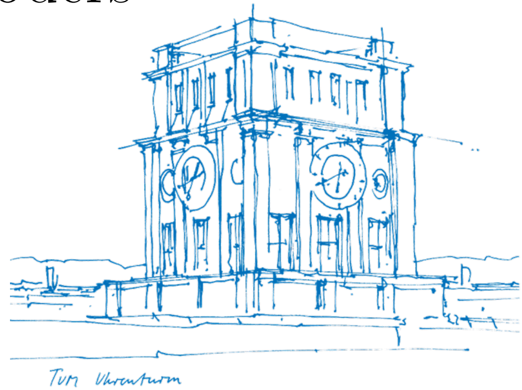
Geometric and Probabilistic Constellation Shaping with Autoencoders

Research Internship

David de Andrés Hernández
Technical University of Munich

Institute for Communications Engineering

October 6, 2022



Introduction

- Sending a single bit per time-frequency slot is inefficient.
- Higher-order modulations like amplitude shift keying (ASK) or quadrature amplitude modulation (QAM) are used for better efficiency.
- However, these schemes present a constant-width gap to the capacity limit.

Agenda

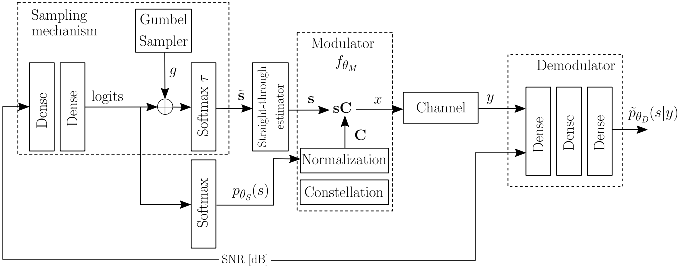
1. Probabilistic Constellation Shaping
2. Autoencoders
3. Contribution
4. Conclusion

First implementation [SAAH19]

Trainable parameters:

- P_M, source's probability distribution learnt by the encoder.
- C_M, spatial distribution of the constellation points learnt by the mapper.
- D, posterior probability distribution learnt by the demapper.

Autoencoder architecture:



Loss Function

The goal of probabilistic constellation shaping is to maximize the MI. To this end, defining an appropriate loss function is critical. Starting from the demodulator, the categorical cross entropy loss

$$L(D, P_M, C_M) \triangleq \mathbb{X}(P_{X|Y}||Q_{X|Y}; D) = \mathbb{E}\left[-\log_2(Q(X|Y; D))\right] \tag{1}$$

is appropriate for training D, but not for P_M and C_M. A modification of this loss function is necessary to ensure that the end-to-end MI is maximized. The following expansions will come handy

$$\mathbb{H}(X) = \mathbb{X}(P_{X|Y}||Q_{X|Y}) - \mathbb{D}(P_{X|Y}||Q_{X|Y}) \tag{2}$$

$$\mathbb{H}(X|Y = y) = \mathbb{X}(P_{X|y}||Q_{X|y}|Y = y) - \mathbb{D}(P_{X|y}||Q_{X|y}|Y = y) \tag{3}$$

$$\mathbb{H}(X|Y) = \mathbb{E}_y\left[\mathbb{X}(P_{X|y}||Q_{X|y}|Y = y)\right] - \mathbb{E}_y\left[\mathbb{D}(P_{X|y}||Q_{X|y}|Y = y)\right]. \tag{4}$$

Loss Function (cont'd)

Using the last expansion we can rewrite the mutual information in terms of the categorical cross entropy

$$\mathbb{I}(X, Y) = \mathbb{H}(X) - \mathbb{X}(P_{X|Y}||Q_{X|Y}) + \mathbb{D}(P_{X|Y}||Q_{X|Y}). \tag{5}$$

And the categorical cross entropy loss function becomes

$$L(D, P_M, C_M) \triangleq \mathbb{H}(X) - \mathbb{I}(X, Y) + \mathbb{D}(P_{X|Y}||Q_{X|Y}). \tag{6}$$

So, if L is minimized during training, the source entropy is unwantedly minimized. To avoid this effect, Stark *et al.* modify the loss function as

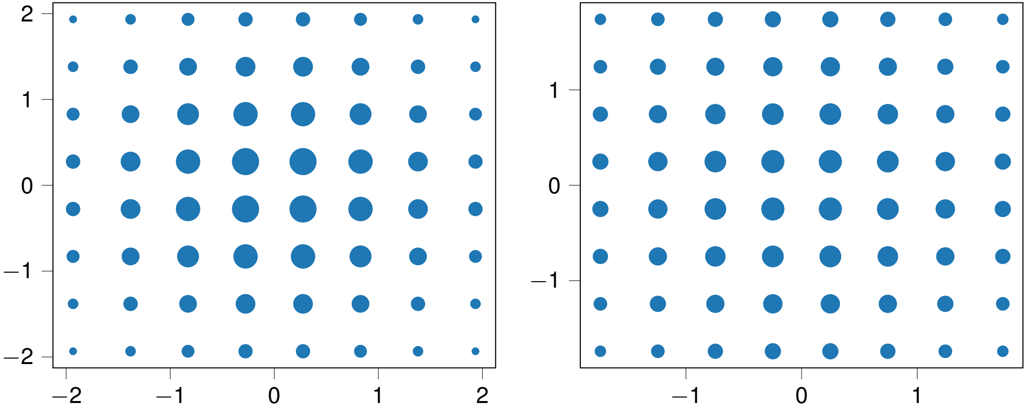
$$\hat{L}(D, P_M, C_M) \triangleq L(D, P_M, C_M) - \mathbb{H}(X). \tag{7}$$

With this correction the optimization problem

$$\min_{D,P_M,C_M} \hat{L}(D, P_M, C_M) = \max_{D,P_M,C_M} \{\mathbb{I}(X, Y) - \mathbb{D}(P_{X|Y}||Q_{X|Y})\} \tag{8}$$

maximizes the MI.

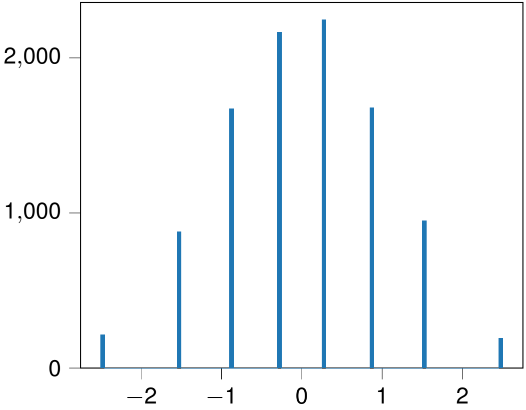
Probabilistic Constellation Shaping



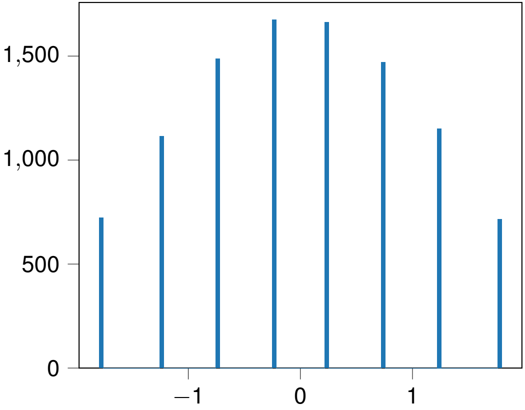
(a) 5dB (b) 18db

Figure: Learnt probabilistic constellation shaping for M = 64. The size of the markers is proportional to the transmission probability of the symbol. When trained under 5dB, the probabilistic shaping approaches a Gaussian. While under 18dB it approaches a uniform distribution.

Joint Probabilistic and Geometric Shaping



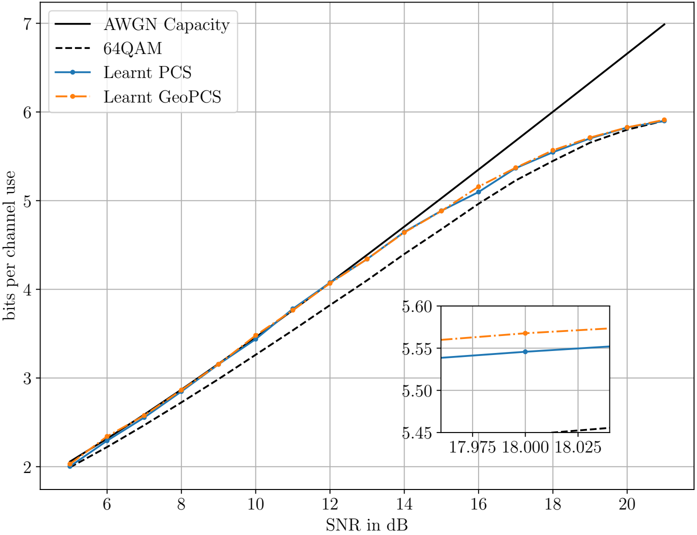
(a) SNR = 5dB



(b) SNR = 18dB

Figure: Learnt joint geometric and probabilistic ASK constellations for M=8.

Overall Performance



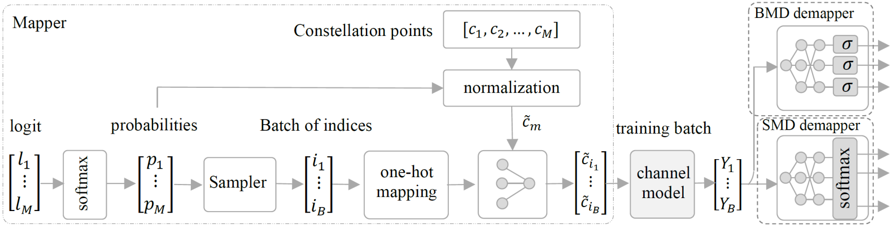
Second implementation [AC21]

What motivates another implementation?

The Gumbel-Softmax trick is complex and numerically unstable.

Trainable parameters:

- P_M, source's probability distribution learnt by the sampler.
- C_M, spatial distribution of the constellation points learnt by the mapper.
- D, posterior probability distribution learnt by the demapper.



Loss Function

The goal is again to to maximize the mutual information

$$\max_{D,P_M,C_M} \mathbb{I}(X, Y; D, P_M, C_M) = \mathbb{H}(X) - \mathbb{X}(P_{X|Y} \| Q_{X|Y}; D, P_M, C_M). \tag{9}$$

Typically, through SGD we adjust the trainable parameters as:

$$\theta_{new} = \theta_{old} + \epsilon \frac{\partial}{\partial \theta_{old}} \mathbb{I}(X, Y; \theta_{old}) \tag{10}$$

for all trainable parameters $\theta \in P_M, C_M, D$. And the MI can be numerically approximated by

$$\mathbb{I}(X, Y) \approx \mathbb{I}(X, Y)_{\text{num}} = \frac{1}{B} \sum_{i=1}^{B} -\log_2(P(x_i)) + \log_2(Q_{X|Y}(x_i|y_i)) \tag{11}$$

$$= \frac{1}{B} \sum_{i=1}^{B} L(x_i, y_i). \tag{12}$$

Next, the following approximation usually allows to adjust the trainable parameters:

$$\frac{\partial}{\partial \theta} \mathbb{I}\left(X, Y; \theta\right) \approx \frac{\partial}{\partial \theta} \mathbb{I}\left(X, Y\right)_{\text{num}} = \frac{1}{B} \sum_{i=1}^{B} L(x_i, y_i). \tag{13}$$

However, Aref claims that although this is true for the constellation locations ($\theta \in C_M$) and the demapper parameters ($\theta \in D$), it does not hold for the constellation probabilities $\{p_1, p_2, \ldots, p_M\} = P_M$

$$\frac{\partial}{\partial p_j} \mathbb{I}\left(X, Y; P_M\right) \not\approx \frac{1}{B} \sum_{i=1}^{B} \frac{\partial}{\partial p_j} L(x_i, y_i) \tag{14}$$

as $\{p_1, p_2, \ldots, p_M\}$ changes the statistics of the training set.

For this reason, (14) must be computed differently.

$$\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}\right) = \sum_{(a,b)\in Supp(P_{XY})} P_X(a)P_{Y|X}(b|a)\log_2(Q_{X|Y}(a|b)). \tag{15}$$

And so, the derivative results

$$\frac{\partial}{\partial p_j}\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}\right) = \sum_{b \text{ if } x=j} P_{Y|X}(b|j)\log_2 Q_{X|Y}(j|b) + \sum_{(a,b)\in Supp(P_{XY})} P_{XY}(a,b)\frac{\partial}{\partial p_j}\log_2 Q_{X|Y}(a|b), \tag{16}$$

which can be rewritten using the expectation operator as

$$\frac{\partial}{\partial p_j}\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}\right) = \mathbb{E}_{Y|X}[\log_2 Q_{X|Y}(j|b)|X=j] + \mathbb{E}_{XY}[\frac{\partial}{\partial p_j}\log_2 Q_{X|Y}(a|b)]. \tag{17}$$

The terms can now be numerically computed as

$$\mathbb{E}_{Y|X}[\log_2 Q_{X|Y}(j|b)|X=j] \approx \frac{1}{Bp_j}\sum_{b \text{ if } x=j}\log_2 Q_{X|Y}(j|b) \tag{18}$$

$$\mathbb{E}_{XY}[\frac{\partial}{\partial p_j}\log_2 Q_{X|Y}(a|b)] \approx \frac{1}{B}\sum_{(a,b)\in Supp(P_{XY})}\log_2 Q_{X|Y}(a|b). \tag{19}$$

On the other hand, the derivative of the entropy w.r.t. p_j is

$$\frac{\partial}{\partial p_j}\mathbb{H}(X) = \frac{\partial}{\partial p_j}\sum_{i=1}^{B} -p_i \log_2(p_i) = -\log_2(p_j) - log_2(e). \tag{20}$$

Now, combining (18), (19), and (20) the derivative of the mutual information w.r.t. p_j, (14), can be computed as

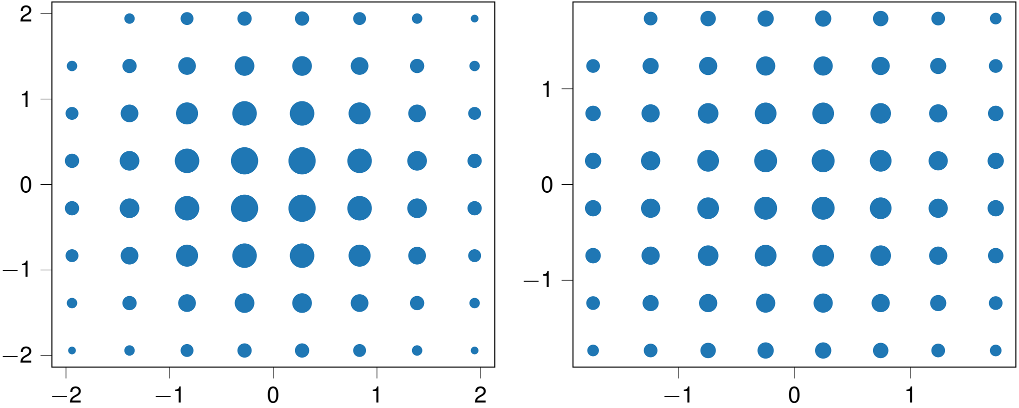
$$\frac{\partial}{\partial p_j}\mathbb{I}\,(X, Y; P_M) \approx -\log_2(p_j) - \log_2(e) + \frac{1}{Bp_j}\sum_{b \text{ if } x=j} \log_2 Q_{X|Y}(j|b) + \frac{1}{B}\sum_{(a,b)} \log_2 Q_{X|Y}(a|b) \tag{21}$$

Aref now indicates that the following terms can be computed via backpropagation

$$-\log_2(p_j) + \frac{1}{Bp_j}\sum_{b \text{ if } x=j} \log_2 Q_{X|Y}(j|b) = \frac{1}{B}\sum_{i=1}^{B} \frac{\partial}{\partial p_j} L(x_i, y_i) \tag{22}$$

while the remaining ones must be explicitly computed and added to the gradient after backpropagating. We call this step *gradient correction* and it is due to the change of statistics in the sampled batch.

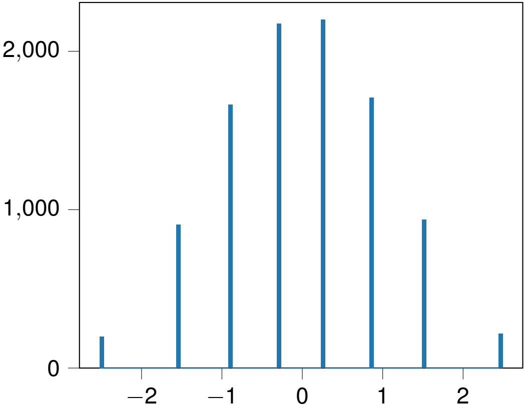
Probabilistic Shaping



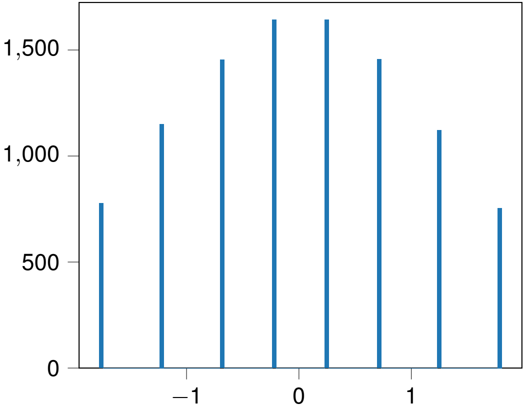
(a) SNR = 5dB (b) SNR = 18dB

Figure: Learnt probabilistic constellation shaping for M = 64. The size of the markers is proportional to the transmission probability of the symbol. When trained under 5dB, the proabability shaping approaches a gaussian. While under 18dB it approaches a uniform distribution.

Joint Geometric and Probabilistic Shaping



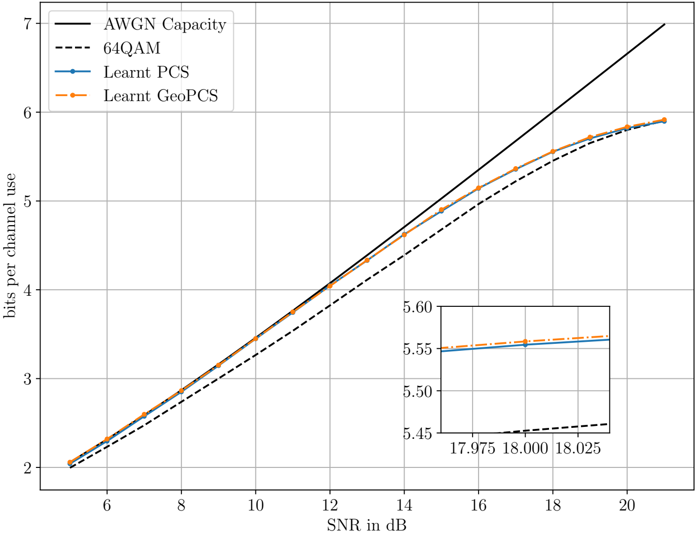
(a) SNR = 5dB



(b) SNR = 18dB

Figure: Learnt joint geometric and probabilistic ASK constellations for M=8.

Overall Performance



Bibliography

[AC21] V. Aref and M. Chagnon, *End-to-end learning of joint geometric and probabilistic constellation shaping*, 2021. DOI: 10.48550/ARXIV.2112.05050. [Online]. Available: https://arxiv.org/abs/2112.05050.

[SAAH19] M. Stark, F. Ait Aoudia, and J. Hoydis, “Joint learning of geometric and probabilistic constellation shaping,” in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6. DOI: 10.1109/GCWkshps45667.2019.9024567.