Deep Learning Based Wiretap Coding via Mutual Information Estimation

Rick Fritschek
Heisenberg Communications and Information Theory Group
Freie Universität Berlin
Berlin, Germany
rick.fritschek@fu-berlin.de

Rafael F. Schaefer
Information Theory and Applications Chair
Technische Universität Berlin
Berlin, Germany
rafael.schaefer@tu-berlin.de

Gerhard Wunder
Heisenberg Communications and Information Theory Group
Freie Universität Berlin
Berlin, Germany
g.wunder@fu-berlin.de

ABSTRACT

Recently, deep learning of encoding and decoding functions for wireless communication has emerged as a promising research direction and gained considerable interest due to its impressive results. A specific direction in this growing field are neural network-aided techniques that work without a fixed channel model. These approaches utilize generative adversarial networks, reinforcement learning, or mutual information estimation to overcome the need of a known channel model for training. This paper focuses on the last approach and extend it to secure channel coding schemes by sampling the legitimate channel and additionally introduce security constraints for communication. This results in a mixed optimization between the mutual information estimate, the reliability of the code and its secrecy constraint. It is believed that this lays the foundation for flexible, generalizable physical layer security approaches due to its independence of specific model assumptions.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Mathematics of computing** → **Information theory**; • **Security and privacy** → **Formal methods and theory of security**.

KEYWORDS

Mutual information estimation, secure encoding, physical layer security, deep learning, autoencoder.

ACM Reference Format:
Rick Fritschek, Rafael F. Schaefer, and Gerhard Wunder. 2020. Deep Learning Based Wiretap Coding via Mutual Information Estimation. In *ACM Workshop on Wireless Security and Machine Learning (WiseML '20), July 13, 2020, Linz (Virtual Event), Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3395352.3402654

WiseML '20, July 13, 2020, Linz (Virtual Event), Austria

ACM ISBN 978-1-4503-8007-2/20/07…$15.00
https://doi.org/10.1145/3395352.3402654

1 INTRODUCTION

Machine learning and in particular neural network (NN) based approaches, so-called deep learning methods, have made a considerable impact in the most recent research for wireless communication. A recent branch of this new direction are end-to-end learning approaches, where autoencoders are used to simultaneously learn appropriate encoding and decoding functions to optimally transmit messages over a noisy channel [28]. This approach was shown to yield results, i.e., neural networks for encoding and decoding, which perform close to baseline techniques [9]. The autoencoder approaches usually involve optimizations over a minimum squared error term or a cross-entropy loss term with variants of stochastic gradient decent.

Recently, it was shown that these autoencoder approaches can also be utilized to learn secure encoding functions by including a secrecy constraint into the corresponding optimization. Here, of particular interest is the information leakage, i.e., the information about the confidential message that is leaked to an eavesdropper, and the most promising approach would be to compute a tight upper bound on the leakage and use this bound as a regularization term in the standard loss function expression. However, this approach is in general ambitious because normally one has only access to samples of the channels and, accordingly, the leakage can only be approximated for certain scenarios. One of these scenarios was investigated in [5], where the information leakage was approximated for a Gaussian mixture model. In [37], neural networks were utilized to learn an appropriate input covariance matrix for precoding for the MIMO Gaussian wiretap channel, which is faster than traditional methods. In [23], an adversarially trained deep joint source-channel coding approach to secure communication is proposed. In this paper, we utilize another approach to provide security by introducing a fake clustered distribution together with a structure enforcing cross-entropy loss. This cross-entropy loss will impose a clustering in the transmit constellations and, accordingly, will imitate the classical co-set coding approach for security. Using this method, it was shown in [14], that a secure encoding can be learned within the autoencoder framework.

However, as previously mentioned, the autoencoder approaches use variants of stochastic gradient decent. To optimize all of the layers, backpropagation is utilized through the whole communication chain, and therefore also through the channel itself. For that, the channel also needs to be modelled as a layer within the autoencoder structure. This is also the main disadvantage of the autoencoder approaches and there are three recent advances to dissolve this

issue: *i) Generative adversarial networks (GANs), ii) Reinforcement learning*, and *iii) Mutual information estimators*.

i) Generative adversarial networks (GANs): Introduced in [18] to learn the channel distribution from samples of the real channel. GANs consist of two NNs: A generator NN that tries to generate a fake distribution from a uniform input as close as possible to the target distribution and a discriminator NN that tries to distinguish between real and fake channel samples. The goal of the generator is to fool the discriminator and both NNs are alternatingly optimized [29, 36]. *ii) Reinforcement learning*: Used to circumvent the backpropagation mechanism with a feedback loop. Here, the communication system can be seen as an agent taking actions in a certain state and environment and receiving rewards for these actions, i.e., a low loss in a certain communication metric. Based on this, the system then learns an optimal policy for future actions, that maximizes these rewards. This idea was introduced in [2], and subsequently extended to noisy feedback links in [19]. The disadvantage of using model-free reinforcement learning is that the approach needs large sample sizes to accurately learn an optimal policy, because the system cannot utilize known structure and side-information.

iii) Mutual information estimators: In this work, we consider the last approach of mutual information estimators. Here, based on NNs as in [4] are used to estimate the mutual information between input and output samples of the wireless channel. The resulting estimate can then be used to train the encoder NN, by maximizing the output value of the then fixed estimator. This approach has the advantage that it is based on information theoretic foundations knowing that an optimal coding strategy will maximize the mutual information. This idea was introduced in [13] and further discussed in [15].

The aim of this paper is now to merge the mutual information estimation-based approach for NN-based message encoding taking the security enabling clustered constellation approach into account. We will show, that the encoder can learn a secure encoding scheme based on a mixed loss function that consists of the approximate mutual information of the legitimate channel, learned from samples, and the cross-entropy function from the clustering enforcing the fake distribution.

2 WIRETAP CHANNEL

In this work, we consider the wiretap channel, which is a three-node network as shown in Fig. 1. It consists of a sender (Alice) which transmits confidential information to a legitimate receiver (Bob) while keeping an external eavesdropper (Eve) ignorant of it. This setup can be seen as the simplest communication scenario that involves both tasks of reliable transmission and secrecy. Accordingly, this is the crucial building block of secure communication to be understood for secure communication in more complex communication systems.

In the following we study the Gaussian wiretap channel. The legitimate channel between Alice and Bob is given by an additive white Gaussian noise (AWGN) channel as

$$Y_i = X_i + N_{B,i} \tag{1}$$

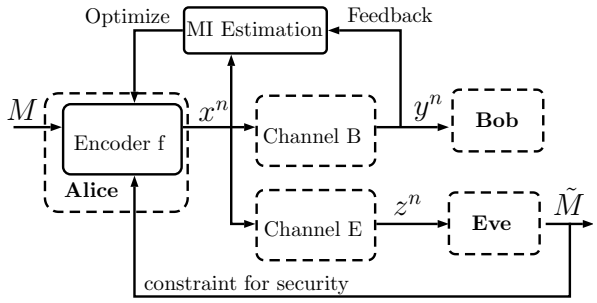


Figure 1: Gaussian wiretap channel. The encoder f of Alice is trained to enable secure communication to Bob. For that channel B (to Bob) is sampled and the encoder f is trained to maximize the estimated mutual information, which itself is learned from the channel B samples. Moreover, a security constraint is included by invoking an exemplary eavesdropper Eve.

where Y_i is the received channel output at Bob, X_i is the channel input of Alice, and $N_{B,i} \sim \mathcal{CN}(0, \sigma_B^2)$ is the additive complex circular symmetric distributed Gaussian noise at Bob at time instant i. The eavesdropper channel to Eve is accordingly given by

$$Z_i = X_i + N_{E,i} \tag{2}$$

where Z_i is the received channel output at Eve and $N_{E,i} \sim \mathcal{CN}(0, \sigma_E^2)$ is the additive complex circular symmetric distributed Gaussian noise at Eve. We further assume $\sigma_E^2 > \sigma_B^2$ which immediately results in a degraded wiretap channel for which the eavesdropper channel output Z_i is strictly worse than the legitimate channel output Y_i. Note that any Gaussian wiretap channel of the more general form $Y_i = h_B X_i + N_{B,i}$ and $Z_i = h_E X_i + N_{E,i}$ with h_B and h_E multiplicative channel gains can be transformed into an equivalent wiretap channel as in (1)-(2). This means that any Gaussian wiretap channel is inherently degraded, cf. for example [6, Sec. 5.1].

The communication task is now as follows: To transmit a message $m \in \mathcal{M} = \{1, ..., |\mathcal{M}|\}$, Alice encodes it into a codeword $x^n(m) = f(m)$ of block length n, where $x^n \in \mathcal{X}^n$ (usually, $\mathcal{X} = \mathbb{C}$ in the complex Gaussian setting). Moreover, we assume an average transmit power constraint $\sum_{i=1}^n |x_i|^2(m) \leq nP$. At the receiver side, Bob obtains an estimate $\hat{m}$ of the transmitted message by decoding its received channel output as $\hat{m} = g(y^n)$. The transmission rate of this code is then given by $R = \log |\mathcal{M}|/n$.

The secrecy of the transmitted message is ensured and measured by information theoretic concepts. There are different criteria of information theoretic secrecy including weak secrecy [35] and strong secrecy [24]. In the end, all criteria have in common that the output at the eavesdropper Z^n should become statistically independent of the transmitted message M implying that no information is leaked to the eavesdropper. For example, strong secrecy is defined as

$$\lim_{n\to\infty} I(M; Z^n) = 0 \tag{3}$$

with $I(M; Z^n) = \sum_{m,z^n}(m, z^n) \log \frac{p(m,z^n)}{p(m)p(z^n)}$ the mutual information between M and Z^n, cf. for example [7].

The secrecy capacity now characterizes the maximum transmission rate R at which Bob can reliably decode the transmitted message while Eve is kept in the dark, i.e., the secrecy criterion (3) is satisfied while achieving a vanishing error probability $P_e = \Pr[M \neq \hat{M}] \to 0$ as $n \to \infty$. The secrecy capacity of the Gaussian wiretap channel is known [6, 22] and is given by

$$\begin{aligned} C_s &= \max_{p(x)} \big[I(X;Y) - I(X;Z)\big] \\ &= \log\left(1 + \frac{P}{\sigma_B^2}\right) - \log\left(1 + \frac{P}{\sigma_E^2}\right). \end{aligned}$$

3 MUTUAL INFORMATION ESTIMATION WITH NEURAL NETWORKS

The computation and evaluation of the mutual information is a challenging task which recently gained interest in the machine learning community due to many applications ranging from representations learning to understanding neural networks. The challenge is mainly due to the fact, that in most applications, the underlying joint probability density is not available and sample-based approximations of the mutual information are needed. The main challenge here is to provide an accurate and stable approximation from low sample sizes of high-dimensional data sets. Common classical approaches are based for example on binning of the probability space [8, 11], k-nearest neighbor statistics [16, 17, 21], maximum likelihood estimation [32], and variational lower bounds [3]. In this work, we use a recently proposed estimator [4], coined *mutual information neural estimation (MINE)*, which utilizes the Donsker-Varadhan representation of the Kullback-Leibler divergence:

$$D_{KL}(P||Q) = \sup_{g:\Omega\to\mathbb{R}} \mathbb{E}_P[g(X,Y)] - \log(\mathbb{E}_Q[e^{g(X,Y)}]) \quad (4)$$

where the supremum is taken over all measurable functions g such that the expectation is finite. The right hand side of (4) yields a lower bound on the KL-divergence, which is tight for optimal functions. In [4], Belghazi *et al.* proposed to choose a neural network, parametrized with $\theta \in \Theta$ as function family $T_\theta : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ for the lower bound. Moreover, identifying P as $p(x,y)$ and Q as $p(x)p(y)$ yields the estimator

$$I(X;Y) \geq \sup_{\theta\in\Theta} \mathbb{E}_{p(x,y)}[T_\theta(X,Y)] - \log \mathbb{E}_{p(x)p(y)}[e^{T_\theta(X,Y)}]. \quad (5)$$

As we do not have access to the expected values, one needs to utilize Monte-Carlo sampling to approximate the values which yields

$$\tilde{I}_\theta(X;Y) := \frac{1}{k}\sum_{i=1}^{k}[T_\theta(x_{(i)}, y_{(i)})] - \log\frac{1}{k}\sum_{i=1}^{k}[e^{T_\theta(x_{(i)},\bar{y}_{(i)})}]. \quad (6)$$

Note that estimating the expectation of the last term leads to a biased estimate, due to the log function, which can be seen by applying Jensen's inequality, see [30] for further information. Due to this, (6) does not represent a valid lower bound on the mutual information anymore. However in our scenario, the estimate is good enough to train the encoder network. Another closely related estimator is based on f-divergence representations [25], which uses the Fenchel duality to bound the f-divergence from below as

$$D_f(P||Q) \geq \sup_{g:\Omega\to\mathbb{R}} \mathbb{E}_P[g(X,Y)] - \mathbb{E}_Q[f^*(g(X,Y))],$$

where the supremum is again over all measurable functions g. Moreover, [26] also proposed to choose a parameterized neural network for this function family and the conjugate dual function $f^* = \exp(x-1)$, to obtain a lower bound on the KL-divergence. This leads to the estimator

$$I(X;Y) \geq \sup_{\theta\in\Theta} \mathbb{E}_{p(x,y)}[T_\theta(X,Y)] - \mathbb{E}_{p(x)p(y)}[e^{T_\theta(X,Y)-1}]. \quad (7)$$

Both lower bounds share the same supremum, however, over the choice of functions T, (5) is closer to the supremum than (7), see [31]. The drawback of both estimators is that they have a large variance. Another approach is via replicates, proposed in [33], which yields a low variance estimate, but is unfortunately bounded by the log of the batch size. In [30], a linear interpolation between the critics of the f-divergence bound and the replicate bound was proposed, which results in a tunable parameter. However, this also increases the sampling and computational complexity, which is why we still use the estimator (6).

4 ENCODING PROCEDURE AND IMPLEMENTATION

The encoder of Alice is modelled as a neural network with one fully-connected hidden layer with an elu activation function, and a linear output layer. The encoder gets one-hot encoded messages, i.e., binary vectors $m^{|\mathcal{M}|} \in \mathbb{F}_2^{|\mathcal{M}|}$ of the form $(0, ..., 0, 1, 0, ..., 0)$ which have a one at the i-th position, representing the i-th message of $\mathcal{M} = \{1, ..., |\mathcal{M}|\}$. The output of the network is then normalized to have unit power and is shaped from $2n$ real values to n complex values or codewords x^n, which are sent over the legitimate channel. As in [13], we sample the sent signals x^n and the received signals y^n and feed the samples into a mutual information (MI) estimation network T_θ.

4.1 Mutual information estimation

The MI estimation network consists of two fully-connected hidden layers with 256 nodes in each layer and relu activation functions. Afterwards, the value of the estimate is calculated, using the MINE approach (6). For that we use the Nadam optimizer [10], which incorporates the Nesterov momentum into the widely used Adam optimizer [20]. This yields the following estimator for k samples

$$\tilde{I}_\theta(X^n;Y^n) := \frac{1}{k}\sum_{i=1}^{k}[T_\theta(x^n_{(i)}, y^n_{(i)})] - \log\frac{1}{k}\sum_{i=1}^{k}[e^{T_\theta(x^n_{(i)},\bar{y}^n_{(i)})}], \quad (8)$$

where the k samples of the joint distribution $p(x^n, y^n)$, for the first term in (8), are produced via uniform generation of messages m and sending them through the initialized encoder, which generates X^n of $p(x^n, y^n)$. The corresponding Y^n is generated by the channel. The marginal distributions can be produced for example by a reshuffling of corresponding pairs.

4.2 Training of the encoder network

To train the encoder, we use a signal-to-noise ratio (SNR) per bit of $E_b/N_0 = 7$ dB. This specifies our noise variance of the direct channel, from which we sample and the codewords are normalized to unity over the batch-size. Moreover, we assume an SNR per bit of $E_b/N_0 = 6$ dB for Eves channel, which corresponds approximately

to an $E_b/N_0 = 12$ dB additional noise factor on top of Bobs channel. The training of the encoder itself is divided into several phases. At first we train the MI estimation network T_θ for an initial round with 500 iterations and a batch size of 64. At this point, the parameters of the encoder NN are only randomly initialized and the estimated mutual information might not reflect the final estimated value. In the second phase, we alternate between maximizing (8) over the encoder weights ϕ and the estimator weights θ

$$\max_{\phi} \max_{\theta} \tilde{I}_\theta(X_\phi^n(m); Y^n).$$

The encoder weights ϕ are trained for 5 epochs, with 400 iterations and batch size of 500 with a learning rate of 0.005, and a second time with a learning rate of 0.0005. After every epoch, the estimator weights θ are trained for one iteration with batch size 250 and learning rate 0.05. Both optimizations are done using the Nadam optimizer.

4.3 Enforcing structure and security constraints on the encoder

We use the same approach to enforce the security as in [14]. For that, we introduce a structure enforcing (SE) decoder with the same noise parameter as Eve to apply the methods of [14] to our case. In this approach, we train the encoder network such that the SE decoder sees all codewords in a certain cluster with the same probability. This is to enforce a coset-like structure on the encoding function. In particular, the actual messages label the cosets, and the particular codeword inside the coset is again chosen at random. This is based on the coset coding method which goes back to the work of [35] and we refer the reader to [27, Appendix A] for an introduction. 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. Therefore, trading a part of the communication rate for the security constraint, one can achieve a secure communication. To enable this cluster structure in our NN encoding, we introduce a cross-entropy loss constraint for our SE decoder which is fed with a modified fake input distribution. This modification is such that clusters of codewords (calculated with the k-means method) have the same input probability. Normally, due to the one-hot encoding approach, a certain symbol has probability one if it was sent in the sample in the batch. Consider for example the training vector batch $m^4 = (1, 2, 3, 4)$, resulting in the one-hot data matrix

$$\mathrm{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the rows are the samples of the batch and the columns indicate the symbol. We now modify the true data matrix towards an equalized matrix $\bar{\mathrm{S}}$:

$$\bar{\mathrm{S}} = \mathrm{SE} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

where for example the first sample has an equal probability to be symbol 1 and symbol 2. The matrix $\mathbf{E}$ can be calculated with the k-means algorithm in conjunction with Algorithm 1 from [14]. The SE decoder cross-entropy loss can now be written as

$$L_{\mathrm{SE}} := H(p_{\overline{\mathrm{data}}}(M)), p_{\mathrm{SE}}(M)) = -\sum_{i=1}^{|\mathcal{M}|} \bar{s}_i \log \tilde{s}_i,$$

where the vectors $\tilde{s}^{|\mathcal{M}|}$ and $\bar{s}^{|\mathcal{M}|}$ can be interpreted as the decoded distribution and as the equalized input symbol distribution as both are normalized to one. This loss is used as our secrecy constraint. Together with the previous encoder loss function, we get our security enabled optimization problem:

$$\begin{aligned} \max_{\phi} \quad & (1-\alpha)\tilde{I}(X_\phi^n(m); Y^n) \\ & - \alpha H(p_{\overline{\mathrm{data}}}(M)), p_{\mathrm{SE},\phi}(M)). \end{aligned} \tag{9}$$

Note that in this formulation, the MI estimator is now fixed with a certain θ learned in the previous phases. Furthermore, the α parameter controls the trade-off between security and communication rate on the legitimate channel. For the security constraint, we need the SE decoder, which is implemented with a standard NN decoder with a hidden layer with elu activation and an output layer with softmax activation. The decoder will be pre-trained with a batch size of 200 and 400 iterations per epoch, for $\{2, 1, 1\}$ epochs with a learning rate of $\{0.005, 0.001, 0.0005\}$ with the Nadam optimizer. Afterwards, we initialize the matrix $\mathbf{E}$ and train the encoder, which will be optimized for secure encoding by maximizing (9) over 2 epochs, with 400 iterations, a learning rate of 0.005 and $\alpha = 0.4$. The simulation code will be made available at [12], implemented with TensorFlow 2.1 [1].

4.4 Evaluation

To evaluate the proposed method, we use standard cross-entropy based NN decoders, both constructed equally for Bob and Eve with a hidden layer with elu activation and an output layer with softmax activation. We train these decoders once after training of the encoder and before training for security. Bob and Eves decoders are trained with the same parameters, as the already mentioned routine for the SE decoder above. Then we test the system with 500000 samples for each E_b/N_0 data point. For these evaluations, we assume an additive fixed noise of $E_b/N_0 = 12$ dB (on top of Bobs noise) for Eves channel, to unify both results in one figure. In the second phase, we train both decoders once again after the encoder is trained for secure encoding. This time we train both decoders with 2 epochs, and 400 iterations per epoch with a learning rate of 0.005. Then we test the system again with 500000 samples for each E_b/N_0 data point. The Figure 2 shows the results of both test runs. One can see that before secure encoding, Bob and Eve achieve a low symbol error rate per batch, where Eves performance is worse due to the higher noise parameters of her channel. After secure encoding, Bobs symbol error rate per batch is increased due to the trade-off between security and communication rate, but still decreases with E_b/N_0. Eves performance on the other hand is capped at a certain minimum, providing security even for high E_b/N_0 values.

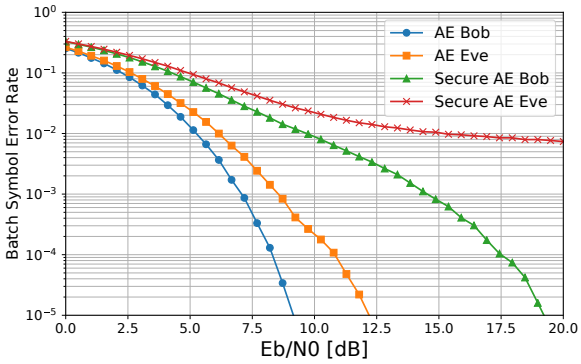
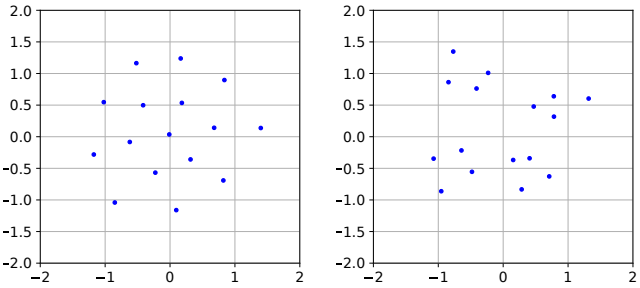
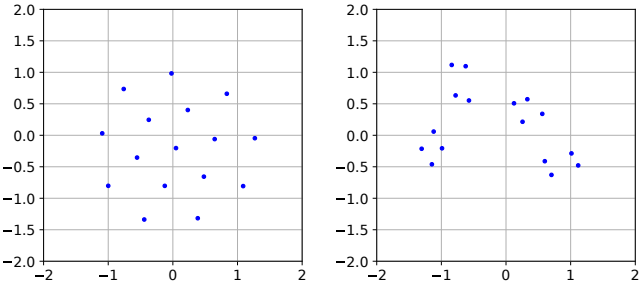


Figure 2: Evaluation of the proposed method for a 32-dimensional codeword constellation. AE Bob and AE Eve show the error rate for transmission of the codewords before secure encoding and, accordingly, Secure AE Bob and Secure AE Eve refer to after secure encoding.



(b) Standard constellation (c) Secure constellation

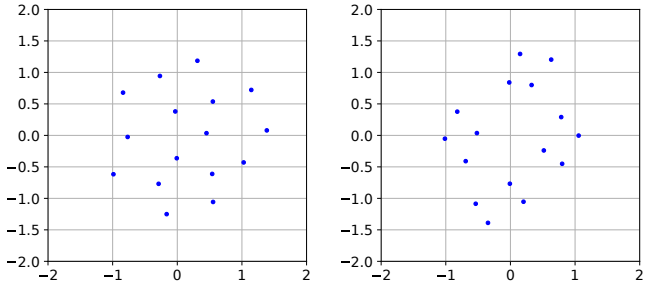
Figure 3: Constellations before and after secure encoding for 2 dimensions and 16 constellation points.



(b) Standard constellation (c) Secure constellation

Figure 4: Constellations before and after secure encoding for 32 dimensions and 16 constellation points. t-SNE was used, to represent the constellation in two dimensions. The left side uses a perplexity of 30, while the right hand side uses a perplexity of 5.

Moreover, to see if the constellation does indeed form clusters with our approach, we trained and evaluated the system for 2 dimensions, as shown in Figure 3. For that we slightly tuned the training learning rate in the second phase from $\{0.005, 0.0005\}$ to $\{0.05, 0.005\}$, respectively. Moreover, the SE decoder is only trained for 2 epochs, with 400 iterations and a learning rate of 0.005. At last, the security enabling training was trained for 1 epoch with 210 iterations and $\alpha = 0.45$. All other parameters are the same as in the 32 dimensional case. Here, it can be seen that the system forms clusters and that the structure enforcing method works as intended.



(b) Standard constellation (c) Secure constellation

Figure 5: Constellations before and after secure encoding for 32 dimensions and 16 constellation points. t-SNE was used, to represent the constellation in two dimensions with a perplexity factor of 8 for both constellations.

Another method to check the constellations in higher dimensions is to use dimensionality reduction techniques to represent the constellation in two dimensions and then visualize them. A particular useful technique to visualize high-dimensional data is the t-Distributed Stochastic Neighbor Embedding (t-SNE) [34]. Note however, that this technique strongly depends on its parameters. In our case, the crucial factor is the perplexity, which balances local and global aspects of the data, such as local clusters or global structure. A low perplexity is more focused on local clusters, as can be seen in Fig. 4 on the right hand side, while a high perplexity focuses on global structure, which can be seen on the left hand side. In Fig. 5, we chose a perplexity of 8 as a trade-of between local and global aspects of the data.

5 CONCLUSIONS

We have shown that a recently proposed approach which uses a NN based mutual information estimator to optimize encoding for channel transmission can be combined with another recent approach to introduce a fake distribution to enforce a co-set structure which enables secure transmission. This shows that channel knowledge is not necessary for secure encoding with neural networks as long as one has enough channel samples of the legitimate channel available and the channel to the eavesdropper is worse than the direct channel in terms of noise.

ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) under Grants FR 4209/1-1 and SCHA 1944/7-1.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Fayçal A. Aoudia and Jakob Hoydis. 2018. End-to-End Learning of Communications Systems Without a Channel Model. In *Proc. 52nd Asilomar Conf. Signals, Systems, and Computers*. Pacific Grove, CA, 298–303.

[3] David Barber and Felix V. Agakov. 2003. The IM Algorithm: A Variational Approach to Information Maximization. In *Proc. Adv. Int. Conf. Neural Inf. Process. Syst.* Vancouver and Whistler, Canada, 201–208.

[4] Mohamed I. Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual Information Neural Estimation. In *Proc. of the 35th Int. Conf. on Machine Learning*, Vol. 80. Stockholm, Sweden, 531–540.

[5] Karl-Ludwig Besser, Pin-Hsun Lin, Carsten R. Janda, and Eduard A. Jorswieck. 2020. Wiretap Code Design by Neural Network Autoencoders. *IEEE Trans. Inf. Forensics Security* 15 (2020), 3374 – 3386.

[6] Matthieu Bloch and João Barros. 2011. *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge University Press, Cambridge, UK.

[7] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory* (2 ed.). Wiley & Sons.

[8] Georges A. Darbellay and Igor Vajda. 1999. Estimation of the Information by an Adaptive Partitioning of the Observation Space. *IEEE Trans. Inf. Theory* 45, 4 (May 1999), 1315–1321.

[9] Sebastian Dörner, Sebastian Cammerer, Jakob Hoydis, and Stephan ten Brink. 2018. Deep Learning Based Communication Over the Air. *IEEE J. Sel. Topics Signal Process.* 12, 1 (Feb. 2018), 132–143.

[10] Timothy Dozat. 2016. Incorporating Nesterov Momentum into Adam. In *Proc. 4th Int. Conf. Learning Representations Workshops*. San Juan, Puerto Rico.

[11] Andrew M. Fraser and Harry L. Swinney. 1986. Independent Coordinates for Strange Attractors from Mutual Information. *Phys. Rev. A* 33, 2 (Feb. 1986), 1134.

[12] Rick Fritschek. 2020. Simulations WiseML (2020). https://github.com/Fritschek.

[13] Rick Fritschek, Rafael F. Schaefer, and Gerhard Wunder. 2019. Deep Learning for Channel Coding via Neural Mutual Information Estimation. In *Proc. 20th Int. Workshop Signal Process. Adv. Wireless Commun.* Cannes, France, 1–5.

[14] Rick Fritschek, Rafael F. Schaefer, and Gerhard Wunder. 2019. Deep Learning for the Gaussian Wiretap Channel. In *Proc. IEEE Int. Conf. Commun.* Shanghai, China, 1–6.

[15] Rick Fritschek, Rafael F. Schaefer, and Gerhard Wunder. 2020. Neural Mutual Information Estimation for Channel Coding: State-of-the-Art Estimators, Analysis, and Performance Comparison. In *Proc. 21th Int. Workshop Signal Process. Adv. Wireless Commun.* Atlanta, GA, USA.

[16] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. 2015. Efficient Estimation of Mutual Information for Strongly Dependent Variables. In *Proc. 18th Int. Conf. Artificial Intelligence and Statistics*. San Diego, 277–286.

[17] Weihao Gao, Sewoong Oh, and Pramod Viswanath. 2018. Demystifying Fixed k-Nearest Neighbor Information Estimators. *IEEE Trans. Inf. Theory* 64, 8 (Aug. 2018), 5629–5661.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proc. Adv. Neural Inf. Process. Syst.* Montréal, Canada, 2672–2680.

[19] Mathieu Goutay, Fayçal A. Aoudia, and Jakob Hoydis. 2018. Deep Reinforcement Learning Autoencoder with Noisy Feedback. *arXiv preprint arXiv:1810.05419* (Oct. 2018).

[20] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (Dec. 2014).

[21] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating Mutual Information. *Phys. Rev. E* 69, 6 (June 2004), 066138.

[22] S. Leung-Yan-Cheong and M.E. Hellman. 1978. The Gaussian Wire-tap Channel. *IEEE Trans. Inf. Theory* 24, 4 (July 1978), 451–456.

[23] Thomas Marchioro, Nicola Laurenti, and Deniz Gündüz. 2020. Adversarial Networks for Secure Wireless Communications. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.* Barcelona, Spain, 8748–8752.

[24] Ueli M. Maurer and Stefan Wolf. 2000. Information-Theoretic Key Agreement: From Weak to Strong Secrecy for Free. In *EUROCRYPT 2000, Lecture Notes in Computer Science*, Vol. 1807. Springer-Verlag, 351–368.

[25] XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. 2010. Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization. *IEEE Trans. Inf. Theory* 56, 11 (Nov. 2010), 5847–5861.

[26] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization. In *Proc. Adv. Neural Inf. Process. Syst.* Barcelona, Spain, 271–279.

[27] Frédérique Oggier, Patrick Solé, and Jean-Claude Belfiore. 2016. Lattice Codes for the Wiretap Gaussian Channel: Construction and Analysis. *IEEE Trans. Inf. Theory* 62, 10 (Oct. 2016), 5690–5708.

[28] Timothy O'Shea and Jakob Hoydis. 2017. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. on Cogn. Commun. Netw.* 3, 4 (Dec. 2017), 563–575.

[29] Timothy J. O'Shea, Tamoghna Roy, Nathan West, and Benjamin C. Hilburn. 2018. Physical Layer Communications System Design Over-the-Air Using Adversarial Networks. In *Proc. 26th European Signal Process. Conf.* Rome, Italy, 529–532.

[30] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. 2018. On Variational Lower Bounds of Mutual Information. In *Proc. Adv. Neural Inf. Process. Syst. Workshop on Bayesian Deep Learning*. Montréal, Canada.

[31] Avraham Ruderman, Mark Reid, Dario Garcia-Garcia, and James Petterson. 2012. Tighter Variational Representations of f-Divergences via Restriction to Probability Measures. *arXiv preprint arXiv:1206.4664* (June 2012).

[32] Taiji Suzuki, Masashi Sugiyama, Jun Sese, and Takafumi Kanamori. 2008. Approximating Mutual Information by Maximum Likelihood Density Ratio Estimation. In *New Challenges for Feature Selection in Data Mining and Knowledge Discovery*. Antwerp, Belgium, 5–20.

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748* (July 2018).

[34] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[35] Aaron D. Wyner. 1975. The Wire-Tap Channel. *Bell Syst. Tech. J.* 54 (Oct. 1975), 1355–1387.

[36] Hao Ye, Geoffrey Y. Li, Biing-Hwang F. Juang, and Kathiravetpillai Sivanesan. 2018. Channel Agnostic End-to-End Learning Based Communication Systems with Conditional GAN. In *Proc. IEEE Global Commun. Conf. Workshops*. Abu Dhabi, UAE, 1–5.

[37] Xinliang Zhang and Mojtaba Vaezi. 2019. Deep Learning based Precoding for the MIMO Gaussian Wiretap Channel. In *Proc. IEEE Global Commun. Conf. Workshops*. Waikoloa, HI, USA.