Deep Learning for the Gaussian Wiretap Channel

- Writers: By Rick Fritschek, Rafael F. Schaefery, and Gerhard Wunder
- Elham Jafari

Contents

1. Introduction
2. Related work
3. Physical Layer Security in Wiretap Channels
4. Neural Network Implementation
5. Training Phases and simulation results
6. Conclusions And Outlook

- Neural networks optimize encoding and decoding functions for reliable message transmission.
- This approach is extended to communication scenarios where eavesdroppers must remain unaware.
- A modified secure loss function based on cross-entropy is used for transmission secrecy.
- The secure loss function approach is applied in a Gaussian wiretap channel setup.
- The neural network learns a trade-off between reliable communication and information secrecy by clustering learned constellations.
- This results in eavesdroppers with higher noise unable to distinguish between symbols.

This work focuses on improving the additive white Gaussian noise channel (AWGN) in communication systems using deep learning and physical layer security approaches.

The authors demonstrate that neural networks can learn encoding and decoding functions without extensive theoretical analysis, enabling on-the-fly system adaptation to new channel scenarios.

They also demonstrate the potential of creating a training environment where two NN decoders compete against each other

Related work

Autoencoder NN concept is used to model communication scenarios, but its drawback is the need for a differentiable channel model.

The learned encoding and decoding rules provide a system similar to classical schemes and perform well on real-world over-the-air transmissions.

Reinforcement learning can be used without a mathematical channel model, demonstrating end-to-end learning of communication systems.

The concept can also be used to learn advanced communication schemes like orthogonal frequency division multiplexing (OFDM), enabling reliable transmission in multi-path channel scenarios.

Physical Layer Security in Wiretap Channels

Wiretap Channel

- A three-node network where a sender transmits confidential information to a legitimate receiver while keeping an external eavesdropper ignorant.

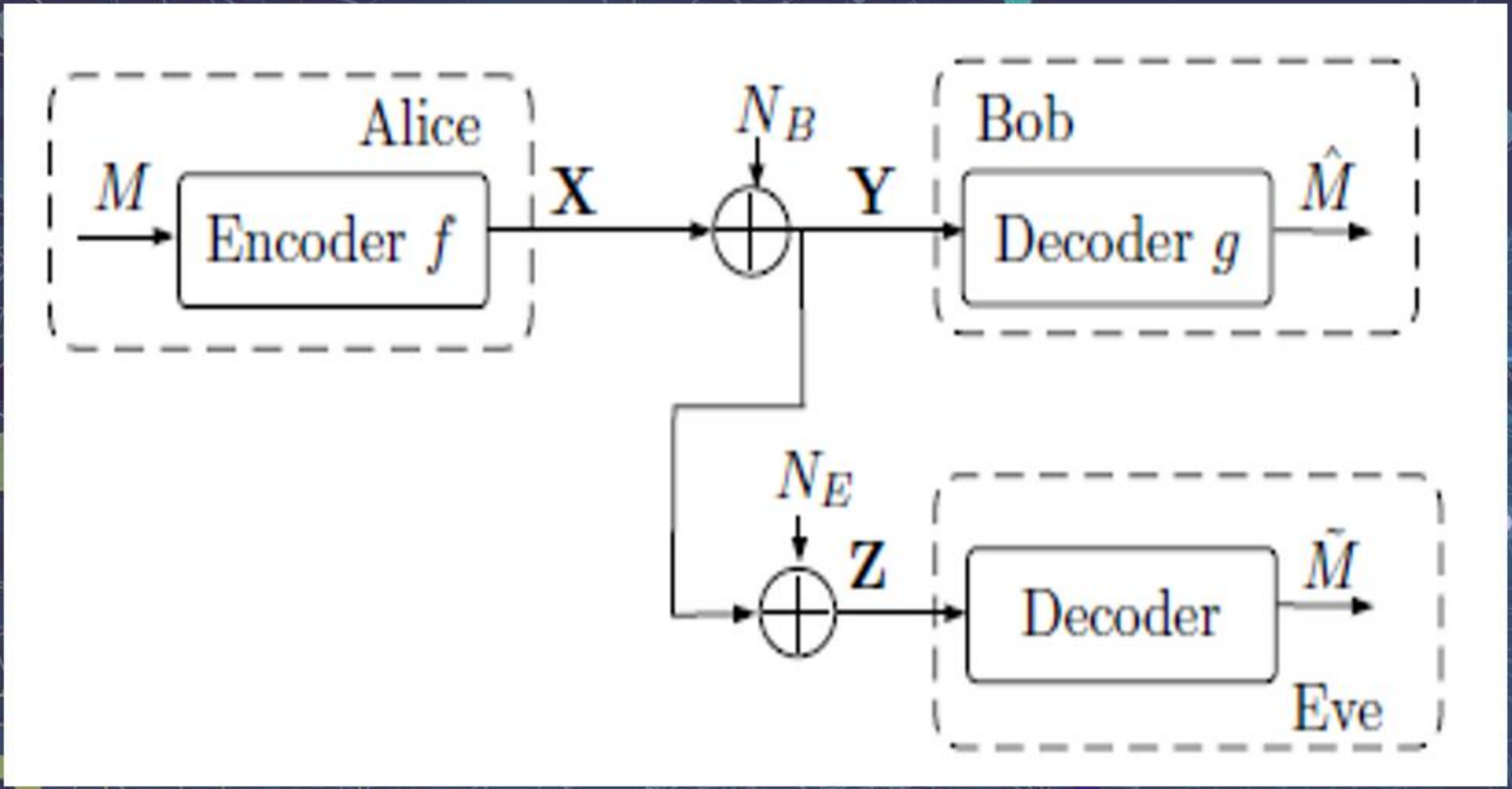
- The wiretap channel is the simplest communication scenario involving both reliable transmission and secrecy.

- The paper studies the degraded Gaussian wiretap channel.

Physical Layer Security in Wiretap Channels

Communication Task

- Alice encodes a message into a codeword of block length n.
- Receiver Bob decodes its received channel output to Receiver Eve.
- Secrecy of the transmitted message is ensured and measured by information theoretic concepts.
- The output on the wiretap should be independent of the message sent, which means that no confidential information is leaked to the wiretap.



Degraded Gaussian wiretap channel.
The confidential communication is between Alice and Bob, while Eve tries to eavesdrop upon it.

Neural Network Implementation

1. General Model

- The communication scenario is implemented using an autoencoder-like network setting.
- The autoencoder consists of an encoder mapping the input to codewords and a decoder estimating the input from the output.
- The encoding function performs dimensionality reduction to learn useful properties of the data for reconstruction.
- The NN learns to represent the input in a higher dimensional space to combat noise corruption

Neural Network Implementation

1. Message Encoding

- The message is one-hot encoded into a binary vector, which can be viewed as a probability distribution.
- The encoder is composed of two fully connected/dense layers, with the first mapping to with a ReLU activation function, and the second mapping the input with no activation function.
- The last layer is a normalization, where the codewords get a unit power constraint

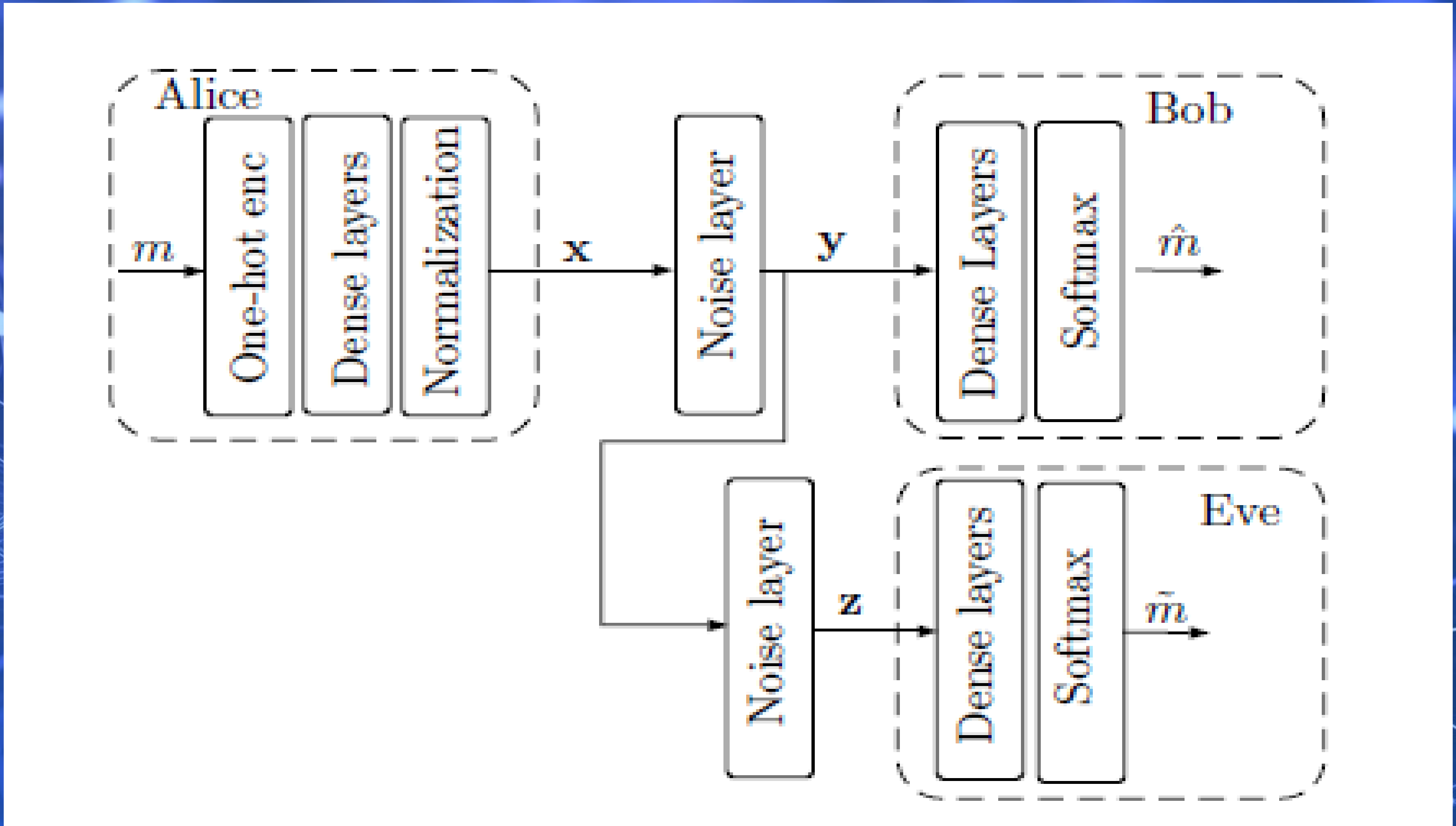
Neural Network Implementation

3. Receiver Blocks

- Two receiver blocks are equally constructed with two fully connected layers.

- The first maps the channel output back with a ReLU activation function and the second maps with a linear activation function.

- The loss function is used as a maximum likelihood estimation of the send signal.



Neural network implementation of the degraded wiretap channel.

Neural Network Implementation

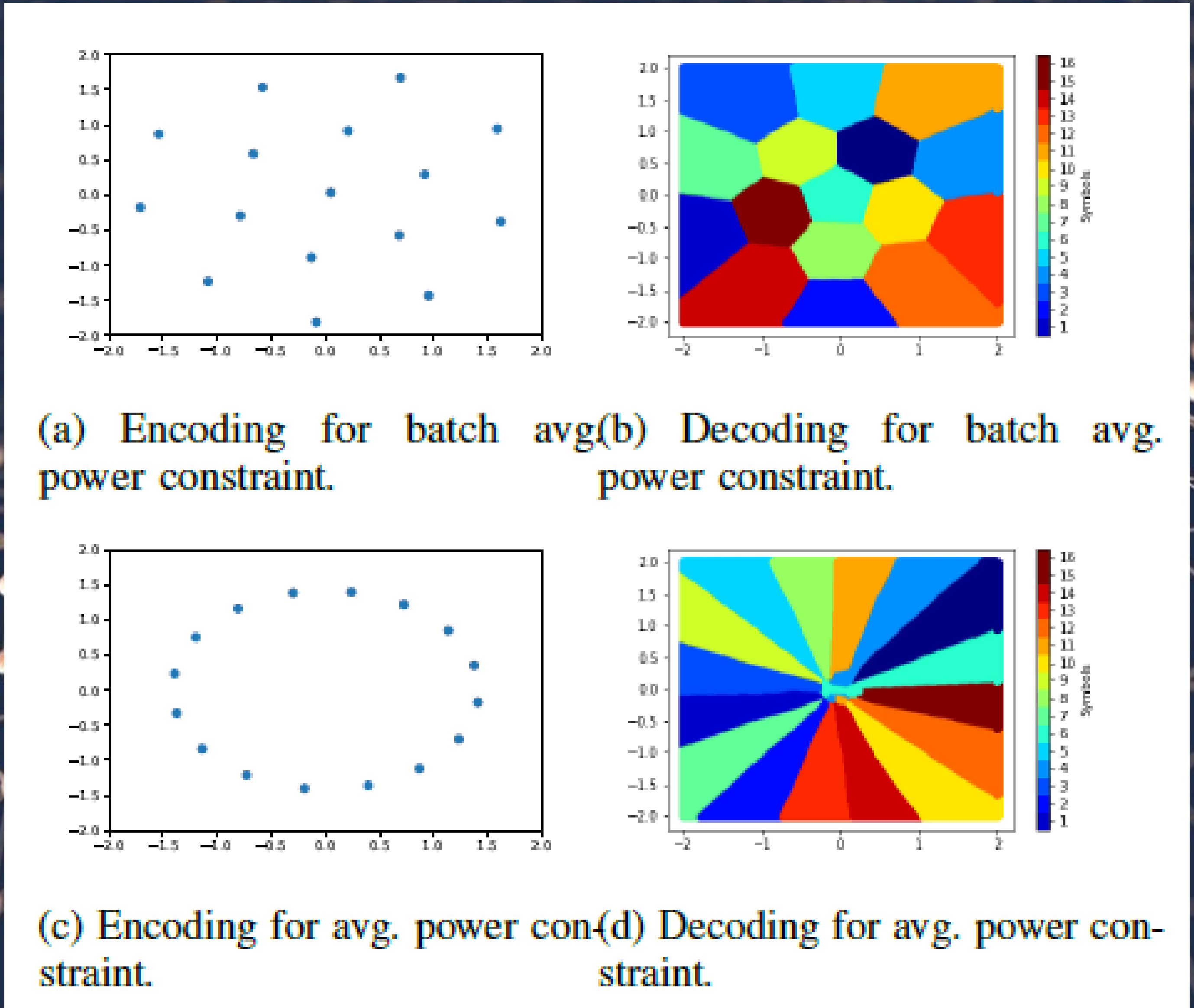
Mimicry of Traditional Wiretap Coding Techniques

- ❖ Theoretically, the usual approach is to map a specific message to a bin of codewords and then select a codeword randomly from this bin.
- ❖ There, intuitively, the randomness is used to confuse the eavesdropper.

The idea is that Eve can only distinguish between clusters of codewords.

Whereas the messages itself are hidden randomly in each cluster.

However, the legitimate receiver has a better channel and can also distinguish between codewords inside the clusters. A modified k-means algorithm, which gives equal cluster sizes for clustering constellation points, is also used.



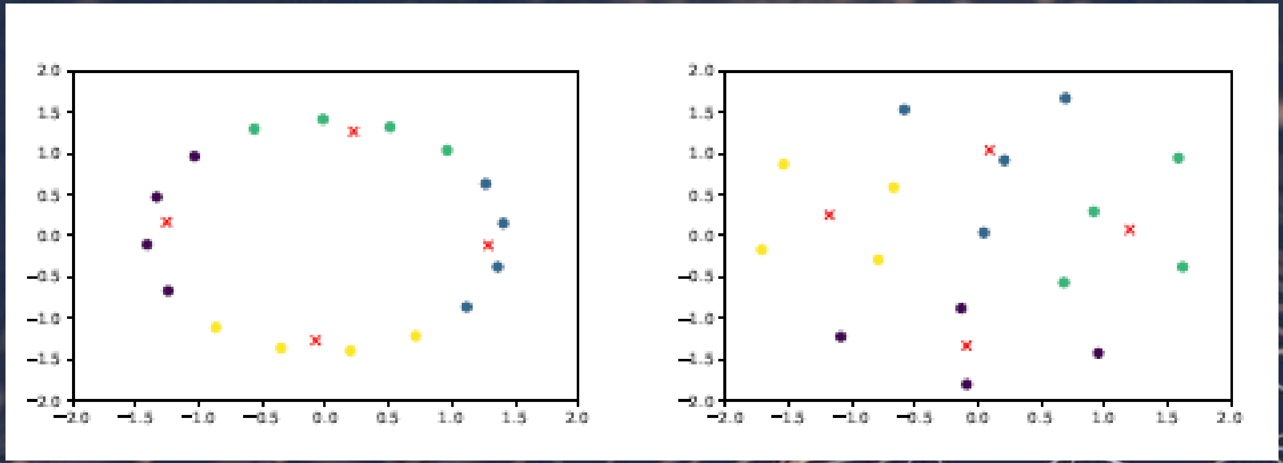
(a) Encoding for batch avg power constraint.

(b) Decoding for batch avg. power constraint.

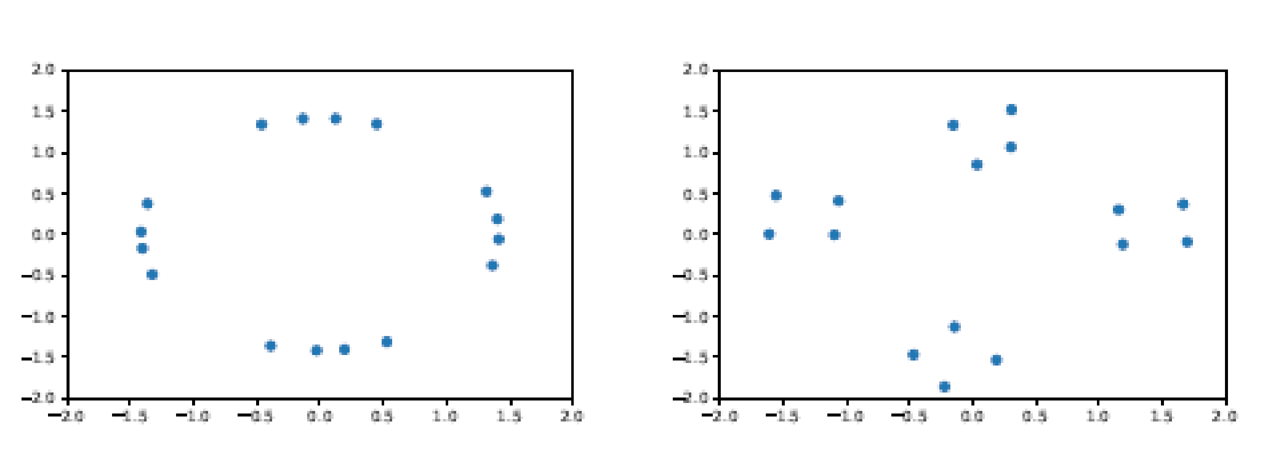
(c) Encoding for avg. power constraint.

(d) Decoding for avg. power constraint.

The figure shows the learned encoder mappings and decoder decision regions of 16 symbols for a batch average power constraint and an average power constraint on the symbols.



The figure shows the clusters for batch average power norm on the right hand side and for average power constraint per symbol on the left hand side. The red crosses show the cluster centers of the k-means algorithm.



The figure shows the learned secure constellations for the decoder with batch average power norm on the right hand side and for average power constraint per symbol on the left hand side.

Training Phases and simulation results

- TensorFlow simulations were conducted using the Adam optimizer and a learning rate from 0:1 to 0:001. Batch size was increased from 25 to 300 during epochs.

- Training was done with a channel layer of the direct link with an SNR of 12 dB and on Eve's link with an SNR of 5 dB.

- The training procedure was divided into four phases:

Training Phases and simulation results

1. Train the encoder and decoder of Bob with the standard cross-entropy.

2. Train Eve on decoding the previously learned encoding scheme with her cross-entropy and the normal input distribution.

3. Freeze the decoding layers of Eve and train the NN with the loss function (10) and � = 0:7.

❖ The training effect is that the NN tries to pull codewords from the same cluster together, close enough that Eve cannot distinguish the symbols in cluster, and loose enough that Bob can still decode them.

Training Phases and simulation results

After the secure training phase, Bob’s and Eve’s decoder were trained again to decode the new secure encoded signals.

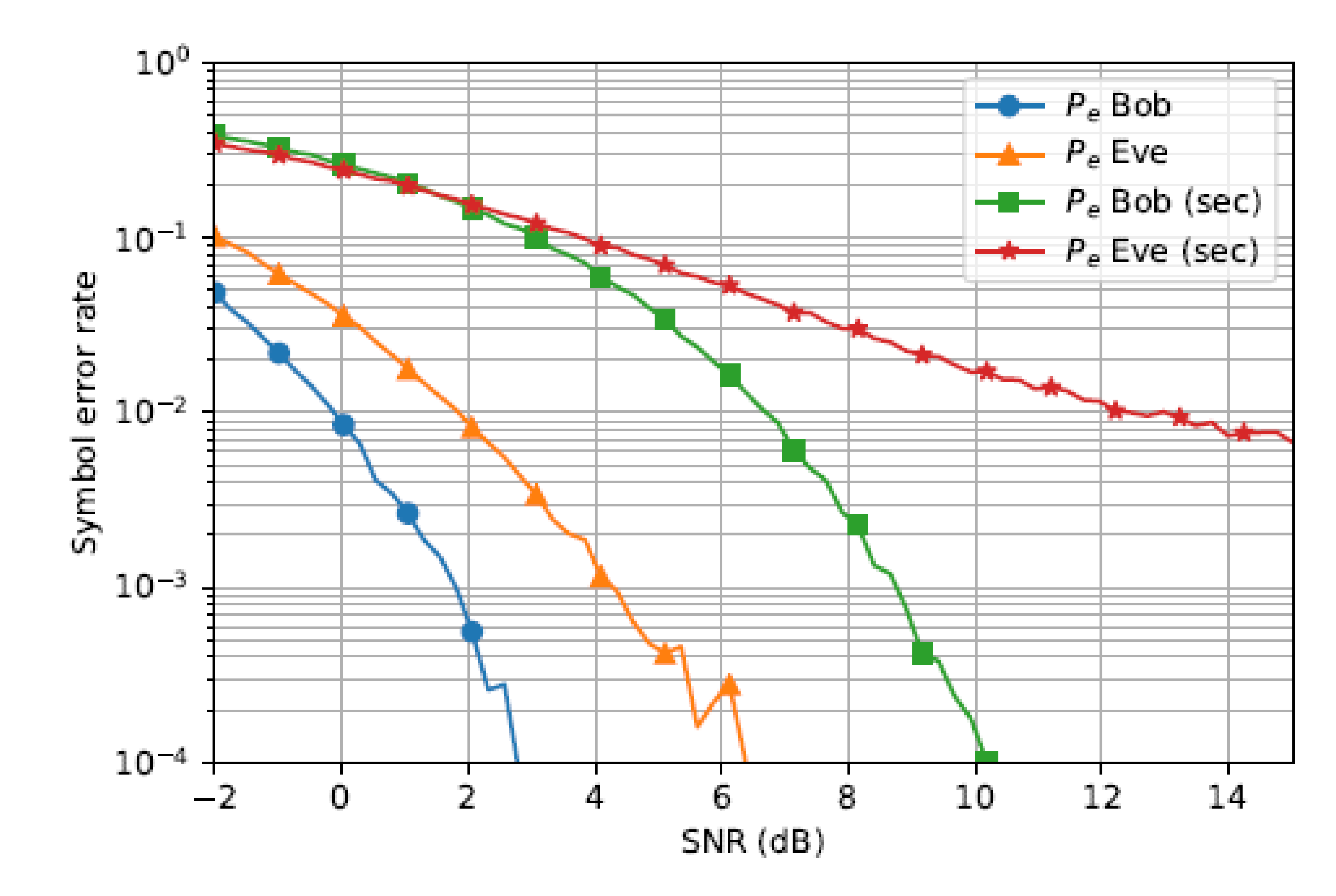
- The training phase for 16 symbols gets more accurate and faster with increasing codeword dimensions n, suggesting that the NN can find better constellations.

- A conservative approach was taken, resulting in n = 32.

- A coset coding algorithm was implemented, using 4 clusters each containing 4 symbols.

- The NN learns a constellation which can be seen as a finite lattice-like structure, on which one can implement the idea of coset coding.

- The actual simulation took a direct SNR of 10 dB and an additional SNR of 7 dB in the adversary link.



The figure shows the symbol error rate to SNR graph for a 16 symbol constellation size with n = 32 channel uses and fixed additional SNR of 17 dB in Eve's channel, before and after secure encoding. Note that the SNR is per symbol and not per bit (EB=N0).

CONCLUSIONS AND OUTLOOK

The study demonstrates that autoencoder neural networks can be used for secure communication scenarios by learning a finite constellation/lattice clustering.

This opens up research in secure communication, as classical secure coding schemes can be applied with a neural network.

However, direct optimization via mutual information terms remains an open problem.