CHALMERS
UNIVERSITY OF TECHNOLOGY

Learning Physical-Layer Communication with Quantized Feedback

Downloaded from: https://research.chalmers.se, 2025-06-18 04:09 UTC

Citation for the original published paper (version of record):

Song, J., Peng, B., Häger, C. et al (2020). Learning Physical-Layer Communication with Quantized Feedback. IEEE Transactions on Communications, 68(1): 645-653.
http://dx.doi.org/10.1109/TCOMM.2019.2951563

N.B. When citing this work, cite the original published paper.

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

Learning Physical-Layer Communication with Quantized Feedback

Jinxiang Song, Bile Peng, Christian Häger, Henk Wymeersch, Anant Sahai

***Abstract*—Data-driven optimization of transmitters and receivers can reveal new modulation and detection schemes and enable physical-layer communication over unknown channels. Previous work has shown that practical implementations of this approach require a feedback signal from the receiver to the transmitter. In this paper, we study the impact of quantized feedback on data-driven learning of physical-layer communication. A novel quantization method is proposed, which exploits the specific properties of the feedback signal and is suitable for non-stationary signal distributions. The method is evaluated for linear and nonlinear channels. Simulation results show that feedback quantization does not appreciably affect the learning process and can lead to similar performance as compared to the case where unquantized feedback is used for training, even with 1-bit quantization. In addition, it is shown that learning is surprisingly robust to noisy feedback where random bit flips are applied to the quantization bits.**

I. Introduction

As communication systems become more complex, physical-layer design, i.e., devising optimal transmission and detection methods, has become harder as well. This is true not only in wireless communication, where hardware impairments and quantization have increasingly become a limitation on the achievable performance, but also in optical communication, for which the nonlinear nature of the channel precludes the use of standard approaches. This has led to a new line of research on physical-layer communication where transmission and detection methods are learned from data. The general idea is to regard the transmitter and receiver as parameterized functions (e.g., neural networks) and find good parameter configurations using large-scale gradient-based optimization approaches from machine learning.

Data-driven methods have mainly focused on learning receivers assuming a given transmitter and channel, e.g., for MIMO detection [1] or decoding [2]. These methods have led to algorithms that either perform better or exhibit lower complexity than model-based algorithms. More recently, end-to-end learning of both the transmitter and receiver has been proposed for various physical-layer applications including wireless [3], [4], nonlinear optical [5]–[7], and visible light communication [8].

In practice, gradient-based transmitter optimization is problematic since it requires a known and differentiable channel model. One approach to circumvent this limitation is to first learn a surrogate channel model, e.g., through an adversarial process, and use the surrogate model for the optimization [9], [10]. We follow a different approach based on stochastic transmitters, where the transmitted symbol for a fixed message is assumed to be a random variable during the training process [11]–[13]. This allows for the computation of *surrogate gradients* which can then be used to update the transmitter parameters. A related approach is proposed in [14].[1]

In order to compute the surrogate gradients, the transmitter must receive a *feedback signal* from the receiver. This feedback signal can either be perfect [11]–[14] or noisy [15]. In the latter case, it was proposed in [15] to regard the feedback transmission as a separate communication problem for which optimized transmitter and receiver pairs can again be learned. The proposed training scheme in [15] alternates between optimizing the different transmitter/receiver pairs, with the intuition that training improvements for one pair lead to better training of the other pair (and vice versa). Thus, both communication systems improve simultaneously and continuously until some predefined stopping criterion is met (see Alg. 3 in [15]). The assumed feedback link in [15] only allowed for the transmission of real numbers over an additive white Gaussian noise (AWGN) channel. In practice, however, signals will be quantized to a finite number of bits, including the feedback signal. To the best of our knowledge, such quantization has not yet been considered in the literature. Studies on quantization have been conducted so far only in terms of the transmitter and receiver processing, for example when the corresponding learned models are implemented with finite resolution [16]–[20].

In this paper, we analyze the impact of quantization of the feedback signal on data-driven learning of physical-layer communication over an unknown channel. Compared to [15], the feedback transmission scheme is not learned. Instead, we show that due to the specific properties of the feedback signal, an adaptive scheme based on simple pre-processing steps followed by a fixed quantization strategy can lead to similar performance as compared to the case where unquantized feedback is used for training, even with 1-bit quantization. We provide a theoretical justification for the proposed approach and perform extensive simulations for both linear Gaussian and nonlinear phase-noise channels. The detailed contributions

J. Song, B. Peng, C. Häger, and H. Wymeersch are with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. email: jinxiang@student.chalmers.se, {bile.peng, christian.haeger, henkw}@chalmers.se. C. Häger is also with the Department of Electrical and Computer Engineering, Duke University, Durham, USA. A. Sahai is with the Department of Electrical Engineering and Computer Science, UC Berkeley, Berkeley, USA. The work of C. Häger was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 749798. H. Wymeersch was supported by the Swedish Research Council under grant No. 2018-03701.

[1] See [12, Sec. III-C] for a discussion about the relationship between the approaches in [11]–[13] and [14].

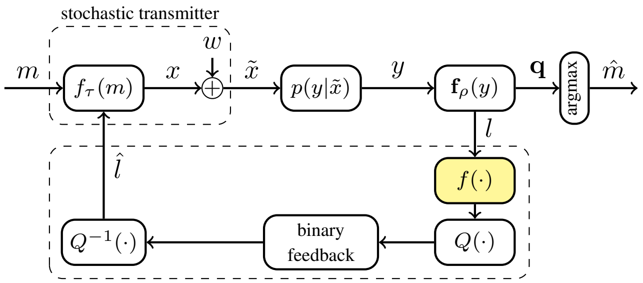


Fig. 1: Data-driven learning model where the discrete time index k (e.g., m_k) is omitted for all variables. The quantization and binary feedback is shown in the lower dashed box, while the proposed pre-processor is highlighted. Note that $w = 0$ for the receiver learning (Sec. III-A).

in this paper are as follows:

1) We propose a novel quantization method for feedback signals in data-driven learning of physical-layer communication. The proposed method addresses a major shortcoming in previous work, in particular the assumption in [15] that feedback losses can be transmitted as unquantized real numbers over an AWGN channel.
2) We conduct a thorough numerical study demonstrating the effectiveness of the proposed scheme. We investigate the impact of the number of quantization bits on the performance and the training process, showing that 1-bit quantization can provide performance similar to unquantized feedback. In addition, it is shown that the scheme is robust to noisy feedback where the quantized signal is perturbed by random bit flips.
3) We provide a theoretical justification for the effectiveness of the proposed approach in the form of Propositions 1 and 2. In particular, it is proved that feedback quantization and bit flips manifest themselves merely as a scaling of the expected gradient used for parameter training. Moreover, upper bounds on the variance of the gradient are derived in terms of the Fisher information matrix of the transmitter parameters.

Notation: Vectors will be denoted with lower case letters in bold (e.g., $\mathbf{x}$), with x_n or $[\mathbf{x}]_n$ referring to the n-th entry in $\mathbf{x}$; matrices will be denoted in bold capitals (e.g., $\mathbf{X}$); $\mathbb{E}(\{\mathbf{x}\}$ denotes the expectation operator; $\mathbb{V}(\mathbf{x})$ denotes the variance (the trace of the covariance matrix) of the random vector $\mathbf{x}$ (i.e., $\mathbb{V}\{\mathbf{x}\} = \mathbb{E}\{\mathbf{x}^\intercal \mathbf{x}\} - (\mathbb{E}\{\mathbf{x}\})^\intercal(\mathbb{E}\{\mathbf{x}\})$).

II. System Model

We wish to transmit messages $m \in \{1, \dots, M\}$ over an a priori unknown static memoryless channel which is defined by a conditional probability density function (PDF) $p(y|x)$, where $x, y \in \mathbb{C}$ and M is the total number of messages.[2] The communication system is implemented by representing the transmitter and receiver as two parameterized functions $f_\tau : \{1, \dots, M\} \to \mathbb{C}$ and $\mathbf{f}_\rho : \mathbb{C} \to [0,1]^M$, where $[a,b]^M$ is the M–fold Cartesian product of the $[a,b]$–interval (i.e., the elements in $[a,b]^M$ are vectors of length M with entries between a and b inclusively) and τ and ρ are sets of transmitter and receiver parameters, respectively. The transmitter maps the k-th message m_k to a complex symbol $x_k = f_\tau(m_k)$, where an average power constraint according to $\mathbb{E}\{|x_k|^2\} \leq P$ is assumed. The symbol x_k is sent over the channel and the receiver maps the channel observation y_k to a probability vector $\mathbf{q}_k = \mathbf{f}_\rho(y_k)$, where one may interpret the components of $\mathbf{q}_k$ as estimated posterior probabilities for each possible message. Finally, the receiver outputs an estimated message according to $\hat{m}_k = \arg\max_m [\mathbf{q}_k]_m$, where $[\mathbf{x}]_m$ returns the m-th component of $\mathbf{x}$. The setup is depicted in the top branch of the block diagram in Fig. 1, where the random perturbation w in the transmitter can be ignored for now.

We further assume that there exists a feedback link from the receiver to the transmitter, which, as we will see below, facilitates transmitter learning. In general, our goal is to learn optimal transmitter and receiver mappings f_τ and $\mathbf{f}_\rho$ using limited feedback.

III. Data-Driven Learning

In order to find good parameter configurations for τ and ρ, a suitable optimization criterion is required. Due to the reliance on gradient-based methods, conventional criteria such as the symbol error probability $\Pr(m_k \neq \hat{m}_k)$ cannot be used directly. Instead, it is common to minimize the expected cross-entropy loss defined by

$$\ell(\tau, \rho) \triangleq -\mathbb{E}\{\log([\mathbf{f}_\rho(y_k)]_{m_k})\}, \tag{1}$$

where the dependence of $\ell(\tau, \rho)$ on τ is implicit through the distribution of y_k.

A major practical hurdle is the fact that the gradient $\nabla_\tau \ell(\tau, \rho)$ cannot actually be evaluated because it requires a known and differentiable channel model. To solve this problem, we apply the alternating optimization approach proposed in [11], [12], which we briefly review in the following. For this approach, one alternates between optimizing first the receiver parameters ρ and then the transmitter parameters τ for a certain number of iterations N. To that end, it is assumed that the transmitter and receiver share common knowledge about a database of training data m_k.

A. Receiver Learning

For the receiver optimization, the transmitter parameters τ are assumed to be fixed. The transmitter maps a mini-batch of uniformly random training messages m_k, $k \in \{1, \dots, B_R\}$, to symbols satisfying the power constraint and transmits them over the channel. The receiver observes $y_1, \dots, y_{B_R}$ and generates B_R probability vectors $\mathbf{f}_\rho(y_1), \dots, \mathbf{f}_\rho(y_{B_R})$.

The receiver then updates its parameters ρ according to $\rho_{i+1} = \rho_i - \alpha_R \nabla_\rho \ell^{\mathrm{e}}_R(\rho_i)$, where

$$\ell^{\mathrm{e}}_R(\rho) = -\frac{1}{B_R} \sum_{k=1}^{B_R} \log([\mathbf{f}_\rho(y_k)]_{m_k}) \tag{2}$$

is the empirical cross-entropy loss associated with the mini-batch and α_R is the learning rate. This procedure is repeated iteratively for a fixed number of iterations N_R.

[2] In this paper, we restrict ourselves to two-dimensional (i.e., complex-valued) channel models, where the generalization to an arbitrary number of dimensions is straightforward.

B. Transmitter Learning

For the transmitter optimization, the receiver parameters are assumed to be fixed. The transmitter generates a mini-batch of uniformly random training messages m_k, $k \in \{1, \dots, B_T\}$, and performs the symbol mapping as before. However, before transmitting the symbols over the channel, a small Gaussian perturbation is applied, which yields $\tilde{x}_k = x_k + w_k$, where $w_k \sim \mathcal{CN}(0, \sigma_p^2)$ and reasonable choices for σ_p^2 are discussed in Sec. V. Hence, we can interpret the transmitter as stochastic, described by the PDF

$$\pi_\tau(\tilde{x}_k|m_k) = \frac{1}{\pi\sigma_p^2} \exp\left(-\frac{|\tilde{x}_k - f_\tau(m_k)|^2}{\sigma_p^2}\right). \tag{3}$$

Based on the received channel observations, the receiver then computes per-sample losses $l_k = -\log([\mathbf{f}_\rho(y_k)]_{m_k}) \in \mathbb{R}$ for $k \in \{1, \dots, B_T\}$, and feeds these back to the transmitter via the feedback link. The corresponding received losses are denoted by $\hat{l}_k$, where ideal feedback corresponds to $\hat{l}_k = l_k$. Finally, the transmitter updates its parameters τ according to $\tau_{i+1} = \tau_i - \alpha \nabla_\tau \ell_T^{\mathrm{e}}(\tau_i)$, where

$$\nabla_\tau \ell_T^{\mathrm{e}}(\tau) = \frac{1}{B_T} \sum_{k=1}^{B_T} \hat{l}_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k). \tag{4}$$

This procedure is repeated iteratively for a fixed number of iterations N_T, after which the alternating optimization continues again with the receiver learning. The total number of gradient steps in the entire optimization is given by $N(N_T + N_R)$.

A theoretical justification for the gradient in (4) can be found in [11]–[13]. In particular, it can be shown that the gradient of $\ell_T(\tau) = \mathbb{E}\{l_k\}$ is given by

$$\nabla_\tau \ell_T(\tau) = \mathbb{E}\{l_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\}, \tag{5}$$

where the expectations are over the message, transmitter, and channel distributions. Note that (4) is the corresponding sample average for finite mini-batches assuming $\hat{l}_k = l_k$.

Remark 1. As pointed out in previous work, the transmitter optimization can be regarded as a simple form of reinforcement learning. In particular, one may interpret the transmitter as an agent exploring its environment according to a stochastic exploration policy defined by (3) and receiving (negative) rewards in the form of per-sample losses. The state is the message m_k and the transmitted symbol $\tilde{x}_k$ is the corresponding action. The learning setup belongs to the class of *policy gradient methods*, which rely on optimizing parameterized policies using gradient descent. We will make use of the following well-known property of policy gradient learning:[3]

$$\mathbb{E}\{\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\} = 0. \tag{6}$$

C. Loss Transformation

The per-sample losses can be transformed through a preprocessing function $f : \mathbb{R} \to \mathbb{R}$, which is known as reward shaping in the context of reinforcement learning [21]. Possible examples for f include:

- Clipping: setting $f(l_k) = \min(\beta, l_k)$ is used to deal with large loss variations and stabilize training [22].
- Baseline: setting $f(l_k) = l_k - \beta$ is called a constant baseline [23] and is often used to reduce the variance of the Monte Carlo estimate of the stochastic gradient [21].
- Scaling: setting $f(l_k) = \beta l_k$ only affects the magnitude of the gradient step, but this can be compensated with methods using adaptive step sizes (including the widely used Adam optimizer [24]). However, aggressive scaling can adversely affect the performance [25], [26].

To summarize, it has been shown that training with transformed losses, i.e., assuming $\hat{l}_k = f(l_k)$ in (4), is quite robust and can even be beneficial in some cases (e.g., by reducing gradient variance through baselines). Hence, one may conclude that the training success is to a large extent determined by the relative ordering of the losses (i.e., the distinction between good actions and bad actions). In this paper, reward shaping is exploited for pre-processing before quantizing the transformed losses to a finite number of bits.

IV. Learning with Quantized Feedback

Previous work has mostly relied on ideal feedback, where $\hat{l}_k = l_k$ [11]–[14]. Robustness of learning with respect to additive noise according to $\hat{l}_k = l_k + n_k$, $n_k \sim \mathcal{N}(0, \sigma^2)$, was demonstrated in [15]. In this paper, we take a different view and assume that there only exists a *binary feedback channel* from the receiver to the transmitter. In this case, the losses must be quantized before transmission.

A. Conventional Quantization

Optimal Quantization: Given a distribution of the losses $p(l_k)$ and q bits that can be used for quantization, the mean squared quantization error is

$$D = \mathbb{E}\{(l_k - Q(l_k))^2\}. \tag{7}$$

With q bits, there are 2^q possible quantization levels which can be optimized to minimize D, e.g., using the Lloyd-Max algorithm [27].

Adaptive Quantization: In our setting, the distribution of the per-sample losses varies over time as illustrated in Fig. 2. For non-stationary variables, adaptive quantization can be used. The source distribution can be estimated based on a finite number of previously seen values and then adapted based on the Lloyd-Max algorithm. If the source and sink adapt based on quantized values, no additional information needs to be exchanged. If adaptation is performed based on unquantized samples, the new quantization levels need to be conveyed from the source to the sink. In either case, a sufficient number of realizations are needed to accurately estimate the loss distribution and the speed of adaptation is fixed.

Fixed Quantization: We aim for a strategy that does not require overhead between transmitter and receiver. A simple non-adaptive strategy is to apply a fixed quantization. Under

[3] To see this, one may first apply $\nabla_\tau \log \pi_\tau = \frac{\nabla_\tau \pi_\tau}{\pi_\tau}$ and then use the fact that $\int \nabla_\tau \pi_\tau(\tilde{x}|m) \mathrm{d}\tilde{x} = 0$ since $\int \pi_\tau(\tilde{x}|m) \mathrm{d}\tilde{x} = 1$.

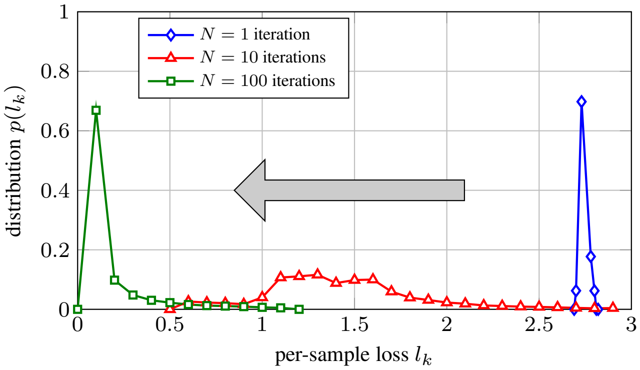


Fig. 2: Illustration of the non-stationary loss distribution as a function of the number of training iterations in the alternating optimization.

fixed quantization, we divide up the range $[0, \bar{l}]$ into $2^q - 1$ equal-size regions of size $\Delta = \bar{l}/2^q$ so that

$$Q(l) = \frac{\Delta}{2} + \Delta \left\lfloor \frac{l}{\Delta} \right\rfloor . \tag{8}$$

Here, $\bar{l}$ is the largest loss value of interest. The corresponding thresholds are located at $m\bar{l}/2^q$, where $m \in \{1, \ldots, 2^q - 1\}$. Hence, the function $Q(l)$ and its inverse $Q^{-1}(l)$ are fully determined by $\bar{l}$ and the number of bits q.

B. Proposed Quantization

Given the fact that losses can be transformed without much impact on the optimization, as described in Sec. III-C, we propose a novel strategy that employs adaptive pre-processing followed by a fixed quantization scheme. The proposed method operates on mini-batches of size B_T. In particular, the receiver (source) applies the following steps:

1) Clipping: we clip the losses to lie within a range $[l_{\text{min}}, l_{\text{max}}]$. Here, l_{min} is the smallest loss in the current mini-batch, while l_{max} is chosen such that the 5% largest losses in the mini-batch are clipped. This effectively excludes very large per-sample losses which may be regarded as outliers. We denote this operation by $f_{\text{clip}}(\cdot)$.
2) Baseline: we then shift the losses with a fixed baseline l_{min}. This ensures that all losses are within the range $[0, l_{\text{max}} - l_{\text{min}}]$. We denote this operation by $f_{\text{bl}}(\cdot)$.
3) Scaling: we scale all the losses by $1/(l_{\text{max}} - l_{\text{min}})$, so that they are within the range $[0, 1]$. We denote this operation by $f_{\text{sc}}(\cdot)$.
4) Fixed quantization: finally, we use a fixed quantization with q bits and send $Q(\tilde{l}_k)$, where $Q(\cdot)$ is defined in (7) and $\tilde{l}_k = f(l_k) = f_{\text{sc}}(f_{\text{bl}}(f_{\text{clip}}(l_k)))$, i.e., $f \triangleq f_{\text{sc}} \circ f_{\text{bl}} \circ f_{\text{clip}}$ denotes the entire pre-processing. For simplicity, a natural mapping of quantized losses to bit vectors $\mathbb{B}^q$ is assumed where quantization levels are mapped in ascending order to $(0, \ldots, 0, 0)^\intercal$, $(0, \ldots, 0, 1)^\intercal$, ..., $(1, \ldots, 1, 1)^\intercal$. In general, one may also try to optimize the mapping of bit vectors to the quantization levels in order to improve the robustness of the feedback transmission.

The transmitter (sink) has no knowledge of the functions $f_{\text{clip}}(\cdot)$, $f_{\text{bl}}(\cdot)$, or $f_{\text{sc}}(\cdot)$, and interprets the losses as being in the interval $[0, 1]$. It thus applies $\hat{l}_k = Q^{-1}(\tilde{l}_k) \in [0, 1]$ and uses the values $\hat{l}_k$ in (4). We note that some aspects of this approach are reminiscent of the Pop-Art algorithm from [28], where shifting and scaling are used to address non-stationarity during learning. In particular, Pop-Art can be used for general supervised learning, where the goal is to fit the outcome of a parameterized function (e.g., a neural network) to given targets (e.g., labels) by minimizing a loss function. Pop-Art adaptively normalizes the targets in order to deal with large magnitude variations and also address non-stationary targets. However, Pop-Art and the proposed method are different algorithms that have been proposed in different contexts, e.g., Pop-Art does not deal with quantization issues during learning.

In terms of complexity overhead, the proposed method requires one sorting operation in order to identify and clip the largest losses in each mini-batch (step 1). The baseline and scaling (steps 2 and 3) can be implemented with one real addition followed by one real multiplication. Finally, the quantizer can be implemented by using a look-up table approach. At the transmitter side (sink), the method only requires the dequantization step, which again can be implemented using a look-up table.

C. Impact of Feedback Quantization

The effect of quantization can be assessed via the Bussgang Theorem [29], which is a generalization of MMSE decomposition. If we assume $l_k \sim p(l)$ with mean μ_l and variance σ_l^2, then

$$Q(l_k) = g l_k + w_k, \tag{9}$$

in which $g \in \mathbb{R}$ is the Bussgang gain and w_k is a random variable, uncorrelated with l_k, provided we set

$$g = \frac{\mathbb{E}\{l_k Q(l_k)\} - \mu_l \mathbb{E}\{Q(l_k)\}}{\sigma_l^2}. \tag{10}$$

In general, the distribution of w_k may be hard (or impossible) to derive in closed form. Note that the mean of w_k is $\mathbb{E}\{Q(l_k)\} - g\mu_l$ and the variance is $\mathbb{V}\{Q(l_k)\} - g^2\sigma_l^2$. When the number of quantization bits q increases, $Q(l_k) \to l_k$ and thus $g \to 1$.

If we replace l_k with $Q(l_k)$ in (5), denote the corresponding gradient function by $\nabla_\tau \ell_T^{\text{q}}(\tau)$, and substitute (9), then the following proposition holds.

Proposition 1. *Let* $\boldsymbol{\gamma}_k = l_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)$, $l_k \in [0, 1]$, *with* $\nabla_\tau \ell_T(\tau) = \mathbb{E}\{\boldsymbol{\gamma}_k\}$, *and* $\boldsymbol{\gamma}_k^{\text{q}} = Q(l_k)\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)$, *then*

$$\mathbb{E}\{\boldsymbol{\gamma}_k^{\text{q}}\} = \nabla_\tau \ell_T^{\text{q}}(\tau) = g\nabla_\tau \ell_T(\tau) \tag{11}$$

$$\mathbb{V}\{\boldsymbol{\gamma}_k^{\text{q}}\} \leq g^2 \mathbb{V}\{\boldsymbol{\gamma}_k\} + (g\bar{w} + \bar{w}^2)\text{tr}\{\mathbf{J}(\tau)\} \tag{12}$$

where $\mathbf{J}(\tau) = \mathbb{E}\{\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k) \nabla_\tau^\intercal \log \pi_\tau(\tilde{x}_k|m_k)\} \succeq 0$ *is the Fisher information matrix of the transmitter parameters* τ *and* $\bar{w} = \max_l |gl - Q(l)| = |1 - 1/2^{q-1} - g|$ *is a measure of the maximum quantization error.*

Proof: See Appendix. ■

Hence, the impact of quantization, under a sufficiently large mini-batch size is a scaling of the expected gradient. Note that this scaling will differ for each mini-batch. The variance is affected in two ways: a scaling with g^2 and an additive term that depends on the maximum quantization error and the Fisher information at τ. When q increases, $g \to 1$ and $\bar{w} \to 0$, so that $\mathbb{V}\{\gamma_k^{\mathrm{q}}\} \to \mathbb{V}\{\gamma_k\}$, as expected.

In general, the value of g is hard to compute in closed form, but for 1-bit quantization and a Gaussian loss distribution, (10) admits a closed-form solution.[4] In particular,

$$g = \begin{cases} 1/\sqrt{8\pi\sigma_l^2} & \mu_l = 1/2 \\ e^{-1/(8\sigma_l^2)}/\sqrt{8\pi\sigma_l^2} & \mu_l \in \{0, 1\}. \end{cases} \qquad (13)$$

In light of the distributions from Fig. 2, we observe that (after loss transformation) for most iterations, $\mu_l \approx 1/2$ and σ_l^2 will be moderate (around $1/(8\pi)$), leading to $g \approx 1$. Only after many iterations $\mu_l < 1/2$ and σ_l^2 will be small, leading to $g \ll 1$. Hence, for sufficiently large batch sizes, 1-bit quantization should not significantly affect the learning convergence rate.

D. Impact of Noisy Feedback Channels

For the proposed pre-processing and quantization scheme, distortions are introduced through the function $f(\cdot)$ (in particular the clipping) and the quantizer $Q(\cdot)$. Moreover, additional impairments may be introduced when the quantized losses are transmitted over a noisy feedback channel. We will consider the case where the feedback channel is a binary symmetric channel with flip probability $p \in [0, 1/2)$. Our numerical results (see Sec. V-B4) indicate that the learning process is robust against such distortions, even for very high flip probabilities. In order to explain this behavior, it is instructive to first consider the case where the transmitted per-sample losses are entirely random and completely unrelated to the training data. In that case, one finds that

$$\mathbb{E}\{\hat{l}_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\} = \mathbb{E}\{\hat{l}_k\}\mathbb{E}\left\{\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\right\} = 0$$

regardless of the loss distribution or quantization scheme. The interpretation is that for large mini-batch sizes, random losses simply "average out" and the applied gradient in (4) is close to zero. We can exploit this behavior and make the following statement.

Proposition 2. *Let* $\gamma_k^{\mathrm{e}} = \hat{l}_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)$ *where the binary version of* $Q(l_k)$ *has been subjected to a binary symmetric channel with flip probability* p *to yield* $\hat{l}_k$*. Then, for* 1*-bit and* 2*-bit quantization with a natural mapping of bit vectors to quantized losses, we have*

$$\mathbb{E}\{\gamma_k^{\mathrm{e}}\} = \nabla_\tau \ell_T^{\mathrm{e}}(\tau) = (1-2p)\nabla_\tau \ell_T^{\mathrm{q}}(\tau).$$

Moreover, for 1*-bit quantization,*

$$\mathbb{V}\{\gamma_k^{\mathrm{e}}\} \leq \mathbb{V}\{\gamma_k^{\mathrm{q}}\} + 4p(1-p)\|\nabla_\tau \ell_T^{\mathrm{q}}(\tau)\|^2 + p\,\mathrm{tr}\{\mathbf{J}(\tau)\}.$$

Proof: See Appendix. ■

Hence, for a sufficiently large mini-batch size, the gradient is simply scaled by a factor $1-2p$. This means that even under very noisy feedback, learning should be possible.

Remark 2. Note that when using small mini-batches, the empirical gradients computed via (4) will deviate from the expected value $(1-2p)\nabla_\tau \ell_T^{\mathrm{q}}(\tau)$: they will not be scaled exactly by $1-2p$ and they will be perturbed by the average value of $p\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)$. Hence, robustness against large p can only be offered for large mini-batch sizes.

V. Numerical Results

In this section, we provide extensive numerical results to verify and illustrate the effectiveness of the proposed loss quantization scheme. In the following, the binary feedback channel is always assumed to be noiseless except for the results presented in Sec. V-B4.[5]

A. Setup and Parameters

1) Channel Models: We consider two memoryless channel models $p(y|x)$: the standard AWGN channel $y = x + n$, where $n \sim \mathcal{CN}(0, \sigma^2)$, and a simplified memoryless fiber-optic channel which is defined by the recursion

$$x_{i+1} = x_i e^{\jmath L\gamma|x_i|^2/K} + n_{i+1}, \quad 0 \leq i < K, \qquad (14)$$

where $x_0 = x$ is the channel input, $y = x_K$ is the channel output, $n_{i+1} \sim \mathcal{CN}(0, \sigma^2/K)$, L is the total link length, σ^2 is the noise power, and $\gamma \geq 0$ is a nonlinearity parameter. Note that this channel reverts to the AWGN channel when $\gamma = 0$. For our numerical analysis, we set $L = 5000\,\mathrm{km}$, $\gamma = 1.27\,\mathrm{rad/W/km}$, $K = 50$, and $\sigma^2 = -21.3\,\mathrm{dBm}$, which are the same parameters as in [6], [12], [30]. For both channels, we define SNR $\triangleq P/\sigma^2$. Since the noise power is assumed to be fixed, the SNR is varied by varying the signal power P.

The model in (14) assumes ideal distributed amplification across the optical link and is obtained from the nonlinear Schrödinger equation by neglecting dispersive effects, see, e.g., [31] for more details about the derivation. Because dispersive effects are ignored, the model does not necessarily reflect the actual channel conditions in realistic fiber-optic transmission. The main interest in this model stems from its simplicity and analytical tractability while still capturing some realistic nonlinear effects, in particular the nonlinear phase noise. The model has been studied intensively in the literature, including detection schemes [32]–[34], signal constellations [33], [35], capacity bounds [30], [31], [36], [37], and most recently also in the context of machine learning [6], [12]. In the following, we refer to the model as the nonlinear phase-noise channel to highlight the fact that it should not be seen as an accurate model for fiber-optic transmission.

2) Transmitter and Receiver Networks: Following previous work, the functions f_τ and $\mathbf{f}_\rho$ are implemented as multi-layer neural networks. A message m is first mapped to a M–dimensional "one-hot" vector where the m–th element is 1 and all other elements are 0. Each neuron takes inputs from the previous layer and generates an output according to a learned linear mapping followed by a fixed nonlinear activation

[4] For Gaussian losses, $\bar{w}$ in Proposition 1 is not defined. The proposition can be modified to deal with unbounded losses.

[5] TensorFlow source code is available at https://github.com/henkwymeersch/quantizedfeedback.

TABLE I: Neural network parameters, where $M = 16$

	transmitter f_τ			receiver $\mathbf{f}_\rho$		
layer	1	2-3	4	1	2-3	4
number of neurons	M	30	2	2	50	M
activation function	-	ReLU	linear	-	ReLU	softmax

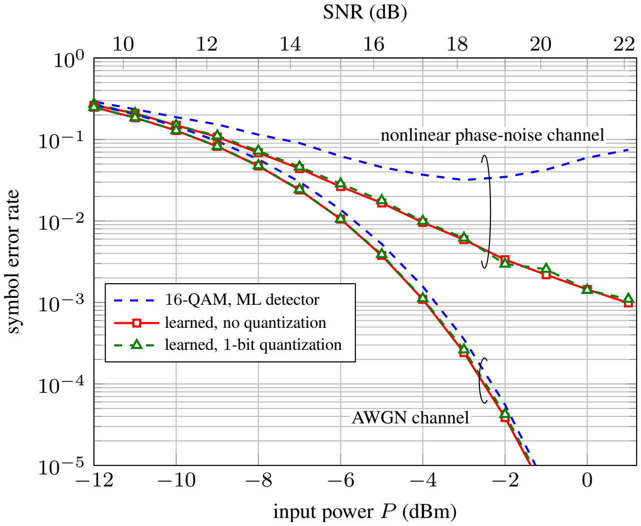
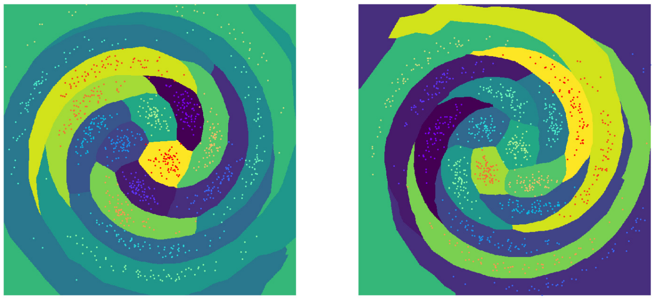


Fig. 3: Symbol error rate achieved for $M = 16$. The training SNR is 15 dB for the AWGN channel, whereas training is done separately for each input power (i.e., SNR) for the nonlinear phase-noise channel.

function. The final two outputs of the transmitter network are normalized to ensure $1/B \sum_{k=1}^{B} |x_k|^2 = P$, $B \in \{B_T, B_R\}$, and then used as the channel input. The real and imaginary parts of the channel observation serve as the input to the receiver network. All network parameters are summarized in Table I, where $M = 16$.

3) Training Procedure: For the alternating optimization, we first fix the transmitter and train the receiver for $N_R = 30$ iterations with a mini-batch size of $B_R = 64$. Then, the receiver is fixed and the transmitter is trained for $N_T = 20$ iterations with $B_T = 64$. This procedure is repeated $N = 4000$ times for the AWGN channel. For the nonlinear phase-noise channel, we found that more iterations are typically required to converge, especially at high input powers, and we consequently set $N = 6000$. The Adam optimizer is used to perform the gradient updates, where $\alpha_T = 0.001$ and $\alpha_R = 0.008$. The reason behind the unequal number of training iterations for the transmitter and receiver is that the receiver network is slightly bigger than the transmitter network and thus requires more training iterations to converge.

4) Transmitter Exploration Variance: We found that the parameter σ_p^2 has to be carefully chosen to ensure successful training. In particular, choosing σ_p^2 too small will result in insufficient exploration and slow down the training process. On the other hand, if σ_p^2 is chosen too large, the resulting noise may in fact be larger than the actual channel noise, resulting in many falsely detected messages and unstable training. In our simulations, we use $\sigma_p^2 = P \cdot 10^{-3}$.



(a)

(b)

Fig. 4: Learned decision regions for the nonlinear phase-noise channel, $M = 16$, and $P = -3$ dBm (a) without quantizing per-sample losses and (b) using the proposed quantization scheme and 1-bit quantization.

B. Results and Discussion

1) Perfect vs Quantized Feedback: We start by evaluating the impact of quantized feedback on the system performance, measured in terms of the symbol error rate (SER). For the AWGN channel, the transmitter and receiver are trained for a fixed SNR = 15 dB (i.e., $P = -6.3$ dBm such that SNR $= P/\sigma^2 = -6.3$ dBm $+$ 21.3 dBm $=$ 15 dB) and then evaluated over a range of SNRs by changing the signal power (similar to, e.g., [12]). For the nonlinear phase-noise channel, this approach cannot be used because optimal signal constellations and receivers are highly dependent on the transmit power.[6] Therefore, a separate transmitter–receiver pair is trained for each input power P. Fig. 3 shows the achieved SER assuming both perfect feedback without quantization and a 1-bit feedback signal based on the proposed method. For both channels, the resulting communication systems with 1-bit feedback quantization have very similar performance to the scenario where perfect feedback is used for training, indicating that the feedback quantization does not significantly affect the learning process. As a reference, the performance of standard 16-QAM with a maximum-likelihood (ML) detector is also shown. The ML detector makes a decision according to

$$\hat{x}_{\text{ML}} = \underset{m \in \{1,\dots,M\}}{\arg\max} \; p(y|s_m), \tag{15}$$

where $s_1, \dots, s_M$ are all constellation points. For the nonlinear phase-noise channel, the channel likelihood $p(y|x)$ can be derived in closed form, see [32, p. 225]. For the AWGN channel, (15) is equivalent to a standard minimum Euclidean-distance detector. The learning approach outperforms this baseline for both channels, which is explained by the fact that the transmitter neural network learns better modulation formats (i.e., signal constellations) compared to 16-QAM.

Fig. 4 visualizes the learned decision regions for the quantized (right) and unquantized (left) feedback schemes assuming the nonlinear phase-noise channel with $P = -3$ dBm. Only slight differences are observed which can be largely attributed to the randomness of the training process.

[6] In principle, the optimal signal constellation may also depend on the SNR for the AWGN channel.

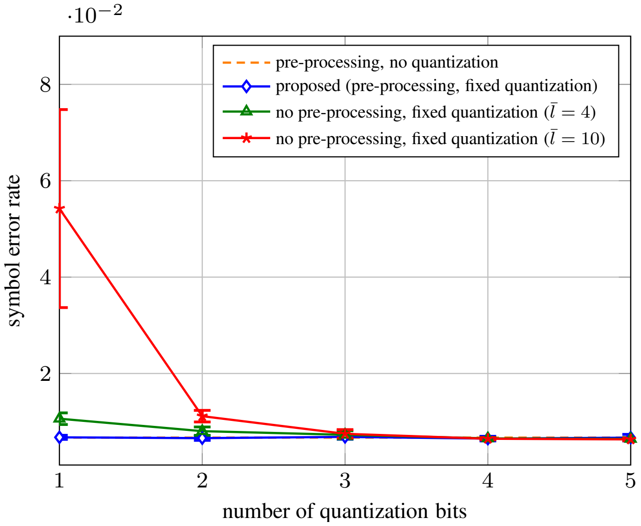


Fig. 5: Impact of the number of quantization bits on the achieved performance for the nonlinear phase-noise channel with $M = 16$, $P = -3$ dBm. Results are averaged over 10 different training runs where error bars indicate the standard deviation between the runs.

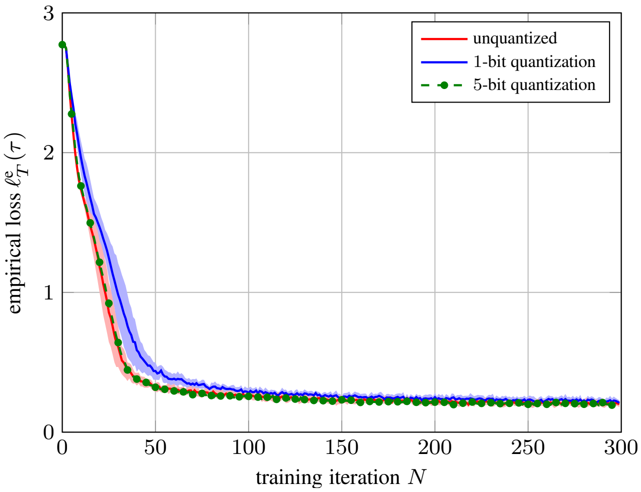


Fig. 6: Evolution of $\ell^{\text{e}}_{T}(\tau)$ during the alternating optimization for the nonlinear phase-noise channel with $M = 16$, $P = -3$ dBm. Results are averaged over 15 different training runs where the shaded area indicates one standard deviation between the runs.

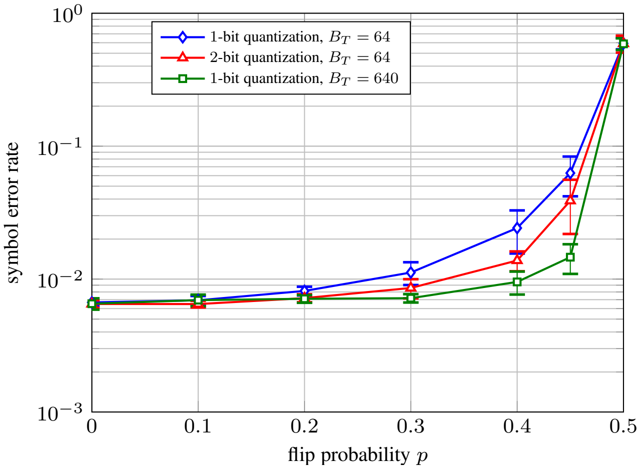


Fig. 7: Performance on the nonlinear phase-noise channel with $M = 16$, $P = -3$ dBm when transmitting quantized losses over a noisy feedback channel modeled as a binary symmetric channel with flip probability p. Results are average over 10 runs where the error bars indicate one standard deviation between runs.

2) Impact of Number of Quantization Bits: Next, the nonlinear phase-noise channel for a fixed input power $P = -3$ dBm is considered to numerically evaluate the impact of the number of quantization bits on the performance. Fig. 5 shows the achieved SER when different schemes are used for quantizing the per-sample losses. For a fixed quantization scheme without pre-processing (see Sec. IV-A), the performance of the trained system is highly sensitive to the number of quantization bits and the assumed quantization range $[0, \bar{l}]$. For $\bar{l} = 10$ with 1 quantization bit, the system performance deteriorates noticeably and the training outcome becomes unstable, as indicated by the error bars (which are averaged over 10 different training runs). For the proposed quantization scheme, the performance of the trained system is (i) essentially independent on the number of bits used for quantization and (ii) virtually indistinguishable from a system trained with unquantized feedback.

3) Impact on Convergence Rate: In Fig. 6, we show the evolution of the empirical cross-entropy loss $\ell^{\text{e}}_{T}(\tau)$ during the alternating optimization for the nonlinear phase-noise channel with $P = -3$ dBm. It can be seen that quantization manifests itself primarily in terms of a slightly decreased convergence rate during training. For the scenario where per-sample losses are quantized with 5 bits, the empirical losses $\ell^{\text{e}}_{T}(\tau)$ converged after about 160 iterations, which is the same as in the case of un-quantized feedback. For 1-bit quantization, the training converges slightly slower, after around 200 iterations, which is a minor degradation compared to the entire training time. However, the slower convergence rate implies that it is harder to deal with changes in the channel. Hence, with 1-bit quantization, the coherence time should be longer compared to with unquantized feedback.

4) Impact of Noisy Feedback: In order to numerically evaluate the effect of noise during the feedback transmission, we consider again the nonlinear phase-noise channel for a fixed input power$P = -3$ dBm. Fig. 7 shows the achieved SER when transmitting the quantized per-sample losses over a binary symmetric channel with flip probability p (see Sec. IV-D). It can be seen that the proposed quantization scheme is highly robust to the channel noise. For the assumed mini-batch size $B_T = 64$, performance starts to decrease only for very high flip probabilities and remains essentially unchanged for $p < 0.1$ with 1-bit quantization and for $p < 0.2$ with 2-bit quantization. A theoretical justification for this behavior is provided in Proposition 2, which states that the channel noise manifests itself only as a scaling of the expected gradient. Thus, one may also expect that the learning process can withstand even higher flip probabilities by simply increasing the mini-batch size. Indeed, Fig. 7 shows that when increasing the mini-batch size from $B_T = 64$ to $B_T = 640$, the noise tolerance for 1-bit quantization increases significantly and performance remains unchanged for flip probabilities as high as $p = 0.3$.

Note that for $p = 0.5$, the achieved SER is slightly better than $(M-1)/M \approx 0.938$ corresponding to random guessing. This is because the receiver learning is still active, even though the transmitter only performs random explorations.

VI. Conclusions

We have proposed a novel method for data-driven learning of physical-layer communication in the presence of a binary feedback channel. Our method relies on an adaptive clipping, shifting, and scaling of losses followed by a fixed quantization at the receiver, and a fixed reconstruction method at the transmitter. We have shown that the proposed method (i) can lead to good performance even under 1-bit feedback; (ii) does not significantly affect the convergence speed of learning; and (iii) is highly robust to noise in the feedback channel.

The proposed method can be applied beyond physical-layer communication, to reinforcement learning problems in general, and distributed multi-agent learning in particular.

Appendix

Proof of Proposition 1

The mean of $\boldsymbol{\gamma}_k^{\mathrm{q}}$ can be computed as

$$\begin{aligned}
&\mathbb{E}\{\boldsymbol{\gamma}_k^{\mathrm{q}}\} = \nabla_\tau \ell_T^{\mathrm{q}}(\tau)\\
&= \mathbb{E}\{Q(l_k)\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\}\\
&= g\mathbb{E}\{l_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\} + \mathbb{E}\{w_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\}\\
&= g\mathbb{E}\{l_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\} + \mathbb{E}\{w_k\}\mathbb{E}\{\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\}\\
&= g\mathbb{E}\{l_k \nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\} = g\nabla_\tau \ell_T(\tau).
\end{aligned}$$

We have made use of the fact that w_k is uncorrelated with l_k and that (6) holds. The variance can similarly be bounded as follows:

$$\begin{aligned}
&\mathbb{V}\{\boldsymbol{\gamma}_k^{\mathrm{q}}\}\\
&= \mathbb{E}\{(Q(l_k))^2\|\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\|^2\} - g^2\|\nabla_\tau \ell_T(\tau)\|^2\\
&= g^2\mathbb{E}\{l_k^2\|\nabla \log \pi_\tau(x_k|m_k)\|^2\} - g^2\|\nabla_\tau \ell_T(\tau)\|^2\\
&+ \mathbb{E}\{w_k^2\|\nabla \log \pi_\tau(x_k|m_k)\|^2\}\\
&+ 2\mathbb{E}\{g l_k w_k\|\nabla \log \pi_\tau(x_k|m_k)\|^2\}\\
&\leq g^2\mathbb{V}\{\boldsymbol{\gamma}_k\} + \bar{w}^2\mathrm{tr}\{\mathbf{J}(\tau)\}\\
&- 2g\mathbb{E}\{w_k l_k\|\nabla \log \pi_\tau(\tilde{x}_k|m_k)\|^2\}\\
&\leq g^2\mathbb{V}\{\boldsymbol{\gamma}_k\} + \bar{w}^2\mathrm{tr}\{\mathbf{J}(\tau)\} + 2g\bar{w}\mathrm{tr}\{\mathbf{J}(\tau)\}
\end{aligned}$$

We have made use of $-w_k l_k = l_k(g l_k - Q(l_k)) \leq \max_{l_k} |g l_k - Q(l_k)| = \bar{w}$, that $l_k \leq 1$, and that $\mathrm{tr}\{\mathbf{J}(\tau)\} = \mathbb{E}\{\|\nabla \log \pi_\tau(x_k|m_k)\|^2\}$.

Proof of Proposition 2

For the proposed adaptive pre-processing and fixed 1-bit quantization, the quantized losses l_k are either $\Delta/2 = 1/4$ or $1 - \Delta/2 = 3/4$. Assuming transmission over the binary symmetric channel, the gradient in (5) can be written as

$$\nabla_\tau \ell_T^{\mathrm{e}}(\tau) = \mathbb{E}\{Q(l_k)^{1-n_k}(1-Q(l_k))^{n_k}\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\},$$

where n_k are independent and identically distributed Bernoulli random variables with parameter p. Since n_k is independent of all other random variables, we can compute

$$\mathbb{E}[Q(l_k)^{1-n_k}(1-Q(l_k))^{n_k} \,|\, Q(l_k)] = (1-2p)Q(l_k) + p.$$

Hence,

$$\begin{aligned}
&\mathbb{E}\{\boldsymbol{\gamma}_k^{\mathrm{e}}\} = \nabla_\tau \ell_T^{\mathrm{e}}(\tau)\\
&= \mathbb{E}\{((1-2p)Q(l_k) + p)\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\}\\
&= (1-2p)\mathbb{E}\{Q(l_k)\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\} + p\mathbb{E}\{\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\}\\
&= (1-2p)\nabla_\tau \ell_T^{\mathrm{q}}(\tau),
\end{aligned}$$

where the last step follows from (6). For 2-bit quantization, the possible values are $\Delta/2 = 1/8$ (corresponding to bits 00), $3\Delta/2 = 3/8$ (corresponding to 01), $1 - 3\Delta/2 = 5/8$ (corresponding to 10), $1 - \Delta/2 = 7/8$ (corresponding to 11). It then follows that when the transmitted loss is $Q(l_k)$, the received loss is

$$\begin{aligned}
Q(l_k) &\text{ with prob. } (1-p)^2\\
1 - Q(l_k) &\text{ with prob. } p^2\\
\text{other} &\text{ with prob. } p(1-p)
\end{aligned}$$

so that the expected received loss is $(1-2p)Q(l_k) + p$.

The variance under 1-bit quantization can be computed as

$$\begin{aligned}
&\mathbb{V}\{\boldsymbol{\gamma}_k^{\mathrm{e}}\}\\
&= \mathbb{E}\{(\boldsymbol{\gamma}_k^{\mathrm{e}})^2\} - (1-2p)^2\|\nabla_\tau \ell_T^{\mathrm{q}}(\tau)\|^2\\
&= \mathbb{E}\{(Q(l_k))^{2(1-n_k)}(1-Q(l_k))^{2n_k}\|\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\|^2\}\\
&- (1-2p)^2\|\nabla_\tau \ell_T^{\mathrm{q}}(\tau)\|^2\\
&= \mathbb{E}\{Q^2(l_k)\|\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\|^2\} + p\mathbb{E}\{\|\nabla \log \pi_\tau(\tilde{x}_k|m_k)\|^2\}\\
&- 2p\mathbb{E}\{Q(l_k)\|\nabla \log \pi_\tau(\tilde{x}_k|m_k)\|^2\} - (1-2p)^2\|\nabla_\tau \ell_T^{\mathrm{q}}(\tau)\|^2\\
&= \mathbb{V}\{\boldsymbol{\gamma}_k^{\mathrm{q}}\} + 4p(1-p)\|\nabla_\tau \ell_T^{\mathrm{q}}(\tau)\|^2 + p\mathrm{tr}\{\mathbf{J}(\tau)\}\\
&- 2p\mathbb{E}\{Q(l_k)\|\nabla_\tau \log \pi_\tau(\tilde{x}_k|m_k)\|^2\}\\
&\leq \mathbb{V}\{\boldsymbol{\gamma}_k^{\mathrm{q}}\} + 4p(1-p)\|\nabla_\tau \ell_T^{\mathrm{q}}(\tau)\|^2 + p\mathrm{tr}\{\mathbf{J}(\tau)\},
\end{aligned}$$

where the last step holds since $Q(l_k) \geq 0$.

References

[1] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO Detection," in *Proc. IEEE Int. Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017.

[2] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep Learning Methods for Improved Decoding of Linear Codes," *IEEE J. Sel. Topics Signal Proc.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.

[3] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Trans. on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[4] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep Learning-Based Communication Over the Air," *IEEE J. Sel. Topics Signal Proc.*, vol. 12, no. 1, pp. 132–143, Feb. 2017.

[5] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4843–4855, 2018.

[6] S. Li, C. Häger, N. Garcia, and H. Wymeersch, "Achievable information rates for nonlinear fiber communication via end-to-end autoencoder learning," in *Proc. European Conf. Optical Communication (ECOC)*, Rome, Italy, 2018.

[7] R. T. Jones, T. A. Eriksson, M. P. Yankov, and D. Zibar, "Deep Learning of Geometric Constellation Shaping including Fiber Nonlinearities," in *Proc. European Conf. Optical Communication (ECOC)*, Rome, Italy, 2018.

[8] H. Lee, I. Lee, and S. H. Lee, "Deep learning based transceiver design for multi-colored VLC systems," *Opt. Express*, vol. 26, no. 5, pp. 6222–6238, Mar. 2018.

[9] T. J. O'Shea, T. Roy, and N. West, "Approximating the Void: Learning Stochastic Channel Models from Observation with Variational Generative Adversarial Networks," *arXiv:1805.06350*, 2018.

[10] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel Agnostic End-to-End Learning based Communication Systems with Conditional GAN," *arXiv:1807.00447*, 2018.

[11] F. A. Aoudia and J. Hoydis, "End-to-End Learning of Communications Systems Without a Channel Model," *arXiv:1804.02276*, 2018.

[12] ——, "Model-free Training of End-to-end Communication Systems," *arXiv:1812.05929*, 2018.

[13] C. de Vrieze, S. Barratt, D. Tsai, and A. Sahai, "Cooperative Multi-Agent Reinforcement Learning for Low-Level Wireless Communication," *arXiv:1801.04541*, 2018.

[14] V. Raj and S. Kalyani, "Backpropagating Through the Air: Deep Learning at Physical Layer Without Channel Models," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2278–2281, Nov. 2018.

[15] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep Reinforcement Learning Autoencoder with Noisy Feedback," *arXiv:1810.05419*, 2018.

[16] M. Kim, W. Lee, J. Yoon, and O. Jo, "Building Encoder and Decoder with Deep Neural Networks: On the Way to Reality," *arXiv:1808.02401*, 2018.

[17] Z.-L. Tang, S.-M. Li, and L.-J. Yu, "Implementation of Deep Learning-based Automatic Modulation Classifier on FPGA SDR Platform," *Electronics*, vol. 7, no. 7, p. 122, 2018.

[18] C.-F. Teng, C.-H. Wu, K.-S. Ho, and A.-Y. Wu, "Low-complexity Recurrent Neural Network-based Polar Decoder with Weight Quantization Mechanism," *arXiv:1810.12154*, 2018.

[19] C. Fougstedt, C. Häger, L. Svensson, H. D. Pfister, and P. Larsson-Edefors, "ASIC Implementation of Time-Domain Digital Backpropagation with Deep-Learned Chromatic Dispersion Filters," in *Proc. European Conf. Optical Communication (ECOC)*, Rome, Italy, 2018.

[20] F. A. Aoudia and J. Hoydis, "Towards Hardware Implementation of Neural Network-based Communication Algorithms," *arXiv:1902.06939*, 2019.

[21] A. Y. Ng, D. Harada, and S. J. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 1999, pp. 278–287.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015.

[25] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine, "Q-prop: Sample-efficient policy gradient with an off-policy critic," *arXiv preprint arXiv:1611.02247*, 2016.

[26] R. Islam, P. Henderson, M. Gomrokchi, and D. Precup, "Reproducibility of benchmarked deep reinforcement learning tasks for continuous control," *arXiv preprint arXiv:1708.04133*, 2017.

[27] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[28] H. P. van Hasselt, A. Guez, M. Hessel, V. Mnih, and D. Silver, "Learning values across many orders of magnitude," in *Advances in Neural Information Processing Systems*, 2016, pp. 4287–4295.

[29] H. Rowe, "Memoryless nonlinearities with Gaussian inputs: Elementary results," *The BELL system technical Journal*, vol. 61, no. 7, pp. 1519–1525, 1982.

[30] K. Keykhosravi, G. Durisi, and E. Agrell, "Accuracy Assessment of Nondispersive Optical Perturbative Models through Capacity Analysis," *Entropy*, vol. 21, no. 8, pp. 1–19, aug 2019.

[31] M. I. Yousefi and F. R. Kschischang, "On the per-sample capacity of nondispersive optical fibers," *IEEE Trans. Inf. Theory*, vol. 57, no. 11, pp. 7522–7541, November 2011.

[32] K.-P. Ho, *Phase-modulated Optical Communication Systems*. Springer, 2005.

[33] A. P. Lau and J. M. Kahn, "16-QAM Signal Design and Detection in Presence of Nonlinear Phase Noise." IEEE, July 2007, pp. 53–54.

[34] A. S. Tan, H. Wymeersch, P. Johannisson, E. Agrell, P. Andrekson, and M. Karlsson, "An ml-based detector for optical communication in the presence of nonlinear phase noise," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.

[35] C. Häger, A. Graell i Amat, A. Alvarado, and E. Agrell, "Design of APSK Constellations for Coherent Optical Channels with Nonlinear Phase Noise," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3362–3373, August 2013.

[36] K. S. Turitsyn, S. A. Derevyanko, I. V. Yurkevich, and S. K. Turitsyn, "Information Capacity of Optical Fiber Channels with Zero Average Dispersion," vol. 91, no. 20, p. 203901, nov 2003.

[37] K. Keykhosravi, G. Durisi, and E. Agrell, "A tighter upper bound on the capacity of the nondispersive optical fiber channel," in *2017 European Conference on Optical Communication (ECOC)*. IEEE, 2017, pp. 1–3.