Joint Learning of Geometric and Probabilistic Constellation Shaping

Maximilian Stark*§¶, Fayçal Ait Aoudia‡§, and Jakob Hoydis‡
*Hamburg University of Technology Institute of Communications, Hamburg, Germany, Email: maximilian.stark@tuhh.de
‡Nokia Bell Labs, Paris, France, Email: {faycal.ait_aoudia, jakob.hoydis}@nokia-bell-labs.com

***Abstract*—The choice of constellations largely affects the performance of communication systems. When designing constellations, both the locations and probability of occurrence of the points can be optimized. These approaches are referred to as geometric and probabilistic shaping, respectively. Usually, the geometry of the constellation is fixed, e.g., quadrature amplitude modulation (QAM) is used. In such cases, the achievable information rate can still be improved by probabilistic shaping. In this work, we show how autoencoders can be leveraged to perform probabilistic shaping of constellations. We devise an information-theoretical description of autoencoders, which allows learning of capacity-achieving symbol distributions and constellations. Recently, machine learning techniques to perform geometric shaping were proposed. However, probabilistic shaping is more challenging as it requires the optimization of discrete distributions. Furthermore, the proposed method enables joint probabilistic and geometric shaping of constellations over any channel model. Simulation results show that the learned constellations achieve information rates very close to capacity on an additive white Gaussian noise (AWGN) channel and outperform existing approaches on both AWGN and fading channels.**

***Index Terms*—Probabilistic shaping, Geometric shaping, Autoencoders**

I. Introduction

Various constellation schemes were developed in digital communications, including quadrature amplitude modulation (QAM), phase-shift keying (PSK), amplitude-shift keying (ASK) etc. Shaping of constellations involves either optimizing the locations of the constellation points in the complex plane, i.e., geometric shaping, or optimizing the probabilities of occurrence of the constellation points, i.e., probabilistic shaping. In either case, the focal aim is to maximize the mutual information $I(X;Y)$ of the channel input X and output Y by optimizing the constellation. This approach follows directly from the definition of the channel capacity C:

$$C = \max_{p(x)} I(X;Y) \tag{1}$$

where $p(x)$ denotes the marginal distribution of X. Usually, finding the optimal $p(x)$ is a difficult problem as it requires the knowledge of the channel distribution $p(y|x)$. Moreover, even if $p(y|x)$ is known, solving (1) is often intractable.

In this work, we present how the recently proposed idea of end-to-end learning of communication systems by leveraging autoencoders [1] can be used to design constellations which maximize $I(X;Y)$, without requiring any tractable model of the channel. Autoencoders have been used in the past to perform geometric shaping [1], [2]. However, to the best of our knowledge, leveraging autoencoders to achieve probabilistic shaping has not been explored. In this paper, probabilistic shaping is learned by leveraging the recently proposed Gumbel-Softmax trick [3] to optimize discrete distributions. Afterwards, joint geometric and probabilistic shaping of constellation is performed. Presented results show that the achieved mutual information outperforms state-of-the-art systems over a wide range of signal-to-noise ratios (SNRs) and on both additive white Gaussian noise (AWGN) and fading channels, whereas current approaches are typically optimized to perform well only on specific channel models and small SNR ranges [4].

The rest of this paper is organized as follows. Section II provides background on autoencoder-based communication systems and motivates their use for constellation shaping. Section III details the considered neural network (NN) architecture and how the Gumbel-Softmax trick is leveraged to achieve probabilistic shaping. Section IV provides results on the mutual information achieved by different schemes. Finally, Section V concludes this paper.

Notations

Random variables are denoted by capital italic font, e.g., X, Y with realizations $x \in \mathcal{X}, y \in \mathcal{Y}$, respectively. $I(X;Y)$, $p(y|x)$ and $p(x,y)$ represent the mutual information, conditional probability and joint probability distribution of the two random variables X and Y. Vectors are represented using a lower case bold font, e.g., $\mathbf{y}$, upper case bold font letters denote matrices, e.g., $\mathbf{C}$.

II. Autoencoder-based communication systems

The key idea of autoencoder-based communication systems is to regard transmitter, channel, and receiver as a single NN such that the transmitter and receiver can be optimized in an end-to-end manner. This idea was pioneered in [1], and has led to many extensions [5]–[8]. Fig. 1 shows the end-to-end communication system considered in this work. The system takes as input a bit sequence denoted by $\mathbf{b}$ which is mapped onto hypersymbols $s \in \mathcal{S}$ such that symbols s appear with frequencies corresponding to a parametric distribution $p_{\boldsymbol{\theta}_S}(s)$ with parameters $\boldsymbol{\theta}_S$. Here, $\mathcal{S} = \{1, \dots, N\}$ is the eventspace of the random variable S, N being the modulation order. The

§Equally contributed.
¶Work carried out at Nokia Bell Labs France.

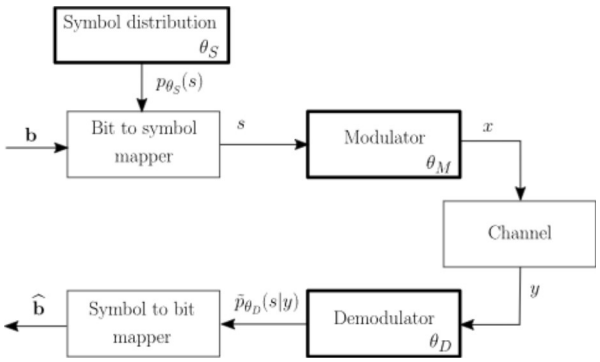

Fig. 1: Trainable end-to-end communication system. Components on which this work focuses are indicated by thicker outlines.

sequence of hypersymbols is fed into a symbol modulator which maps each symbol s into a constellation point $x \in \mathbb{C}$. The modulator is implemented as an NN $f_{\boldsymbol{\theta}_M}$ with trainable parameters $\boldsymbol{\theta}_M$.

The demodulator is also implemented as an NN with trainable parameters $\boldsymbol{\theta}_D$, which maps each received sample $y \in \mathbb{C}$ to a probability vector over the set of symbols $\mathcal{S}$. The mapping defined by the demodulator is denoted by $\tilde{p}_{\boldsymbol{\theta}_D}(s|y)$, and defines, as it will be seen below, an approximation of the true posterior distribution $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(s|y)$. Finally, the sent bits are reconstructed by the symbols to bits mapper from $\tilde{p}_{\boldsymbol{\theta}_D}(s|y)$.

A. Mutual Information Perpective on Autoencoders

In this work, it is assumed that a bits to symbols mapper exists, which maps the bits from $\mathbf{b}$ to symbols $s \in \mathcal{S}$ according to the distribution $p_{\boldsymbol{\theta}_S}(s)$. This can be done, e.g., using the algorithm presented in [9]. Therefore, in the rest of this work, the transmitter directly outputs the transmit symbols sampled from $p_{\boldsymbol{\theta}_S}(s)$, and the receiver aims to reconstruct the transmitted symbols by approximating the posterior distribution $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(s|y)$. Thus, only the signal processing blocks surrounded by thicker outlines in Fig. 1 are of interest in this work. The distribution of X equals

$$p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x) = \sum_{s=1}^{N} \delta\left(x - f_{\boldsymbol{\theta}_M}(s)\right) p_{\boldsymbol{\theta}_S}(s). \tag{2}$$

where $\delta(.)$ denotes the Dirac distribution. Please recall, that, as defined in (1), the target of constellation shaping is to find $p_{\boldsymbol{\theta}_S}(s)$, such that $I(X;Y)$ is maximized. One performs constellation shaping by optimizing $p_{\boldsymbol{\theta}_S}$ (probabilistic shaping) or $f_{\boldsymbol{\theta}_M}$ (geometric shaping) so that $I(X;Y)$ is maximized.

As the demodulator performs a classification task, for training, the categorical cross entropy

$$\mathcal{L}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_M, \boldsymbol{\theta}_D) \triangleq \mathbb{E}_{s,y}\left\{-\log\left(\tilde{p}_{\boldsymbol{\theta}_D}(s|y)\right)\right\} \tag{3}$$

$$= -\sum_{s=1}^{N} p_{\boldsymbol{\theta}_S}(s) \int_y p\left(y|f_{\boldsymbol{\theta}_M}(s)\right) \log\left(\tilde{p}_{\boldsymbol{\theta}_D}(s|y)\right) dy \tag{4}$$

is used as loss function. Rewriting the loss function yields

$$\mathcal{L}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_M, \boldsymbol{\theta}_D) = H_{\boldsymbol{\theta}_S}(S) - I_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(X;Y) + \mathbb{E}_y\left\{\mathrm{D_{KL}}\left(p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x|y)||\tilde{p}_{\boldsymbol{\theta}_D}(x|y)\right)\right\} \tag{5}$$

where $\mathrm{D_{KL}}$ is the Kullback–Leibler (KL) divergence. A more detailed derivation is given in the Appendix.

Notice that if only geometric shaping is performed, no optimization with respect to (w.r.t.) $\boldsymbol{\theta}_S$ is done and therefore the first term in (5) is a constant. However, when performing probabilistic shaping, minimizing $\mathcal{L}$ leads to the minimization of $H_{\boldsymbol{\theta}_S}(S)$. To avoid this unwanted effect, we define the loss function

$$\widehat{\mathcal{L}}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_M, \boldsymbol{\theta}_D) \triangleq \mathcal{L}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_M, \boldsymbol{\theta}_D) - H_{\boldsymbol{\theta}_S}(S). \tag{6}$$

Training the end-to-end system by minimizing $\widehat{\mathcal{L}}$ corresponds to maximizing the mutual information of the channel inputs X and outputs Y, while minimizing the KL divergence between the true posterior distribution $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x|y)$ and the one learned by the receiver $\tilde{p}_{\boldsymbol{\theta}_D}(x|y)$. Moreover, the NN implementing the receiver should approximate the posterior distribution $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x|y)$ of a constellation maximizing the mutual information with high precision. This avoids learning a constellation where the posterior distribution is well approximated, but which does not maximize the mutual information. In practice, this is ensured by choosing the NN implementing the demodulator complex enough to ensure that the trainable receiver is capable of approximating a wide range of posterior distribution with high precision.

Recently, [10] proposed to leverage the mutual information neural estimator (MINE) approach as described in [11] to train the transmitter such that the estimated mutual information of the input and output of the channel is maximized. Therefore, training a transmitter with this approach is similar to training a transmitter as a part of an autoencoder, as in both cases the transmitter is trained to maximize the mutual information. Optimization using MINE does not require to train the receiver. However, the autoencoder approach jointly learns the transmitter and the receiver including the corresponding posterior distribution. The respective soft information output by the learned receiver can then be used in subsequent units, e.g., a channel decoder. Moreover, whereas MINE requires an additional NN only to approximate the mutual information, using an autoencoder the mutual information can be estimated from the loss as $-\widehat{\mathcal{L}}$ provides a tight lower bound, assuming a sufficiently complex NN implementing the receiver.

As by training an autoencoder-based communication system one maximizes the mutual information of X and Y, it can be used to perform constellation shaping. Although, geometric shaping using autoencoders has been done in the past [1], [2], [12], performing probabilistic shaping is less straightforward as it requires to optimize the sampling mechanism for symbols s drawn from $\mathcal{S}$.

III. Learning Constellation Shaping

The end-to-end system considered in this work is presented in Fig. 1. This section details the architecture of each trainable

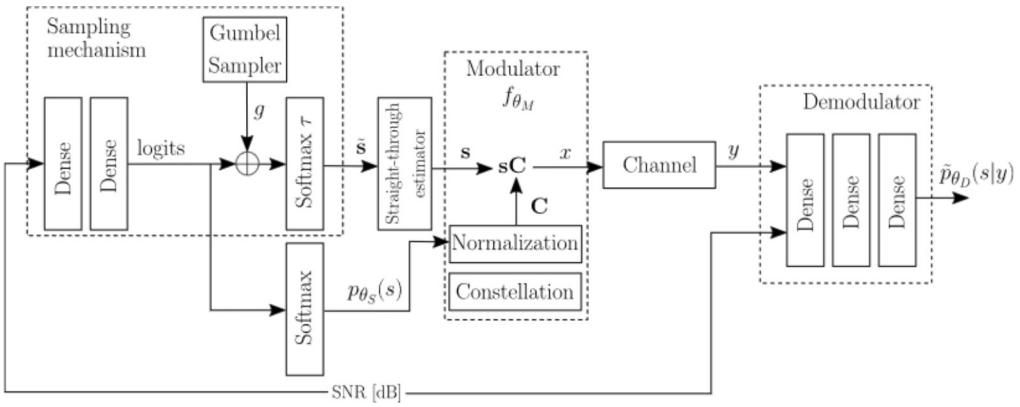

Fig. 2: End-to-end system architecture

element, i.e., the symbol distribution, the modulator and the demodulator. Fig. 2 shows in detail the architecture of the considered end-to-end system.

A. Symbols Distribution

The challenge of performing probabilistic shaping with machine learning-based algorithms comes from the difficulty of training a sampling mechanism for symbols s drawn from the finite set $\mathcal{S}$. This issue is addressed in this work by leveraging the Gumbel-Softmax trick [3], an extension of the Gumbel-Max trick [13]. The Gumbel-Max trick provides a convenient way to sample a discrete distribution $p_{\boldsymbol{\theta}_S}(s)$, by computing the samples as follows:

$$s = \underset{i=1,\dots,S}{\arg\max} \ (g_i + \log\left(p_{\boldsymbol{\theta}_S}(i)\right)) \tag{7}$$

where g_i are independent and identically distributed (i.i.d.) samples drawn from a standard Gumbel distribution. Because the $\arg\max$ operator is not differentiable, one cannot train $p_{\boldsymbol{\theta}_S}(s)$ using usual stochastic gradient descent (SGD) methods. The key idea of the Gumbel-Softmax trick is to use the softmax function as a differentiable approximation to $\arg\max$. More precisely, one generates a vector of dimension N, denoted by $\tilde{\mathbf{s}}$, with components

$$\tilde{s}_i = \frac{\exp\left(\left(g_i + \log\left(p_{\boldsymbol{\theta}_S}(i)\right)\right)/\tau\right)}{\sum_{j=1}^{S} \exp\left(\left(g_j + \log\left(p_{\boldsymbol{\theta}_S}(j)\right)\right)/\tau\right)}, \ i = 1, \dots, N \tag{8}$$

where τ is a positive parameter called the *temperature*. $\tilde{\mathbf{s}}$ is a probability vector which is such that $\arg\max_i \ \tilde{s}_i = s$. It is an approximation of the *one-hot* representation of s denoted by $\mathbf{s}$, i.e., the $N-$dimensional vector for which all elements are set to zero except the sth which is set to one. As the temperature goes to zero, samples generated by the Gumbel-Softmax method become closer to one-hot vectors, and their distribution becomes closer to $p_{\boldsymbol{\theta}_S}(s)$ [3].

Fig. 2 shows the architecture of the sampling mechanism. The optimal probabilistic shaping depends on the SNR of the respective channel, which therefore must be *a priori* known by the transmitter [4]. The SNR (in dB) is fed to an NN with trainable parameters $\boldsymbol{\theta}_S$, and therefore the NN generates a continuum of distributions $p_{\boldsymbol{\theta}_S}$ that are determined by the SNR. The NN is made of two dense layers, and generates the *logits* of the symbols distribution $p_{\boldsymbol{\theta}_S}(s)$. The logits are the unormalized log probabilities, and the distribution $p_{\boldsymbol{\theta}_S}(s)$ can be retrieved by applying a softmax activation to the logits. The first dense layer is made of 128 units with ReLU activations, and the second layer of N units with linear activations. By tuning the NN parameters $\boldsymbol{\theta}_S$, one therefore optimizes the distribution $p_{\boldsymbol{\theta}_S}(s)$. The Gumbel-Softmax trick is then applied to the logits.

B. Modulator

The modulator is made of a matrix of dimension $N \times 2$ followed by a normalization layer, as shown in Fig. 2. The matrix consists of the unnormalized constellation point locations. The normalized constellation is denoted by $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_i, \dots, \mathbf{c}_N]^\mathsf{T}$ where $\mathbf{c}_i \in \mathbb{R}^2$, $i = 1, \dots, N$. By taking the product of a one-hot vector $\mathbf{s}$ with the sth element set to one and $\mathbf{C}$, one selects a constellation point. Normalization is performed to ensure the expected energy of the constellation equals unity, i.e.,

$$\sum_{s \in \mathcal{S}} p_{\boldsymbol{\theta}_S}(s) \, |x_s|_2^2 = 1. \tag{9}$$

If only probabilistic shaping is performed, the constellation is not trained and some fixed constellation, e.g., QAM, is used. When geometric shaping is performed, the constellation is trainable, and $\boldsymbol{\theta}_M$ corresponds to the unnormalized constellation points.

A drawback of the Gumbel-Softmax trick is that the generated vector $\tilde{\mathbf{s}}$ is only an approximation of a true one-hot vector $\mathbf{s}$. As a consequence, taking the product of $\tilde{\mathbf{s}}$ and $\mathbf{C}$ results in a convex combination of multiple constellation points $\mathbf{c}_s$. To avoid this issue, we take advantage of the straight-through estimator [14], which uses the true one-hot vectors $\mathbf{s}$ for the forward pass and the approximate one-hot vector $\tilde{\mathbf{s}}$ for the backward pass at training.

C. Demodulator

The trainable demodulator consists of three dense layers, as shown in Fig. 2. The first two layers are made of 128 units

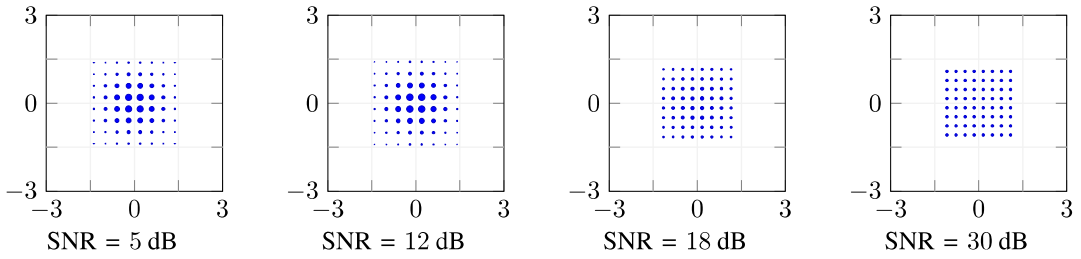

Fig. 3: Learned probabilistic shaping for $N = 64$. The size of the points is proportional to their probabilities of occurrence.

with ReLU activations, while the last layer is made of N units with softmax activation, to output a probability vector over the set of symbols $\mathcal{S}$. As opposed to prior art, the demodulator takes as input the SNR (in dB). This was motivated by the observation that the posterior distribution depends on the SNR, and it was experimentally found out to be crucial to achieve the best performance.

IV. Simulations Results

In this section, the mutual information of the channel input and output achieved by the proposed scheme is compared to state-of-the-art modulation schemes considering AWGN and Rayleigh channels. Training of the end-to-end system introduced in the previous section is performed w.r.t. to the loss function $\widehat{\mathcal{L}}$ defined in (6), as opposed to previous works which train autoencoders w.r.t. to the usual cross-entropy $\mathcal{L}$ defined in (5). Using $\widehat{\mathcal{L}}$ as loss function is crucial to enable probabilistic shaping, as using $\mathcal{L}$ would rather lead to a minimization of the source entropy $H_{\boldsymbol{\theta}_S}(S)$ instead of maximization of the mutual information $I(X;Y)$. The autoencoder was implemented with the TensorFlow framework [15]. Training was performed with the Adam SGD variant [16], with batch sizes progressively increasing from 100 to 10000, and learning rates progressively decreasing from 10^{-3} to 10^{-5}. When probabilistic shaping was performed, the temperature in (8) was set to 10. First, we present the results for learned probabilistic shaping of a QAM modulation. Afterwards, we show the results obtained when both the locations and probabilities of the constellation points are optimized, i.e., joint geometric and probabilistic shaping. The considered modulation orders N are 16, 64, 256, and 1024.

A. Probabilistic Shaping over the AWGN Channel

Probabilistic shaping of QAM is studied in great detail in [4]. It is well know [17] that for an AWGN channel, distributions $p(s)$ from the Maxwell-Boltzmann family maximize the mutual information $I(X;Y)$ and, thus, allow to achieve the capacity in a certain regime. However, in this work, we do not enforce the learning of distributions from this family. The following modulation schemes are compared: QAM with no probabilistic shaping, QAM with probabilistic shaping optimized as proposed in this work, and QAM with Maxwell-Boltzmann distribution as probabilistic shaping as in [4]. QAM with Maxwell-Boltzmann from [4] is only presented for

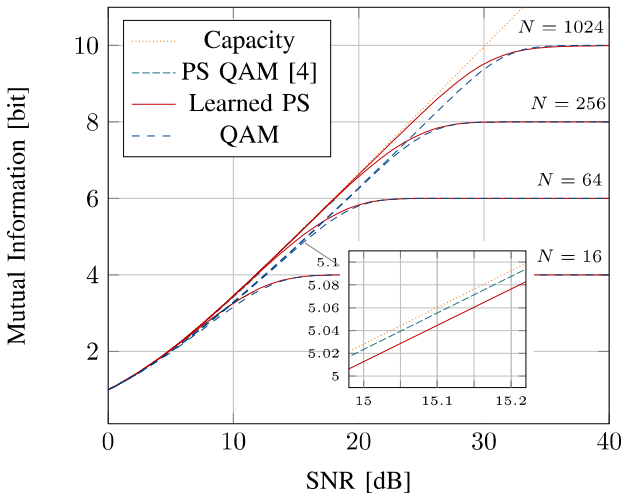

Fig. 4: Mutual information achieved by the reference schemes and the learned probabilistic shaping on the AWGN channel. Magnification is done for $N = 256$.

modulation orders of 16, 64, and 256, and the distributions are optimized for specific SNR values. Notice that the proposed approach has the benefit of training an NN which computes optimized shaping distributions over a wide range of SNRs. For this evaluation, the sampling mechanism was trained for SNR values ranging from -2 dB to 40 dB.

The learned probability distributions are depicted for $N = 64$ in Fig. 3 and for several SNRs. The size of the points is proportional to their probabilities of occurrence. It can be seen that the learned distributions look similar to a circular two-dimensional Gaussian distribution. With increasing SNRs, the learned distribution approaches a uniform distribution. In Fig. 4, the mutual information $I(X;Y)$ achieved by the proposed scheme (cf. red solid curve) is compared to several reference schemes. First QAM with no probabilistic shaping is considered (cf. blue dashed curve). Clearly, the proposed schemes outperform non-shaped QAM. Furthermore, Fig. 4 includes the mutual information curves of the probabilistic shaping scheme proposed in [4] (cf. green dash dotted curve). Interestingly, the learned scheme is able to achieve a performance close to the probabilistic shaping from [4], which leverages the optimum Maxwell-Boltzmann distribution. No-

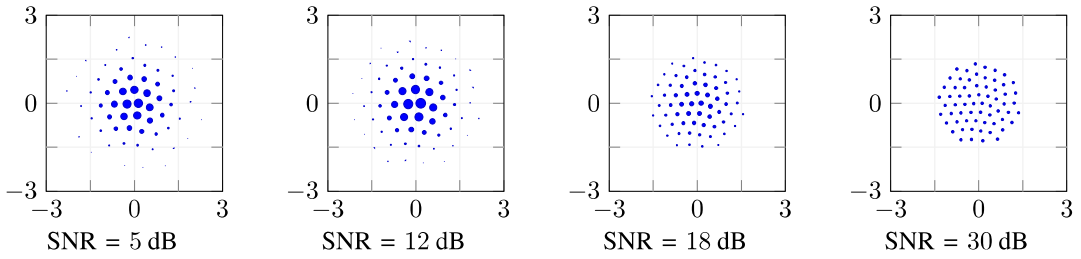

Fig. 5: Learned joint shaping for $N = 64$. The size of the points is proportional to their probabilities of occurrence.

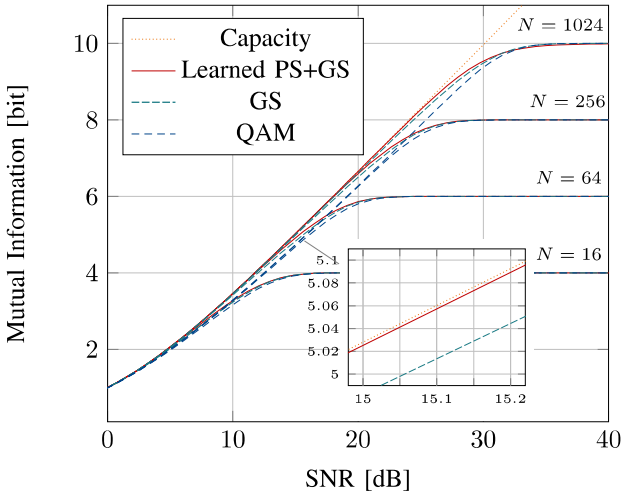

Fig. 6: Mutual information achieved by the reference schemes and the learned joint probabilistic and geometric shaping on the AWGN channel. Magnification is done for $N = 256$.

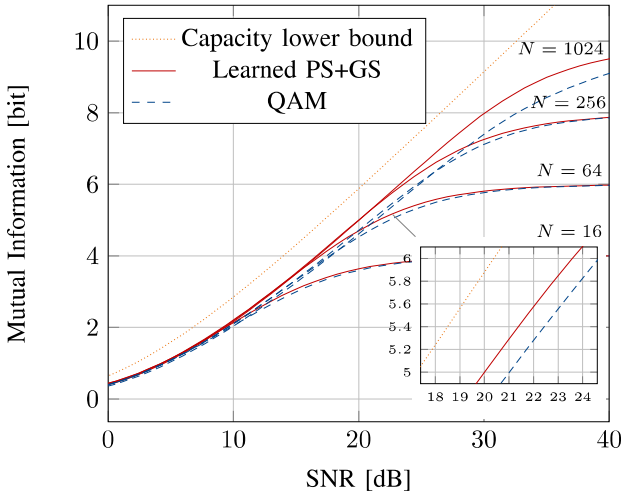

Fig. 7: Mutual information achieved by the reference schemes and the learned joint probabilistic and geometric shaping on the Rayleigh channel. Magnification is done for $N = 256$.

tice that no assumption on the channel was made to achieve this result.

B. Joint Shaping over the AWGN Channel

In this section, both the probability distribution and the geometry are assumed to be trainable. Like in the previous section, the modulation orders 16, 64, 256, and 1024 and an AWGN channel are considered. Fig. 5 shows the joint probabilistic and geometric constellations for various SNR values and for $M = 64$. It can be observed that the learned shaping is similar to a two-dimensional Gaussian distribution. For lower SNRs, the learned shaping favors constellation points closer to the origin. Due to the normalization, which ensures that $\mathbb{E}\{|x|^2\} = 1$, the less frequently transmitted outer points are placed further apart form the origin. As the SNR increases, the distribution becomes uniform.

In Fig. 6, the mutual information $I(X;Y)$ achieved by the proposed joint shaping scheme (cf. red solid curve) is compared to several reference schemes. First, QAM with no probabilistic shaping is considered (cf. blue dashed curve). As for probabilistic shaping alone, the proposed joint shaping outperforms non-shaped QAM. Furthermore, Fig. 4 includes the mutual information curves performing only geometric shaping (cf. green dash dotted curve). Joint geometric and probabilistic is also superior to geometric shaping alone. Also, one observes that for all the modulation orders, probabilistic shaping achieves higher rates than geometric shaping, showing that probabilistic shaping of QAM constellations enables higher gains than geometric shaping without probabilistic shaping.

It can be seen that the proposed NN learns a joint probabilistic and geometric shaping which operates very close to capacity for a wide range of SNRs. From Fig. 4 and Fig. 6, one observes that the achievable gains enabled by constellation shaping are more significant the higher the modulation order gets. In addition, comparing Fig. 4 and Fig. 6 reveals that for all the modulation orders, the highest mutual information is achieved by the joint geometric and probabilistic scheme, which outperforms PS-QAM from [4].

C. Joint Shaping over the Rayleigh Channel

A Rayleigh channel with linear minimum mean square error (LMMSE) estimation and equalization is considered in this section. Fig. 7 shows the lower bound on the capacity from [18, Corollary 1.3] (cf. orange dotted curve), and the mutual information achieved by the proposed joint

shaping scheme (cf. red curve). To the best of our knowledge, no optimal shaping scheme is known for the mismatched Rayleigh channel and, therefore, unshaped QAM is considered as a baseline (cf. blue dashed plot). One observes that the proposed joint-shaping approach enables significant gains over non-shaped QAM. An important aspect of this result is that no theoretical analysis or assumption on the channel was required to perform joint shaping. This result is therefore encouraging to apply the proposed approach to other channels with untractable models. Moreover, one could also perform shaping over non-differentiable or unknown channel models using recently proposed approaches [12] that remove the need of backpropagating gradients through the channel at training.

V. Conclusion

This paper introduced a machine-learning based solution to the problem of constellation shaping. In contrast to prior works on probabilistic shaping, the proposed scheme operates over a wide range of SNRs and is not limited to specific families of distributions. Instead, using the Gumbel-Softmax trick, a continuum of arbitrary distributions being function of the SNR was learned. As an extension, the presented approach was generalized to joint geometric and probabilistic shaping. It was shown that joint shaping outperforms single geometric shaping or probabilistic shaping of QAM regarding the achieved mutual information, and nearly reaches the capacity on an AWGN channel. On a Rayleigh channel, the proposed joint-shaping scheme outperforms unshaped QAM. The presented results are promising for other channel models, left as future research directions.

Appendix

The categorical cross entropy loss function can be rewritten as

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_M, \boldsymbol{\theta}_D) \\
&= -\int_x p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x) \int_y p(y|x) \log\left(\tilde{p}_{\boldsymbol{\theta}_D}(x|y)\right) dy dx \\
&= -\int_x \int_y p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x,y) \log\left(\tilde{p}_{\boldsymbol{\theta}_D}(x|y)\right) dy dx \\
&= -\int_x p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x) \log\left(p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x)\right) dx \\
&\quad - \int_x \int_y p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x,y) \log\left(\frac{\tilde{p}_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M,\boldsymbol{\theta}_D}(x,y)}{p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(y) p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x)}\right) dy dx
\end{aligned} \tag{10}$$

where $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x,y) = p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x)p(y|x)$, $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(y) = \int_x p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x,y)$, and $\tilde{p}_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M,\boldsymbol{\theta}_D}(x,y) = \tilde{p}_{\boldsymbol{\theta}_D}(x|y)p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(y)$.

It is important to notice that $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x,y)$ is the true joint distribution of (X,Y), whereas $\tilde{p}_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M,\boldsymbol{\theta}_D}(x,y)$ is the joint distribution computed from the posterior approximated by the demodulator $\tilde{p}_{\boldsymbol{\theta}_D}(x|y)$. Assuming that there is no $(s,t) \in \mathcal{S}^2$ such that $s \neq t$ and $f_{\boldsymbol{\theta}_M}(s) = f_{\boldsymbol{\theta}_M}(t)$, meaning that each symbol $s \in \mathcal{S}$ is uniquely mapped to a constellation point $x \in \mathbb{C}$, then the first term of (10) is the entropy of S, denoted by $H_{\boldsymbol{\theta}_S}(S)$. One can see that

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_M, \boldsymbol{\theta}_D) &= H_{\boldsymbol{\theta}_S}(S) - I_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(X;Y) \\
&\quad + \mathbb{E}_y\left\{\mathrm{D_{KL}}\left(p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x|y) || \tilde{p}_{\boldsymbol{\theta}_D}(x|y)\right)\right\}
\end{aligned} \tag{11}$$

where $\mathrm{D_{KL}}$ is the KL divergence, and $p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x|y) = \frac{p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(x,y)}{p_{\boldsymbol{\theta}_S,\boldsymbol{\theta}_M}(y)}$ is the true posterior of X.

References

[1] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[2] R. T. Jones, T. A. Eriksson, M. P. Yankov, B. J. Puttnam, G. Rademacher, R. S. Luis, and D. Zibar, "Geometric constellation shaping for fiber optic communication systems via end-to-end learning," *arXiv preprint arXiv:1810.00774*, 2018.

[3] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[4] G. Böcherer, F. Steiner, and P. Schulte, "Bandwidth efficient and rate-matched low-density parity-check coded modulation," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 4651–4665, Dec 2015.

[5] H. Kim, Y. Jiang, R. B. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," in *Int. Zurich Seminar Inf. Commun. (IZS)*, Feb. 2018, pp. 48 – 50.

[6] M. Kim, W. Lee, and D. H. Cho, "A novel PAPR reduction scheme for OFDM system based on deep learning," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 510–513, Mar. 2018.

[7] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," in *Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 76–80.

[8] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, May 2019, pp. 4774–4778.

[9] P. Schulte and G. Böcherer, "Constant composition distribution matching," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 430–434, Jan 2016.

[10] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," *arXiv preprint arXiv:1903.02865*, 2019.

[11] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proceedings of the 35th Int. Conf. on Mach. Learning*, vol. 80, 10–15 Jul 2018, pp. 531–540.

[12] F. Ait Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE J. Sel. Areas Commun.*, 2019.

[13] T. Hazan and T. Jaakkola, "On the partition function and random maximum a-posteriori perturbations," in *Proceedings of the 29th Int. Conf. on Mach. Learning*, 2012, pp. 1667–1674.

[14] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," *arXiv preprint arXiv:1308.3432*, 2013.

[15] "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, accessed: 2019-15-07. [Online]. Available: http://tensorflow.org/

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–15.

[17] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for gaussian channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, 1993.

[18] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017. [Online]. Available: http://dx.doi.org/10.1561/2000000093