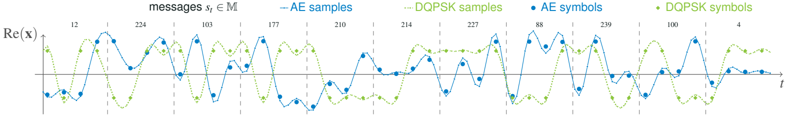
University of Stuttgart
Institute of Telecommunications

Autoencoder NN Layer Structure

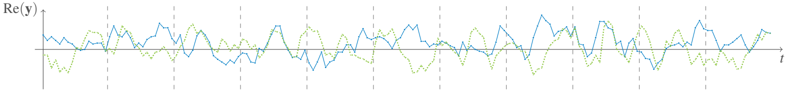


Physical Layer Signals

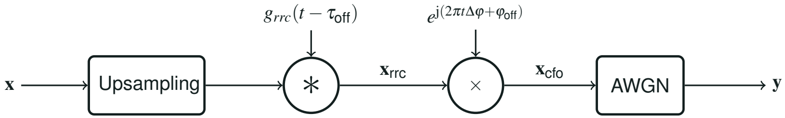
Sent Sample Sequence:



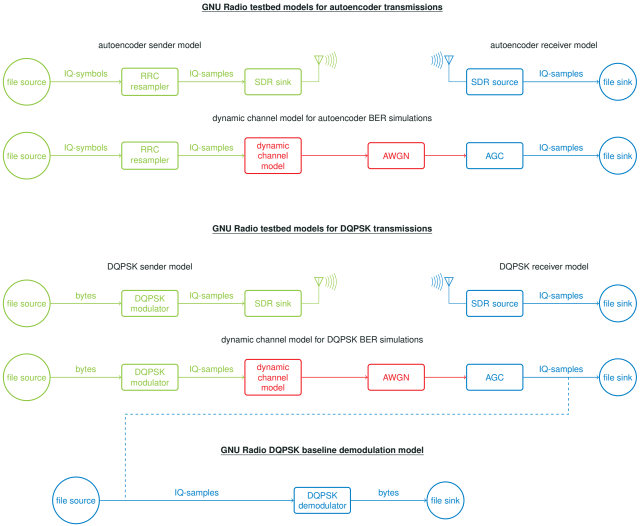
Received Sample Sequence:



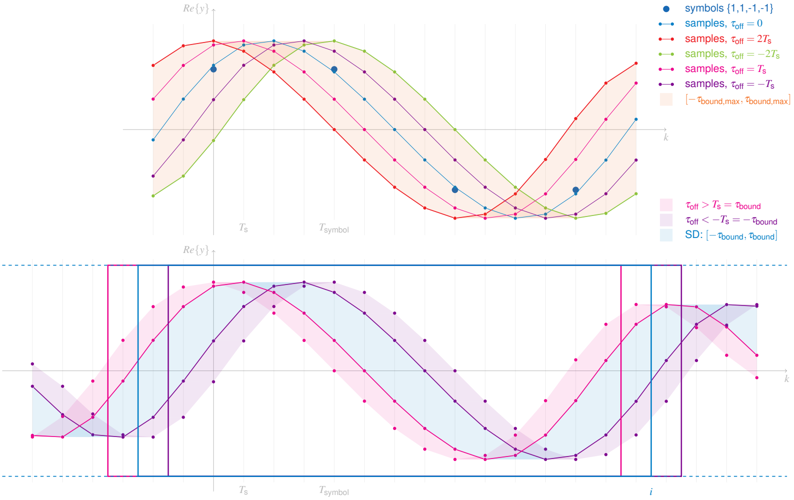
Synthetic LOS Channel Model



GnuRadio Measurement Setup



Sample Frequency Offset Impact



OE's decision impact to an exemplary window slice of length $N_{\text{seq}} = 17$:

skip offset $\hat{i} = 1$ | **no offset** $\hat{i} = 0$ | **repeat sample** $\hat{i} = -1$

Development Server Setup