The Rate-Distortion-Perception-Classification Tradeoff: Joint Source Coding and Modulation via Inverse-Domain GANs

Junli Fang*, João F. C. Mota*, Baoshan Lu, Weicheng Zhang, Xuemin Hong

***Abstract*—The joint source-channel coding (JSCC) framework leverages deep learning to learn from data the best codes for source and channel coding. When the output signal, rather than being binary, is directly mapped onto the IQ domain (complex-valued), we call the resulting framework joint source coding and modulation (JSCM). We consider a JSCM scenario and show the existence of a strict tradeoff between channel rate, distortion, perception, and classification accuracy, a tradeoff that we name RDPC. We then propose two image compression methods to navigate that tradeoff: the RDPCO algorithm which, under simple assumptions, directly solves the optimization problem characterizing the tradeoff, and an algorithm based on an inverse-domain generative adversarial network (ID-GAN), which is more general and achieves extreme compression. Simulation results corroborate the theoretical findings, showing that both algorithms exhibit the RDPC tradeoff. They also demonstrate that the proposed ID-GAN algorithm effectively balances image distortion, perception, and classification accuracy, and significantly outperforms traditional separation-based methods and recent deep JSCM architectures in terms of one or more of these metrics.**

***Index Terms*—Image compression, joint source-channel coding, joint source coding and modulation, generative adversarial networks, rate-distortion-perception-classification tradeoff.**

I. Introduction

Traditional communication systems follow the celebrated source-channel coding theorem by Shannon [1], which states that source coding and channel coding can be designed separately without loss of optimality. Source coding removes redundant information from a signal, for example, by representing it in a different domain and zeroing out small coefficients. Channel coding, on the other hand, adds to the resulting compressed signal additional information, error-correcting codes, to make its transmission via a noisy channel more robust. Such a modular design, while optimal for memoryless ergodic channels with codes of infinite block length, becomes unsuitable for extreme scenarios, e.g., when bandwidth is highly limited or the channel varies rapidly. An example is underwater acoustic communication, in which multipath interference and noise are so large that the performance of separate source-channel coding schemes sharply drops below a certain signal-to-noise ratio (SNR), a phenomenon known as *the cliff effect* [2]. Traditional techniques like fast adaptive modulation and channel coding rarely work in such an environment, especially for long source bit sequences like images.

Joint source coding and modulation. The above problem can be addressed by jointly designing the source coding, channel coding, and modulation schemes, a framework we call *joint source coding and modulation* (JSCM). JSCM directly maps signals to the IQ domain and generalizes *joint source and channel coding* (JSCC) [3], in which the output signal is binary rather than complex-valued.

In the context of large signals (like images), and under extreme compression requirements (as in underwater communication), the selection of the features to be compressed strikes tradeoffs between different metrics: for example, optimizing for image reconstruction may reduce the perceptual quality of the reconstructed image or decrease the accuracy of a subsequent image classification algorithm. The study of such type of tradeoffs started with the seminal work in [4], which modified the rate-distortion framework [1] to study the tradeoff between perception and distortion metrics of image restoration algorithms. In this paper, we extend the study of such tradeoffs to a JSCM scenario. This requires considering not only vector signals, and thus the possibility to reduce their dimension, but also various metrics, including rate, distortion, perception, and classification performance.

A. The RDPC function and problem statement

We introduce the rate-distortion-perception-classification (RDPC) function in a JSCM scenario. To define it, we consider an n-dimensional source signal $\boldsymbol{X} \in \mathbb{R}^n$ that can be drawn from one of L classes:

$$\boldsymbol{X}|H_l \sim p_{\boldsymbol{X}|H_l}\,, \qquad l = 1, \ldots, L\,, \tag{1}$$

where H_l represents the hypothesis that $\boldsymbol{X}$ is drawn from class l, which occurs with probability $p_l := \mathbb{P}(H_l)$. The communication process is modeled as a Markov chain

$$H_l \xrightarrow{p_{\boldsymbol{X}|H_l}} \boldsymbol{X} \xrightarrow{p_{\boldsymbol{Y}|\boldsymbol{X}}} \boldsymbol{Y} \xrightarrow{p_{\widehat{\boldsymbol{Y}}|\boldsymbol{Y}}} \widehat{\boldsymbol{Y}} \xrightarrow{p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}} \widehat{\boldsymbol{X}}\,, \tag{2}$$

where $\boldsymbol{X}$, $\widehat{\boldsymbol{X}} \in \mathbb{R}^n$ represent the source and reconstructed signals, and $\boldsymbol{Y}$, $\widehat{\boldsymbol{Y}} \in \mathbb{R}^m$, with $m < n$, represent the

Junli Fang, Weicheng Zhang, and Xuemin Hong are with the School of Informatics, Xiamen University, Xiamen, China. (e-mail: {junlifang,zhangweicheng}@stu.xmu.edu.cn, xuemin.hong@xmu.edu.cn)

Baoshan Lu is with the School of Electronics and Information Engineering, Guangxi Normal University, Guilin, China. (e-mail: baoshanlu@gxnu.edu.cn)

João F. C. Mota is with the School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK. (e-mail: j.mota@hw.ac.uk)

Work supported in part by the National Natural Science Foundation of China under Grant 62077040, by Guangxi Natural Science Foundation (Grant 2024GXNSFBA010309), and by UK's EPSRC New Investigator Award (EP/T026111/1).

*equal contribution

Paper accepted in IEEE Transactions on Signal Processing, 2024.

transmitted and received signals. The distribution $p_{\boldsymbol{Y}|\boldsymbol{X}}$ (resp. $p_{\widehat{\boldsymbol{Y}}|\boldsymbol{Y}}$ and $p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}$) characterizes the encoder (resp. channel and decoder). We assume the channel adds zero-mean Gaussian noise to $\boldsymbol{Y}$, i.e., $p_{\widehat{\boldsymbol{Y}}|\boldsymbol{Y}}(\widehat{\boldsymbol{y}}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is an $m \times m$ diagonal matrix whose ith diagonal entry $\boldsymbol{\Sigma}_{ii} > 0$ represents the *noise power* of channel i. Motivated by recent work on task-aware image compression [5]–[7], the purpose of the communication channel in (2) is to transmit images that can be used across different tasks, each of which may have different requirements in terms of image fidelity, perception, or classification. Considering classification as a task justifies the assumption in (1) that $\boldsymbol{X}$ always belongs to a given class.

Our goal is to design the encoder-decoder pair $(p_{\boldsymbol{Y}|\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}})$ and the noise power $\boldsymbol{\Sigma}$ so that the channel rate is minimized while satisfying three constraints:

$$R(D,P,C) = \min_{p_{\boldsymbol{Y}|\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}},\, \boldsymbol{\Sigma}} \quad \sum_{i=1}^{m} \log\left(1 + \tfrac{1}{\boldsymbol{\Sigma}_{ii}}\right) \qquad (3)$$
$$\text{s.t.} \quad \mathbb{E}\big[\Delta(\boldsymbol{X}, \widehat{\boldsymbol{X}})\big] \le D$$
$$d(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}) \le P$$
$$\mathbb{E}\big[\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}})\big] \le C\,.$$

The first constraint bounds below $D \ge 0$ the expected distortion between $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$, as measured by $\Delta : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$.[1] The second constraint enforces a minimal perception quality on $\widehat{\boldsymbol{X}}$ by bounding below $P \ge 0$ the distance between the probability distributions $p_{\boldsymbol{X}}$ of $\boldsymbol{X}$ and $p_{\widehat{\boldsymbol{X}}}$ of $\widehat{\boldsymbol{X}}$, as measured by $d : \mathcal{P}_{\boldsymbol{X}} \times \mathcal{P}_{\boldsymbol{X}} \to \mathbb{R}_+$.[2] And the third constraint bounds below $C \ge 0$ the expected classification error achieved by an arbitrary classifier c_0, as measured by $\epsilon_{c_0} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$. We implicitly assume $\boldsymbol{\Sigma}_{ii} > 0$ and $\boldsymbol{\Sigma}_{ij} = 0$ for $i \ne j$. The objective of (3) defines the channel rate assuming, without loss of generality, that the encoder normalizes its output to have unit power. The reason is to avoid spurious degrees of freedom when defining the rate. In a more practical scenario, one may equivalently have an estimate of the channel noise power and, instead, adjust the power of the output of the encoder. Henceforth, whenever we mention *rate*, we mean *channel rate* (not to be confused with source rate). We will call (3) the RDPC function.

Problem statement. Our goal is to characterize and solve the problem in (3). Specifically, we aim to understand how the different values of D, P, and C affect the achievable rate $R(D,P,C)$. We also aim to design an encoder $p_{\boldsymbol{Y}|\boldsymbol{X}}$, decoder $p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}$, and noise matrix $\boldsymbol{\Sigma}$ that solve (3).

B. Our approach and contributions

As existing characterizations of tradeoffs between, for example, distortion and perception [4], rate, distortion, and perception [8], or classification error, distortion, and perception [9], we show the existence of a tradeoff between rate, distortion, perception, and classification error. Our setup [cf. (2)] is more general than the ones in [4], [8], [9], as we consider vector signals (not necessarily scalar) and their compression in terms of dimensionality. We also establish a strict tradeoff between all the above quantities, i.e., that the function $R(D,P,C)$ is *strictly* convex in D, P, and C.

It is difficult to solve (3) in full generality. So, under the assumption that the source signals are drawn from a Gaussian mixture model (GMM) with two classes ($L = 2$) and that the encoder and decoder are linear maps, we design an algorithm, RDPCO, that directly attempts to solve (3). In addition, leveraging the capacity of generative adversarial networks (GANs) to model probability distributions [10], we also propose to use inverse-domain GAN (ID-GAN) [11] to design an image compression algorithm that achieves both extremely high compression rates and good quality in terms of reconstruction, perception, and classification. Compared to the original GAN [10], ID-GAN [11] learns how to map not only a latent code to an image, but also an image to a latent code. Despite several differences, experimental results show that algorithms RDPCO and ID-GAN exhibit a similar behavior. We summarize our contributions as follows:

- We show the existence of a strict tradeoff between rate, distortion, perception, and classification error in joint source coding and modulation (JSCM).
- We propose two algorithms to solve the JSCM tradeoff problem: a simple algorithm (RDPCO) that directly solves the tradeoff problem but applies only under restrictive assumptions, and another based on inverse-domain GAN (ID-GAN) [11] which can transmit images under extreme compression rates, handling low-capacity channels and preserving semantic information, perception quality, and reconstruction fidelity. In particular, we port techniques from [11], originally applied to image editing, to a JSCM scenario.
- We upper bound the optimal value in (3) when the input signal is a GMM and the encoder and decoder are linear. To achieve this, we derive a new bound on the Wasserstein-1 distance between GMMs in terms of their parameters. See Lemma 3.
- Simulation results show that RDPCO and ID-GAN exhibit the same behavior and reveal further insights about the RDPC problem. In addition, the proposed ID-GAN algorithm achieves a better RDPC tradeoff than a traditional method with source coding and modulation designed separately (JPEG+LDPC+BPSK) and than AE+GAN [12], a recent deep algorithm (modified to a JSCM scenario). It also achieves much better perception and classification accuracy than D-JSCC [2], at the cost of a slight increase in distortion.

C. Organization

We overview related work in Section II and characterize the tradeoff problem (3) in Section III. Section IV analyzes the RDPC tradeoff under GMM source signals and linear encoders/decoders, and proposes an algorithm to achieve the optimal tradeoff. Section V develops the ID-GAN algorithm. The performance of both methods is then assessed in Section VI, and Section VII concludes the paper.

[1] We assume $\Delta(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$.

[2] $\mathcal{P}_{\boldsymbol{X}}$ is the set of probability measures on the measurable space where $\boldsymbol{X}$ is defined, e.g., $\mathbb{R}^n$ and the d-Cartesian product of Borel σ-algebras. We assume $d(p,q) = 0$ if and only if $p = q$.

II. Related Work

We now review prior work on JSCM and then describe existing analyses of tradeoffs in image-based compression.

A. Joint source coding and modulation (JSCM)

JSCM schemes outperform classical source-channel separation methods. Prior work on JSCM methods, also called JSCC even when they output complex-valued signals, can be divided into two categories according to the type of channel: basic channel transmission, in which the channel is simple like a Gaussian or Rayleigh channel, and advanced channel transmission, in which more realistic models for the channels are adopted, and the emphasis is on optimizing the transmission aspect of the system.

JSCM for basic channels. Methods in this category typically focus on designing neural networks that optimize the compression performance of the JSCC/JSCM system, while neglecting aspects of transmission optimization, such as radio resource allocation. For example, [2] proposed a deep joint source-channel coding (D-JSCC) algorithm based on an autoencoder and showed that besides outputting images with quality superior to separation-based schemes, the algorithm exhibits graceful performance degradation in low SNR. Techniques vary according to the domain of the data, e.g., text, image, video, or multimodal data. For example, the JSCM system designed in [13] used a recurrent neural network for transmitting text. Also focusing on text transmission, [14] proposed a semantic communication system (DeepSC) based on a transformer and, to evaluate performance, also a novel metric to measure sentence similarity. DeepSC was extended in [15] for speech transmission. JSCM has also been applied to the transmission of multimodal data. For instance, [16] proposed a cooperative scheme to transmit audio, video, and sensor data from multiple end devices to a central server. And concentrating on text and images, [17] designed a coarse-to-fine multitask semantic model using an attention mechanism. The theory and algorithms we derive in this paper fall under this category, as we consider Gaussian channels.

JSCM for advanced channels. Methods in this category adopt more realistic channel, like the erasure channel [18], feedback channel with channel state information (CSI) [19], [20], and the waveform (OFDM, etc.) or multi-user channels. They focus on optimizing transmission. For example, [21] designed retrieval-oriented image compression schemes, [19] used channel feedback to improve the quality of transmission, and [18] considered an adaptive bandwidth to transmit information progressively under an erasure channel. Furthermore, [20] designed an end-to-end approach for D-JSCC [2] with channel state information (CSI) feedback. The main idea was to apply a non-linear transform network to compress both the data and the CSI. Finally, [22] designed a scheme for orthogonal frequency division multiplexing (OFDM) transmission that directly maps the source images onto complex-valued baseband samples.

B. GAN-based compression

Most algorithms for image transmission are based on autoencoders [23], e.g., [2], [18], [19], [21], [24]. Autoencoders, however, compress signals only up to moderate compression ratios. Although they achieve high-quality reconstruction, this is at the cost of communication efficiency. Extreme compression has been achieved instead by using GANs [10], which are generative models that learn, without supervision, both a low-dimensional representation of the data and its distribution [25]. This gives them the potential to achieve extreme compression without undermining image perception quality. For example, [12] proposed an autoencoder-GAN (AE+GAN) image compression system in which the encoder and decoder are trained simultaneously. The resulting method can achieve extremely low bitrates. One of the algorithms we propose, ID-GAN, requires less training (as encoder and decoder are trained separately), but attains a performance similar to or better than AE+GAN [12].

Also related to our work, [26] proposed two algorithms, inverse-JSCC and generative-JSCC, to reconstruct images passed through a fixed channel with a high compression ratio. The inverse-JSCC algorithm views image reconstruction as an inverse problem and uses a powerful GAN model, StyleGAN-2 [27], as a regularizer together with a distortion loss that aligns with human perception, LPIPS [28]. It is thus an unsupervised method. Generative-JSCC transforms inverse-JSCC into a supervised method by learning the parameters of an encoder/decoder pair while keeping the parameters of StyleGAN-2 fixed. This work differs from ours in several ways. First, we consider not only distortion and perception metrics, but also classification accuracy and channel rate. In particular, the experiments in [26] do not consider any classification task. We also characterize the tradeoff between all these metrics. Second, our metric for perception, the Wasserstein-1 distance between the input and output distributions, differs from the LPIPS metric. Third, we train both the encoder and the decoder adversarially, while [26] uses a pre-trained GAN for the decoder. Finally, while training StyleGAN-2 in [26] (on a database of faces) requires tremendous computational resources, training our ID-GAN can be done with less resources.

C. Tradeoff analyses

The study of tradeoffs in lossy compression can be traced back to rate-distortion theory [1], which characterizes the rate-distortion function

$$R(D) = \min_{p_{\widehat{\boldsymbol{X}}|\boldsymbol{X}}} \; I(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \quad \text{s.t.} \quad \mathbb{E}\big[\Delta(\boldsymbol{X}, \widehat{\boldsymbol{X}})\big] \leq D\,, \tag{4}$$

where $I(\boldsymbol{X}, \widehat{\boldsymbol{X}})$ is the mutual information between $\boldsymbol{X}$ and its reconstruction $\widehat{\boldsymbol{X}}$. The $R(D)$ function has a closed-form expression under some simple source distributions and distortion metrics. Recent work has gone beyond using reconstruction metrics, e.g., the mean squared error (MSE), to assess image quality, considering also perception and semantic metrics.

The PD tradeoff. For example, [4] studied the perception-distortion (PD) tradeoff by replacing the objective in (4) with a

divergence metric $d(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}})$ [cf. (3)]. Assuming that the input signal follows a Rademacher distribution, they proved the existence of a tradeoff between the best achievable divergence and the allowable distortion D.

The RDP tradeoff. Building on [4], [8] studied the rate-distortion-perception (RDP) tradeoff. The problem they analyzed was a variation of (3), without the last constraint (on classification error) and with $I(\boldsymbol{X}, \widehat{\boldsymbol{X}})$ in the objective, instead of the rate. Assuming a Bernoulli input, they showed that in lossy image compression, the higher the perception quality of the output images, the lower the achievable rate. Although insightful, the analysis in [8] is not applicable to our scenario, as it considers only scalar signals, thus ignoring the possibility of compressing them, and also skips the quantization step. The work in [29] further improved on [8] and showed that, for a fixed bit rate, imposing a perfect perception constraint doubles the lowest achievable MSE. It further proposed a training framework to achieve the lowest MSE distortion under a perfect perception constraint at a given bit rate.

The CDP tradeoff. The work in [9] analyzed instead the classification-distortion-perception (CDP) tradeoff, i.e., a modification of problem (3) in which $\mathbb{E}[\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}})]$ is minimized subject to the first two constraints (the rate is ignored). Assuming an input signal that is drawn from a Gaussian mixture model with two classes, they showed the existence of a tradeoff. Our setup is more general, as we do not require the input to be Gaussian nor to be drawn from just two classes.

Our approach. In all the above work, the signals are assumed scalar, which is not suitable to study compression in terms of dimensionality reduction. By contrast, in (3), we consider vector signals and minimize the channel rate subject to constraints on distortion, perception, and classification error. Furthermore, we show the existence of a strict tradeoff, rather than just a simple tradeoff (as in [4], [8], [9]) between rate and all the constraints of (3).

III. The RDPC Tradeoff

We now establish the existence of an inherent tradeoff in solving problem (3). Recall our multiclass signal model in (1) and the channel model in (2). Recall also that we assume a Gaussian channel $p_{\widehat{\boldsymbol{Y}}|\boldsymbol{Y}}(\widehat{\boldsymbol{y}}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}_{ii} > 0$ is the noise power of channel i, and $\boldsymbol{\Sigma}_{ij} = 0$ for $i \neq j$.

We assume a deterministic classifier $c_0 : \mathbb{R}^n \to \{1, \dots, L\}$ which, for $l = 1, \dots, L$, decides $c_0(\widehat{\boldsymbol{X}}) = l$ whenever $\widehat{\boldsymbol{X}}$ belongs to a fixed region $\mathcal{R}_l \subset \mathbb{R}^n$. Assuming ϵ_{c_0} is the 0-1 loss, the expected classification error is then

$$\begin{aligned}\mathbb{E}\Big[\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}})\Big] &= \mathbb{P}\big(\text{class}(\boldsymbol{X}) \neq c_0(\widehat{\boldsymbol{X}})\big) \\ &= \sum_{i<j} \mathbb{P}\Big(c_0(\widehat{\boldsymbol{X}}) = i \,\Big|\, H_j\Big) \cdot p_j \\ &= \sum_{i<j} p_j \cdot \int_{\mathcal{R}_i} \mathrm{d}\, p_{\widehat{\boldsymbol{X}}|H_j}\,, \end{aligned} \tag{5}$$

where $p_j := \mathbb{P}(H_l)$ is the probability of $\boldsymbol{X}$ being drawn from class j. Our main result is as follows.

Theorem 1. *Let $\boldsymbol{X}$ be a multiclass model as in (1). Consider the communication scheme in (2) and the associated RDPC problem in (3). Assume the classifier c_0 is deterministic and that the perception function $d(\cdot, \cdot)$ is convex in its second argument. Then, the function $R(D, P, C)$ is strictly convex, and it is non-increasing in each argument.*

Proof. See Appendix A. □

Theorem 1 is generic and applies to any distortion metric Δ, perception metric d, and classifier c_0. The main assumption is that the perception metric $d(\cdot, \cdot)$ is convex in the second argument, which holds for a variety of divergences, e.g., f-divergence (including total variation, Kullback-Leibler, and Hellinger distance) and Rényi divergence [30], [31]. The same assumption was used in [4], [8], [9]. The theorem says that if we optimize the channel for the smallest possible rate, the encoding-decoding system cannot achieve arbitrarily small distortion, perception error, and classification error. These metrics are in conflict and we need to strike a tradeoff between them. This behavior will be observed in practice when we design algorithms to (approximately) solve the RDPC problem. Note that while prior work [4], [8], [9] shows the existence of a tradeoff by proving that a certain function is convex in each argument, we establish a *strict* tradeoff by proving that $R(D, P, C)$ is *strictly* convex in each argument.

As solving the RDPC problem in (3) in full generality i.e., non-parametrically, is difficult, in the next two sections we propose two algorithms that approximately solve that problem under different assumptions. As we will see in the experiments in Section VI, both algorithms exhibit the tradeoff behavior stipulated by Theorem 1.

IV. RDPC Tradeoff under GMM signals and linear encoder and decoder

To make problem (3) more tractable, in this section we assume that the source signal $\boldsymbol{X}$ in (1) is a Gaussian mixture model (GMM) drawn from two classes and that the encoder and decoder are linear. This will enable us to approximate (3) with a problem whose optimal cost function upper bounds the optimal cost of (3) (Section IV-A). We then develop an algorithm, RDPCO, to solve the resulting problem (Section IV-B). More formally, we make the following assumptions.

Assumption 2. *In (1)-(3), we assume:*

1) *The source $\boldsymbol{X} \in \mathbb{R}^n$ is drawn from a two-class GMM:*

$$\boldsymbol{X}|H_0 \sim \mathcal{N}(\boldsymbol{0}_n, \boldsymbol{I}_n) \tag{6a}$$
$$\boldsymbol{X}|H_1 \sim \mathcal{N}(\boldsymbol{c}_n, \boldsymbol{I}_n)\,, \tag{6b}$$

   where $\boldsymbol{0}_n$ is the all-zeros vector in $\mathbb{R}^n$, $\boldsymbol{I}_n$ the identity matrix, and $\boldsymbol{c}_n \in \mathbb{R}^n$ a fixed vector. That is, we set $L = 2$ in (1) and assume $\boldsymbol{X}|H_l$ is Gaussian, $l = 1, 2$.
2) *The encoder $e : \mathbb{R}^n \to \mathbb{R}^m$ and decoder $d : \mathbb{R}^m \to \mathbb{R}^n$ are linear and deterministic, i.e., they are implemented by full-rank matrices $\boldsymbol{E} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{D} \in \mathbb{R}^{n \times m}$.*
3) *We use the mean-squared error (MSE) as a metric for distortion, i.e., $\Delta(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = \|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_2^2$, and the*

Wasserstein-1 distance[3] $W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}})$ *as a metric for perception, where* $p_{\boldsymbol{X}}$ *and* $p_{\widehat{\boldsymbol{X}}}$ *are the distributions of* $\boldsymbol{X}$ *and* $\widehat{\boldsymbol{X}}$.

4) *The classifier* c_0 *is an optimal Bayes classifier. Specifically, given an observation* $\widehat{\boldsymbol{x}}$ *of* $\widehat{\boldsymbol{X}}$, *it decides* H_1 *if* $\mathbb{P}(H_1|\widehat{\boldsymbol{x}}) \geq \mathbb{P}(H_0|\widehat{\boldsymbol{x}})$, *and* H_0 *otherwise.*

Assumptions 1) and 2) imply that the reconstructed signal $\widehat{\boldsymbol{X}}$ is also a GMM. To see this, first note that the output signal is $\widehat{\boldsymbol{X}} = \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{X}+\boldsymbol{N}\big)$, where $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{\Sigma})$. Since the sum of two Gaussian random variables is also Gaussian, we obtain

$$\widehat{\boldsymbol{X}} \,|\, H_0 \;\sim\; \mathcal{N}\Big(\boldsymbol{0}_n \,,\, \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma}\big)\boldsymbol{D}^\top\Big), \tag{7a}$$

$$\widehat{\boldsymbol{X}} \,|\, H_1 \;\sim\; \mathcal{N}\Big(\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n \,,\, \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma}\big)\boldsymbol{D}^\top\Big). \tag{7b}$$

Note that $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ has more rows than columns ($n > m$), making the covariance matrix $\widehat{\boldsymbol{\Sigma}} := \boldsymbol{D}(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma})\boldsymbol{D}^\top$ in (7) rank-deficient, and thus both distributions in (7) degenerate. Henceforth, $\widehat{\boldsymbol{\Sigma}}^{-1}$ will thus refer to the generalized inverse of $\widehat{\boldsymbol{\Sigma}}$. Specifically, let $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ be an eigenvalue decomposition of $\widehat{\boldsymbol{\Sigma}}$, with $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$ being a diagonal matrix of eigenvalues. Define $\boldsymbol{\Lambda}^{-1}$ as the diagonal matrix with diagonal entries $1/\lambda_i$ if $\lambda_i > 0$, and 0 otherwise. Then, $\widehat{\boldsymbol{\Sigma}}^{-1} := \boldsymbol{Q}\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^\top$. Similarly, the generalized determinant of $|\widehat{\boldsymbol{\Sigma}}|$ is the product of the positive entries of $\boldsymbol{\Lambda}$.

A. Problem formulation

Under Assumption 2, problem (3) becomes

$$\begin{aligned} R(D,P,C) = \min_{\boldsymbol{E},\boldsymbol{D},\boldsymbol{\Sigma}} \;& \textstyle\sum_{i=1}^m \log\big(1+\frac{1}{\boldsymbol{\Sigma}_{ii}}\big) \\ \text{s.t.} \;& \mathbb{E}\big[\big\|\boldsymbol{X}-\widehat{\boldsymbol{X}}\big\|_2^2\big] \leq D \\ & W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}) \leq P \\ & \mathbb{E}\Big[\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}})\Big] \leq C\,, \end{aligned} \tag{8}$$

where we omitted the dependence of $\widehat{\boldsymbol{X}}$ on $\boldsymbol{E}$ and $\boldsymbol{D}$ for simplicity. Despite the simplifications made under Assumption 2, problem (8) is still challenging, and we will solve instead an approximation by relaxing its last two constraints. Before doing so, we analyze each constraint in detail.

Distortion constraint. To derive an expression for the first constraint in (8), we first condition the expected values:

$$\begin{aligned} \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}-\boldsymbol{X}\big\|_2^2\Big] = \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}-\boldsymbol{X}\big\|_2^2 \,\big|\, H_0\Big]\cdot p_0 \\ + \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}-\boldsymbol{X}\big\|_2^2 \,\big|\, H_1\Big]\cdot p_1\,, \end{aligned} \tag{9}$$

where $p_l = \mathbb{P}(H_l)$, $l = 0, 1$. Notice that for $l = 0, 1$,

$$\begin{aligned} \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}-\boldsymbol{X}\big\|_2^2 \,\big|\, H_l\Big] = \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}\big\|_2^2 \,\big|\, H_l\Big] \\ - 2\,\mathbb{E}\Big[\widehat{\boldsymbol{X}}^\top\boldsymbol{X} \,\big|\, H_l\Big] + \mathbb{E}\Big[\big\|\boldsymbol{X}\big\|_2^2 \,\big|\, H_l\Big]. \end{aligned} \tag{10}$$

[3]The Wasserstein-p distance between two probability measures $p_{\boldsymbol{X}}, p_{\boldsymbol{Y}}$ in $\mathbb{R}^n$ is $W_p(p_{\boldsymbol{X}}, p_{\boldsymbol{Y}}) = \big(\inf_{\gamma\in\Pi(p_{\boldsymbol{X}},p_{\boldsymbol{Y}})} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim\gamma}\big[\|\boldsymbol{X}-\boldsymbol{Y}\|_2^p\big]\big)^{1/p}$, where $\Pi\big(p_{\boldsymbol{X}}, p_{\boldsymbol{Y}}\big)$ is the set of all joint distributions with marginals $p_{\boldsymbol{X}}$ and $p_{\boldsymbol{Y}}$, and $1 \leq p \leq +\infty$.

Under hypothesis H_0, the last term is simply a constant:

$$\begin{aligned} \mathbb{E}\Big[\big\|\boldsymbol{X}\big\|_2^2 \,\big|\, H_0\Big] &= \mathbb{E}\Big[\mathrm{tr}\big(\boldsymbol{X}\boldsymbol{X}^\top\big) \,\big|\, H_0\Big] = \mathrm{tr}\Big(\mathbb{E}\big[\boldsymbol{X}\boldsymbol{X}^\top \,\big|\, H_0\big]\Big) \\ &= \mathrm{tr}(\boldsymbol{I}_n) = n\,, \end{aligned}$$

where we used the linearity of the trace $\mathrm{tr}(\cdot)$ in the second equality, and (6a) in the third equality. Similarly, under H_1,

$$\begin{aligned} \mathbb{E}\Big[\big\|\boldsymbol{X}\big\|_2^2 \,\big|\, H_1\Big] &= \mathrm{tr}\Big(\mathbb{E}\big[\boldsymbol{X}\boldsymbol{X}^\top \,\big|\, H_1\big]\Big) = \mathrm{tr}(\boldsymbol{I}_n + \boldsymbol{c}_n\boldsymbol{c}_n^\top) \\ &= n + \big\|\boldsymbol{c}_n\big\|_2^2\,, \end{aligned}$$

due to (6b). Similar reasoning applies to the first term of (10):

$$\begin{aligned} \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}\big\|_2^2 \,\big|\, H_0\Big] &= \mathrm{tr}\big(\widehat{\boldsymbol{\Sigma}}\big) \\ \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}\big\|_2^2 \,\big|\, H_1\Big] &= \mathrm{tr}\big(\widehat{\boldsymbol{\Sigma}}\big) + \boldsymbol{c}_n^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n\,, \end{aligned}$$

where $\widehat{\boldsymbol{\Sigma}} := \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma}\big)\boldsymbol{D}^\top$. Finally, the second term of the right-hand side of (10) can be rewritten for $l = 0, 1$ as

$$\begin{aligned} \mathbb{E}\Big[\widehat{\boldsymbol{X}}^\top\boldsymbol{X} \,\big|\, H_l\Big] &= \mathbb{E}\Big[\boldsymbol{X}^\top\boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{X}+\boldsymbol{N}\big) \,\big|\, H_l\Big] \\ &= \mathbb{E}\big[\mathrm{tr}\big(\boldsymbol{E}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{D}\big) \,\big|\, H_l\big] + \mathbb{E}\Big[\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{N} \,\big|\, H_l\Big] \\ &= \mathrm{tr}\Big(\boldsymbol{E}\,\mathbb{E}\big[\boldsymbol{X}\boldsymbol{X}^\top \,\big|\, H_l\big]\boldsymbol{D}\Big)\,, \end{aligned} \tag{11}$$

where we used $\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{B}\boldsymbol{A})$ (since the dimensions allow) in the first equality and the independence between $\boldsymbol{X}$ and $\boldsymbol{N}$ in the last equality. Plugging (10)-(11) into (9),

$$\begin{aligned} \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}-\boldsymbol{X}\big\|_2^2\Big] = \Big[\mathrm{tr}\big(\widehat{\boldsymbol{\Sigma}}\big) - 2\,\mathrm{tr}\big(\boldsymbol{E}\boldsymbol{D}\big) + n\Big]p_0 + \Big[\mathrm{tr}\big(\widehat{\boldsymbol{\Sigma}}\big) \\ + \boldsymbol{c}_n^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n - 2\mathrm{tr}\Big(\boldsymbol{E}(\boldsymbol{I}_n + \boldsymbol{c}_n\boldsymbol{c}_n^\top)\boldsymbol{D}\Big) + n + \big\|\boldsymbol{c}_n\big\|_2^2\Big]p_1\,. \end{aligned} \tag{12}$$

Perception constraint. We now consider the perception constraint in (8), which upper bounds the Wasserstein-1 distance $W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}})$ by P. Both $p_{\boldsymbol{X}}$ and $p_{\widehat{\boldsymbol{X}}}$ are Gaussian mixture models for which, to the best of our knowledge, there is no closed-form expression for their Wasserstein-p distance. There is, however, a closed-form expression for the Wasserstein-2 distance between Gaussian distributions. Specifically, let $\boldsymbol{X} \sim p_{\boldsymbol{X}} = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{X}}, \boldsymbol{\Sigma}_{\boldsymbol{X}})$ and $\boldsymbol{Y} \sim p_{\boldsymbol{Y}} = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{Y}}, \boldsymbol{\Sigma}_{\boldsymbol{Y}})$ be two Gaussian random vectors with means $\boldsymbol{\mu}_{\boldsymbol{X}}, \boldsymbol{\mu}_{\boldsymbol{Y}} \in \mathbb{R}^n$ and positive semidefinite covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{X}}, \boldsymbol{\Sigma}_{\boldsymbol{Y}} \succeq \boldsymbol{0}_{n\times n}$. It can be shown that the squared Wasserstein-2 distance between them is [32], [33]

$$\big\|\boldsymbol{\mu}_{\boldsymbol{X}} - \boldsymbol{\mu}_{\boldsymbol{Y}}\big\|_2^2 + \mathrm{tr}\Big(\boldsymbol{\Sigma}_{\boldsymbol{X}} + \boldsymbol{\Sigma}_{\boldsymbol{Y}} - 2\Big(\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{\frac{1}{2}}\boldsymbol{\Sigma}_{\boldsymbol{X}}\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{\frac{1}{2}}\Big)^{\frac{1}{2}}\Big)\,.$$

In the case where $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{Y}}$ commute, i.e., $\boldsymbol{\Sigma}_{\boldsymbol{X}}\boldsymbol{\Sigma}_{\boldsymbol{Y}} = \boldsymbol{\Sigma}_{\boldsymbol{Y}}\boldsymbol{\Sigma}_{\boldsymbol{X}}$, the expression simplifies to

$$W_2^2(p_{\boldsymbol{X}}, p_{\boldsymbol{Y}}) = \big\|\boldsymbol{\mu}_{\boldsymbol{X}} - \boldsymbol{\mu}_{\boldsymbol{Y}}\big\|_2^2 + \big\|\boldsymbol{\Sigma}_{\boldsymbol{X}}^{\frac{1}{2}} - \boldsymbol{\Sigma}_{\boldsymbol{Y}}^{\frac{1}{2}}\big\|_F^2\,, \tag{13}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Our objective is thus to upper bound $W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}})$ as a function of $W_2(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}\,|\,H_0)$ and $W_2(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}\,|\,H_1)$, which we define as in footnote 3 [or, in dual form, as in (25) below] with expected values conditioned on H_0 or H_1. We have the following result.

Lemma 3. *Let $p_{\boldsymbol{X}}$ (resp. $p_{\widehat{\boldsymbol{X}}}$) be a GMM modeled as (6) [resp. (7)], in which the probability of hypothesis H_0 is p_0 and of hypothesis H_1 is $p_1 = 1 - p_0$. Then,*

$$W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}) \leq \left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F + \left\|\boldsymbol{DEc_n} - \boldsymbol{c_n}\right\|_2 \cdot p_1 \,. \tag{14}$$

Proof. See Appendix B. □

To enforce the second constraint in (8), we will thus impose the right-hand side of (14) to be bounded by P.

Classification constraint. We now address the last constraint of (8). As in Assumption 2.4), we assume a Bayes classifier, which achieves a minimal probability of error. Such a probability, however, does not have a closed-form expression, but is upper bounded by the Bhattacharyya bound [34]. For a two-class GMM $\boldsymbol{X} \sim p_0\,\mathcal{N}(\boldsymbol{\mu}_0,\, \boldsymbol{\Sigma}_0) + p_1\,\mathcal{N}(\boldsymbol{\mu}_1,\, \boldsymbol{\Sigma}_1)$, the bound is

$$\begin{aligned}\mathbb{P}(\text{error}^\star) &\leq \sqrt{p_0\,p_1}\int_{\mathbb{R}^n}\sqrt{p_{\boldsymbol{X}|H_0}(\boldsymbol{x})\,p_{\boldsymbol{X}|H_1}(\boldsymbol{x})}\;\mathrm{d}\boldsymbol{x}\\ &= \sqrt{p_0\,p_1}\,\exp\left[-\frac{1}{8}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^\top\left[\frac{\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1}{2}\right]^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\right.\\ &\qquad\left.-\frac{1}{2}\log\frac{\left|(\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1)/2\right|}{\sqrt{|\boldsymbol{\Sigma}_0||\boldsymbol{\Sigma}_1|}}\right],\end{aligned} \tag{15}$$

where $|\cdot|$ is the determinant of a matrix, and error* is the classification error achieved by a Bayes classifier. We apply (15) to $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$, whose models are in (6) and (7). Thus, $\boldsymbol{\mu}_0 = \boldsymbol{0}_n$, $\boldsymbol{\mu}_1 = \boldsymbol{DEc}_n$, and $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{D}(\boldsymbol{EE}^\top + \boldsymbol{\Sigma})\boldsymbol{D}^\top$. Hence,

$$\begin{aligned}\mathbb{E}\left[\epsilon_{c_0}(\boldsymbol{X},\,\widehat{\boldsymbol{X}})\right] &= \mathbb{P}\big(\text{class}(\widehat{\boldsymbol{X}}) \neq c_0(\widehat{\boldsymbol{X}})\big)\\ &\leq \sqrt{p_0 p_1}\exp\left[-\frac{1}{8}\boldsymbol{c}_n^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{DEc}_n\right].\end{aligned} \tag{16}$$

So, in (8), rather than bounding $\mathbb{E}\big[\epsilon_{c_0}(\boldsymbol{X},\,\widehat{\boldsymbol{X}})\big] \leq C$, we impose instead that the right-hand side of (16) is upper bounded by C, which is equivalent to

$$\boldsymbol{c}_n^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{DEc}_n \geq -8\log\frac{C}{\sqrt{p_0 p_1}}\,. \tag{17}$$

This defines a nonconvex set over $\boldsymbol{E}$, $\boldsymbol{D}$, and $\boldsymbol{\Sigma}$ (via $\widehat{\boldsymbol{\Sigma}}$).

Bound on RDPC. Instead of solving (8), we will aim to solve a problem that upper bounds its optimal value:

$$\begin{aligned}R(D,P,C) \leq \min_{\boldsymbol{E},\boldsymbol{\Sigma},\boldsymbol{D}}\quad & \textstyle\sum_{i=1}^m \log\left(1+\frac{1}{\boldsymbol{\Sigma}_{ii}}\right)\\ \text{s.t.}\quad & (12) \leq D\\ & (14) \leq P\\ & (17)\,,\end{aligned} \tag{18}$$

where (12) and (14) refer to the right-hand side of the respective equations. While the first constraint is exact, the second and third constraints are more stringent versions of the original constraints in (8). The resulting problem, however, is still nonconvex and will require approximation techniques.

B. RDPCO: Heuristic algorithm for RDPC optimization

Solving (18) is difficult, as it is nonconvex and has an infinite number of solutions. Indeed, $\boldsymbol{E}$ and $\boldsymbol{D}$ appear in the constraints of (18) always as the product $\boldsymbol{DE}$. Thus, if $(\boldsymbol{E}^\star, \boldsymbol{\Sigma}^\star, \boldsymbol{D}^\star)$ is a solution of (18) so is $(\boldsymbol{E}^\star\boldsymbol{M}, \boldsymbol{\Sigma}^\star, \boldsymbol{D}^\star\boldsymbol{M}^{-1})$ for any invertible matrix $\boldsymbol{M}$. This means there are too many degrees of freedom. We will leverage this to first design the output covariance matrix $\widehat{\boldsymbol{\Sigma}}$, and then alternatively find the encoder-decoder pair $(\boldsymbol{E}, \boldsymbol{D})$, via intuitive principles, and the rate matrix $\boldsymbol{\Sigma}$, via a barrier-type method applied to (18).

Design of $\widehat{\boldsymbol{\Sigma}}$. While the original signals in (6) have non-degenerate distributions, the decoded signals in (7) have degenerate distributions. Specifically, assuming that $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ have full rank and that $\text{range}(\boldsymbol{E}) \cap \text{null}(\boldsymbol{D}) = \emptyset$, the output signals in (7) live in an m-dimensional subspace. If the fixed vector $\boldsymbol{c}_n$, which represents the distance between $\boldsymbol{X}|H_0$ and $\boldsymbol{X}|H_1$, is orthogonal to that subspace (equivalently $\boldsymbol{DEc}_n = \boldsymbol{0}_n$), then $\widehat{\boldsymbol{X}}|H_0$ and $\widehat{\boldsymbol{X}}|H_1$ become indistinguishable. In this case, classification is impossible and perception is also undermined [note that the second term in (14) requires $\|\boldsymbol{DEc}_n - \boldsymbol{c}_n\|_2$ to be small].

To avoid this, we first generate the (degenerate) covariance matrix $\widehat{\boldsymbol{\Sigma}} := \boldsymbol{D}(\boldsymbol{EE}^\top + \boldsymbol{\Sigma})\boldsymbol{D}^\top$ by guaranteeing that the distance between $\boldsymbol{X}|H_0$ and $\boldsymbol{X}|H_1$ is preserved after transmitting these signals through the channel. We achieve this by guaranteeing that $\boldsymbol{c}_n$ is an eigenvector of $\widehat{\boldsymbol{\Sigma}}$ associated to eigenvalue 1, while the remaining eigenvectors are associated to eigenvalues of smaller magnitude. Specifically, we set $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, where the first column of $\boldsymbol{Q}$ is $\boldsymbol{c}_n$ and the remaining ones are the output of Gram-Schmidt orthogonalization. Also, $\boldsymbol{\Lambda} = \text{Diag}(1, \lambda_2, \ldots, \lambda_m, 0, \ldots, 0)$, with λ_i being drawn uniformly at random from $[0, 1]$, for $i = 2, \ldots, m$.

Once $\widehat{\boldsymbol{\Sigma}}$ is fixed, we alternate between computing the encoder-decoder pair $(\boldsymbol{E}, \boldsymbol{D})$ and the rate matrix $\boldsymbol{\Sigma}$.

Finding $(\boldsymbol{E}, \boldsymbol{D})$. With $\widehat{\boldsymbol{\Sigma}}$ fixed and assuming that, at iteration k, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma_{k-1}}$ is also fixed, we seek a factorization $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma_{k-1}D}^\top$. We do so via an intuitive process that leads to a unique factorization. Specifically, we design $\boldsymbol{E}$ and $\boldsymbol{D}$ such that $\boldsymbol{DEE}^\top\boldsymbol{D}^\top$ is as close to the identity matrix as possible (to preserve signals passing through the channel), while $\boldsymbol{D\Sigma_{k-1}D}^\top$ is as small as possible (to mitigate the effects of noise). Also, we ensure the principal direction $\boldsymbol{c}_n$ is preserved: $\boldsymbol{DEc}_n \simeq \boldsymbol{c_n}$. These requirements, weighted equally, can be cast as an optimization problem:

$$\begin{aligned}\min_{\boldsymbol{E},\boldsymbol{D}}\quad & \tfrac{1}{2}\left\|\boldsymbol{I}_n - \boldsymbol{DEE}^\top\boldsymbol{D}^\top\right\|_F^2 + \tfrac{1}{2}\left\|\boldsymbol{D\Sigma_{k-1}D}^\top\right\|_F^2\\ & \qquad + \tfrac{1}{2}\left\|\boldsymbol{c_n} - \boldsymbol{DEc_n}\right\|_2^2\\ \text{s.t.}\quad & \widehat{\boldsymbol{\Sigma}} = \boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma_{k-1}D}^\top\,,\end{aligned} \tag{19}$$

which, eliminating the constraint, can be written as

$$\begin{aligned}\min_{\boldsymbol{E},\boldsymbol{D}}\ & \frac{1}{2}\left\|\boldsymbol{I}_n - \widehat{\boldsymbol{\Sigma}} + \boldsymbol{D\Sigma_{k-1}D}^\top\right\|_F^2 + \frac{1}{2}\left\|\boldsymbol{D\Sigma_{k-1}D}^\top\right\|_F^2\\ & + \frac{1}{2}\left\|\boldsymbol{c_n} - \boldsymbol{DEc_n}\right\|_2^2.\end{aligned} \tag{20}$$

We apply gradient descent to (20) in order to find $(\boldsymbol{E_k}, \boldsymbol{D_k})$. It can be shown that the partial derivatives of the objective $g(\boldsymbol{E}, \boldsymbol{D})$ of (20) are

$$\frac{\partial g(\boldsymbol{E}, \boldsymbol{D})}{\partial \boldsymbol{E}} = \boldsymbol{D}^\top (\boldsymbol{D}\boldsymbol{E}\boldsymbol{c_n} - \boldsymbol{c_n})\boldsymbol{c_n}^\top \tag{21a}$$

$$\frac{\partial g(\boldsymbol{E}, \boldsymbol{D})}{\partial \boldsymbol{D}} = 4\boldsymbol{D}\boldsymbol{\Sigma_k}\boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{\Sigma_{k-1}} + 2(\boldsymbol{I_n} - \widehat{\boldsymbol{\Sigma}})\boldsymbol{D}\boldsymbol{\Sigma_{k-1}} + (\boldsymbol{D}\boldsymbol{E}\boldsymbol{c_n} - \boldsymbol{c_n})\boldsymbol{c_n}^\top \boldsymbol{E}^\top . \tag{21b}$$

Finding $\boldsymbol{\Sigma}$. Once the encoder-decoder pair is fixed at $(\boldsymbol{E_k}, \boldsymbol{D_k})$, we find the diagonal rate matrix $\boldsymbol{\Sigma} := \mathrm{Diag}(\boldsymbol{\sigma}) := \mathrm{Diag}(\sigma_1, \dots, \sigma_m)$ by applying a barrier method [35] to (18), i.e., we solve a sequence of problems in t, each of which is

$$\min_{\boldsymbol{\Sigma}=\mathrm{Diag}(\boldsymbol{\sigma})} \; t\, h_r(\boldsymbol{\sigma}) - \lambda_D h_D(\boldsymbol{\Sigma}) - \lambda_P h_P(\boldsymbol{\Sigma}) - \lambda_C h_C(\boldsymbol{\Sigma}) , \tag{22}$$

where $\lambda_D, \lambda_P, \lambda_C \geq 0$ are regularization parameters, and

$$h_r(\boldsymbol{\sigma}) = \sum_{i=1}^{m} \log\left(1 + \frac{1}{\sigma_i}\right) \tag{23a}$$

$$h_d(\boldsymbol{\Sigma}) = \log\left[D_k - \mathrm{tr}\big(\boldsymbol{D_k}\boldsymbol{\Sigma}\boldsymbol{D_k}^\top\big)\right] \tag{23b}$$

$$h_p(\boldsymbol{\Sigma}) = \log\left[P_k - \big\|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{\frac{1}{2}} - \boldsymbol{I_n}\big\|_F^2\right] \tag{23c}$$

$$h_c(\boldsymbol{\Sigma}) = \log\left[\boldsymbol{c_n}^\top \boldsymbol{E_k}^\top \boldsymbol{D_k}^\top \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{-1} \boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{c_n} + 8 \log \frac{C}{\sqrt{p_0 p_1}}\right] . \tag{23d}$$

where $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}} = \boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{E_k}^\top\boldsymbol{D_k}^\top + \boldsymbol{D_k}\boldsymbol{\Sigma}\boldsymbol{D_k}^\top$. In (23b), D_k absorbs all the terms independent from $\boldsymbol{\Sigma}$ when we set $\mathbb{E}[\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_2^2] \leq D$ in (12) [including D]. To obtain (23c), note that imposing the right-hand side of (14) to be smaller than P is equivalent to $\big\|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{\frac{1}{2}} - \boldsymbol{I_n}\big\|_F^2 \leq \big(P - \|\boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{c_n} - \boldsymbol{c_n}\|_2 \cdot p_1\big)^2 =: P_k$. And $h_p(\boldsymbol{\Sigma})$ depends on $\boldsymbol{\Sigma}$ via $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}$. Finally, (23d) is the direct application of the log-barrier function to (17).

To solve each instance of (22), we apply again gradient descent. While the gradients of h_r in (23a) and h_d in (23b) can be computed directly, namely $\mathrm{d}h_r(\boldsymbol{\sigma})/\,\mathrm{d}\sigma_i = -1/(\sigma_i^2 + \sigma_i)$, for $i = 1, \dots, m$, and $\nabla_{\boldsymbol{\sigma}} h_d(\boldsymbol{\Sigma}) = -\mathrm{diag}(\boldsymbol{D_k}^\top \boldsymbol{D_k})/[D_k - \mathrm{tr}(\boldsymbol{D_k}\boldsymbol{\Sigma}, \boldsymbol{D_k}^\top)]$, where $\mathrm{diag}(\cdot)$ extracts the diagonal entries of a matrix into a vector, computing the gradients of h_p in (23c) and h_c in (23d) is more laborious. Their expressions are shown in (24) where, for simplicity, we omitted the iteration index.

RDPCO algorithm. We summarize all the above steps in Algorithm 1, which we name RDPCO for RDPC Optimization. Steps 1-4 describe the procedure to generate the covariance matrix $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{D}(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma})\boldsymbol{D}^\top$, whose factors are then computed in steps 5-16. The barrier method in steps 8-11 stops whenever the duality gap is below 0.01 or the number of iterations reaches $m/100$, both parameters determined experimentally. The remaining parameters that we used in our experiments are reported in Section VI-A.

Algorithm 1 RDPCO algorithm

Input: mean $\boldsymbol{c_n} \in \mathbb{R}^n$; probabilities $p_0, p_1 = 1 - p_0$; bounds on distortion D, perception P, and classification C; initial barrier parameter t_0 and update parameter μ; max. # of iterations K; stopping criteria parameter ϵ; parameters $\lambda_D, \lambda_P, \lambda_C$.
Initialization: $\boldsymbol{\Sigma_0} = \boldsymbol{I}_m$

Generate $\widehat{\boldsymbol{\Sigma}}$
1: Set $\widetilde{\boldsymbol{Q}} = \big[\boldsymbol{c_n} \;\; \boldsymbol{R}\big]$, where $\boldsymbol{R} \in \mathbb{R}^{n \times n-1}$ has i.i.d. $\mathcal{N}(0,1)$ entries
2: Apply Gram-Schmidt orthogonalization to $\widetilde{\boldsymbol{Q}}$ to obtain $\boldsymbol{Q}$
3: Generate $\lambda_i \in [0,1]$, $i = 2, \dots, m$ randomly and build $\boldsymbol{\Lambda} = \mathrm{Diag}(1, \lambda_2, \dots, \lambda_m, 0, \dots, 0) \in \mathbb{R}^{n \times n}$
4: Set $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$

Find $\boldsymbol{E}, \boldsymbol{D}, \boldsymbol{\Sigma}$
5: **for** $k = 1, \dots, K$ **do**
6: Find $(\boldsymbol{E_k}, \boldsymbol{D_k})$ via gradient descent applied to (20) [cf. (21)]
7: Set $t = t_0$
8: **for** $r = 1, \dots, \lceil m/100 \rceil$ **do**
9: Find $\boldsymbol{\Sigma_r}$ via gradient descent applied to (22)
10: $t \leftarrow \mu t$
11: **end for**
12: Set $\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma_r}$
13: **if** $\|(\boldsymbol{E_k}, \boldsymbol{D_k}, \boldsymbol{\Sigma_k}) - (\boldsymbol{E_{k-1}}, \boldsymbol{D_{k-1}}, \boldsymbol{\Sigma_{k-1}})\|_F \leq \epsilon$ **then**
14: Stop
15: **end if**
16: **end for**

V. Solving RDPC with Inverse-Domain GAN

RDPCO attempts to solve the RDPC problem (3) under restrictive assumptions [see Assumption 2]. In this section, leveraging the modeling power of neural networks, in particular generative adversarial networks (GANs) [10], we propose an algorithm that works under more general assumptions. In our channel diagram (2), we will thus model the encoder $p_{\boldsymbol{Y}|\boldsymbol{X}}$ with a neural network $e(\cdot\,; \boldsymbol{\theta}_e) : \mathbb{R}^n \to \mathbb{R}^m$ parameterized by $\boldsymbol{\theta}_e$, and the decoder $p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}$ as neural network $d(\cdot\,; \boldsymbol{\theta}_d) : \mathbb{R}^m \to \mathbb{R}^n$ parameterized by $\boldsymbol{\theta}_d$. These networks will be trained as in ID-GAN [11] which, however, was proposed for a task different from JSCC/JSCM. Specifically, given an (adversarially-trained) image generator, the goal in [11] was to train an encoder to obtain a semantically-meaningful latent code for image editing. We adopt this process of training the generator first, and then the encoder.

A. Proposed scheme

Fig. 1 shows our framework based on ID-GAN. As in [11], we first train an image generator/decoder $d(\cdot\,; \boldsymbol{\theta}_d)$ (Fig. 1, top) adversarially against discriminator f_1, which learns to distinguish a real signal from a randomly generated one, $d(\boldsymbol{Z}\,; \boldsymbol{\theta}_d)$, where $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)$ is a vector of i.i.d. standard Gaussians. This is the conventional GAN setup [10], [36]. As the discriminator is a particular case of a classifier, outputting just a binary signal, it is also known as a *critic*. Once the decoder is trained, we fix it and train the encoder $e(\cdot\,; \boldsymbol{\theta}_e)$ together with its own critic f_2, which again learns to distinguish real signals from randomly generated from ones (Fig. 1, bottom). Comparing (2) and Fig. 1 (bottom), we see that $p_{\boldsymbol{Y}|\boldsymbol{X}}$ is implemented by $e(\cdot\,; \boldsymbol{\theta_e})$, $p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}$ is implemented by $d(\cdot\,; \boldsymbol{\theta_d^\star})$, and the Gaussian channel noise $p_{\widehat{\boldsymbol{Y}}|\boldsymbol{Y}}(\widehat{\boldsymbol{y}}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Sigma})$ has (diagonal) covariance matrix $\boldsymbol{\Sigma} = \sigma_t^2 \boldsymbol{I}_m$, where σ_t is a parameter we learn (or fix) during training. If we normalize the output $\boldsymbol{Y}$ of the encoder to have unit power,

$$\nabla_{\boldsymbol{\sigma}} h_p(\boldsymbol{\Sigma}) = \left[\operatorname{diag}\left(\boldsymbol{D}^\top\left(\boldsymbol{I_m} - 2\left(\boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma D}^\top\right)^{-1}\right)\boldsymbol{D}\right)\right] \Big/ \left(P_k - \left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F^2\right) \tag{24a}$$

$$\nabla_{\boldsymbol{\sigma}} h_c(\boldsymbol{\Sigma}) = \frac{\operatorname{diag}\left(\boldsymbol{D}^\top\left(\boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma D}^\top\right)^{-1}\boldsymbol{DEcc}^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\left(\boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma D}^\top\right)^{-1}\boldsymbol{D}\right)}{\boldsymbol{c_n}^\top\boldsymbol{E_k}^\top\boldsymbol{D_k}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{c_n} + 8\log\frac{C}{\sqrt{p_0 p_1}}} \tag{24b}$$

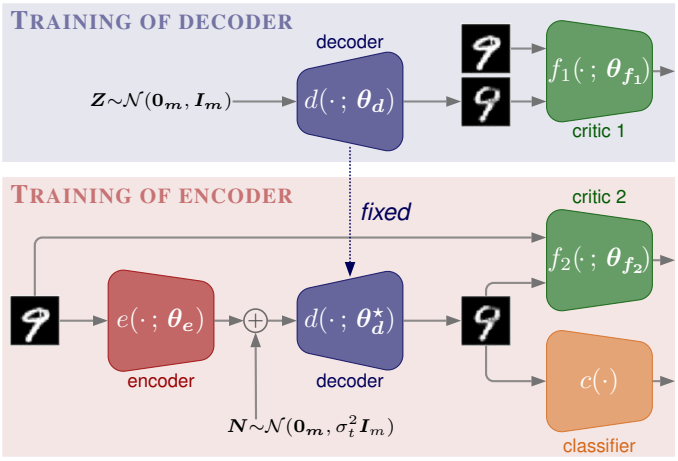


Fig. 1. Proposed ID-GAN framework for solving the RDPC problem (3). The decoder is first trained adversarially with critic 1 in the first step (top). The decoder is then fixed and coupled with an encoder, which is in turn trained with critic 2 in order to preserve both reconstruction quality and classification accuracy (bottom). Critics 1 and 2 have the same architecture.

then the signal-to-noise ratio (SNR) is determined by σ_t as $\text{SNR}_t = -10\log_{10}\sigma_t^2$, and the channel rate is

$$R = m\log_2\left(1 + \frac{1}{\sigma_t^2}\right). \tag{24}$$

Next, we explain the training processes of the decoder and encoder in more detail.

B. Training the decoder

To train the decoder $d(\cdot\,;\boldsymbol{\theta}_d)$ and critic f_1 as in Fig. 1 (top), we use the Wasserstein GAN (WGAN) [36] framework. This consists of finding the parameters $\boldsymbol{\theta}_d$ of the decoder that minimize the Wasserstein-1 (or earth-mover) distance $W_1(p_r, p_{\boldsymbol{\theta}_d})$ between the distribution p_r of real data and the distribution $p_{\boldsymbol{\theta}_d}$ of data generated by $d(\boldsymbol{Z}\,;\boldsymbol{\theta}_d)$, with $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)$. We consider the Kantarovich-Rubinstein dual form of the Wasserstein-1 distance:

$$W_1(p_r, p_{\boldsymbol{\theta}_d}) = \sup_{\|f_1\|_L \le 1} \mathbb{E}_{\boldsymbol{X}\sim p_r}[f_1(\boldsymbol{X})] - \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{\theta}_d}}[f_1(\boldsymbol{X})], \tag{25}$$

where the supremum is over the functions $f_1 : \mathbb{R}^n \to \mathbb{R}$ that are 1-Lipschitz continuous. While the critic f_1 is found by maximizing the argument in (25), the parameters of the decoder are found by minimizing the full Wasserstein distance $W_1(p_r, p_{\boldsymbol{\theta}_d})$. This distance enables overcoming the mode collapse observed in the original GAN framework [10], which used instead the Jensen-Shannon divergence. To enforce Lipschitz-continuity of the critic f_1, [36] proposed to limit its parameters to a small box around the origin. The work in [37], however, found that this technique leads to instabilities in training (exploding/vanishing gradients) and showed that a gradient penalty solves these problems. We thus adopt the loss suggested in [37] for finding critic f_1:

$$L_{f_1} = \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{\theta}_d}}[f_1(\boldsymbol{X})] - \mathbb{E}_{\boldsymbol{X}\sim p_r}[f_1(\boldsymbol{X})] + \lambda_g\,\mathbb{E}_{\widetilde{\boldsymbol{X}}\sim p_{\boldsymbol{\theta}_d, r}}\left[\left(\|\nabla_{\widetilde{\boldsymbol{x}}} f_1(\widetilde{\boldsymbol{X}})\|_2 - 1\right)^2\right], \tag{26}$$

where $\lambda_g \ge 0$, and $\widetilde{\boldsymbol{X}}$ is a point sampled uniformly on the line joining a real data point $\boldsymbol{X} \sim p_r$ and a point $\boldsymbol{Y} \sim p_{\boldsymbol{\theta}_d}$ generated by the decoder. The third term in (26) eliminates the need to constrain the critic to be Lipschitz-continuous [constraint in (25)]; see [37] for more details. In turn, the parameters $\boldsymbol{\theta}_d$ of the decoder are found by minimizing (25) which, when f_1 is fixed, is equivalent to minimizing

$$L_{\boldsymbol{\theta}_d} = -\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{\theta_d}}}[f_1(\boldsymbol{X})] = -\mathbb{E}_{\boldsymbol{Z}\sim\mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)}\left[f_1\left(d(\boldsymbol{Z}\,;\boldsymbol{\theta}_d)\right)\right]. \tag{27}$$

During training, there are two nested loops: the outer loop updates $\boldsymbol{\theta_d}$; and the inner loop, which runs for n_{critic} iterations, updates the parameters of the critic such that the supremum in (25) is reasonably well computed. See [36], [37] for details.

C. Training the encoder

After training the decoder, we fix its parameters to $\boldsymbol{\theta}_d^\star$ and consider the scheme in Fig. 1 (bottom) to train the encoder $e(\cdot\,;\boldsymbol{\theta}_e)$. As the decoder, the encoder is also trained adversarially against a critic f_2 to enhance perception quality, but also takes into account the reconstruction quality and semantic meaning of the reconstruction. The former is captured by an MSE loss between the original and reconstructed images, and the latter by a cross-entropy loss between the image label and the output of a pre-trained classifier $c(\cdot)$ applied to the reconstructed image.

Derivation of the loss for the encoder. To motivate our loss for the encoder, we start from the RDPC problem (3), considering $\Delta(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$ as the distortion metric, $d(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}) = W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}})$ as the perception metric, and the cross-entropy $\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = \text{CE}(c(\widehat{\boldsymbol{X}}), \ell(\boldsymbol{X}))$ as the classification loss, where $\ell(\boldsymbol{X})$ denotes the class of $\boldsymbol{X}$. Then, there exist constants λ_d, λ_p, and λ_c (related to D, P, and C), such that (3) and

$$\min_{p_{\boldsymbol{Y}|\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}, \boldsymbol{\Sigma}} \sum_{i=1}^{m} \log\left(1 + \frac{1}{\boldsymbol{\Sigma}_{ii}}\right) + \mathbb{E}\left[\lambda_d\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_2^2\right.$$

Algorithm 2 ID-GAN compression: training of the encoder

Input: Training images/labels $\{(\boldsymbol{x}^{(t)}, \boldsymbol{\ell}^{(t)})\}_{t=1}^{T}$, pretrained decoder $d(\cdot)$, pretrained classifier $c(\cdot)$, learning rate α, momentum parameters β_1, β_2, batch size S, number of iterations of critic n_{critic}, and loss hyperparameters λ_g, λ_d, λ_p, λ_c

Initialization: Set encoder $\boldsymbol{\theta}_e^{(1)}$ and critic $\boldsymbol{\theta}_f^{(1)}$ parameters randomly; set channel noise standard deviation $\sigma_t^{(1)} > 0$ randomly

In each epoch:

1: $\mathcal{S} = \text{randperm}(\{1, 2, \ldots, T\})$
2: **for** $j = 1, \ldots, \lceil T/S \rceil$ **do**
3: Select next S batch indices $\mathcal{S}_j$ from $\mathcal{S}$
4: **for** $k = 1$ to n_{critic} **do**
5: Generate (channel) noise $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)$
6: $L_{f_2}^{(1)} = \frac{1}{S} \sum_{s \in \mathcal{S}_j} f_2\Big(d\big(e(\boldsymbol{x}^{(s)}\,;\, \boldsymbol{\theta}_e^{(j)}) + \sigma_t^{(j)} \boldsymbol{Z}\big)\,;\, \boldsymbol{\theta}_f^{(k)}\Big)$
7: $L_{f_2}^{(2)} = \frac{1}{S} \sum_{s \in \mathcal{S}_j} f_2\Big(\boldsymbol{x}^{(s)}\,;\, \boldsymbol{\theta}_f^{(k)}\Big)$
8: **for** $s \in \mathcal{S}_j$ **do**
9: Draw $\epsilon \sim \mathcal{U}(0,1)$ randomly
10: $\widetilde{\boldsymbol{x}}^{(s)} = (1-\epsilon) \cdot \boldsymbol{x}^{(s)} + \epsilon \cdot d\big(e\big(\boldsymbol{x}^{(s)}\,;\, \boldsymbol{\theta}_e^{(j)}\big) + \sigma_t^{(j)} \boldsymbol{Z}\big)$
11: **end for**
12: $L_{f_2}^{(3)} = \frac{1}{S} \sum_{s \in \mathcal{S}_j} \Big(\big\|\nabla_{\widetilde{\boldsymbol{x}}} f_2(\widetilde{\boldsymbol{x}}^{(s)})\big\|_2 - 1\Big)^2$
13: $L_{f_2} = L_{f_2}^{(1)} - L_{f_2}^{(2)} + \lambda_g L_{f_2}^{(3)}$
14: $\boldsymbol{\theta}_f^{(k+1)} = \text{Adam}\Big(\boldsymbol{\theta}_f^{(k)},\, \lambda_p L_{f_2},\, \alpha,\, \beta_1,\, \beta_2\Big)$
15: **end for**
16: $\boldsymbol{\theta}_f^{(j)} = \boldsymbol{\theta}_f^{(n_{\text{critic}})}$
17: **for** $s \in \mathcal{S}_j$ **do**
18: $\widehat{\boldsymbol{x}}^{(s)} = d\Big(e\big(\boldsymbol{x}^{(s)}\,;\, \boldsymbol{\theta}_e^{(j)}\big) + \sigma_t^{(j)} \boldsymbol{Z}\Big)$, w/ $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)$
19: **end for**
20:
$$L_e = m \log_2\Big(1 + 1/{\sigma_t^{(j)}}^2\Big) + \frac{1}{S} \sum_{s \in \mathcal{S}_j} \lambda_d \Big\|\boldsymbol{x}^{(s)} - \widehat{\boldsymbol{x}}^{(s)}\Big\|_2^2 + \lambda_c \text{CE}\big(c(\widehat{\boldsymbol{x}}^{(s)}), \boldsymbol{\ell}^{(s)}\big) - \lambda_p f_2\Big(\widehat{\boldsymbol{x}}^{(s)}\,;\, \boldsymbol{\theta}_f^{(j)}\Big)$$
21: $\Big(\boldsymbol{\theta}_e^{(j+1)}, \sigma_t^{(j+1)}\Big) = \text{Adam}\Big(\big(\boldsymbol{\theta}_e^{(j)}, \sigma_t^{(j)}\big),\, L_e,\, \alpha,\, \beta_1,\, \beta_2\Big)$
22: **end for**

$$+ \lambda_c \text{CE}\big(c(\widehat{\boldsymbol{X}}),\, \ell(\boldsymbol{X})\big)\Big] + \lambda_p W_1\big(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}}\big) \quad (28)$$

have the same solution. Problem (28) is non-parametric, i.e., the functions representing the encoder $p_{\boldsymbol{Y}|\boldsymbol{X}}$ and the decoder $p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}$ have no structure. As mentioned, we assume they are implemented by neural networks $e(\cdot\,;\boldsymbol{\theta_e})$ and $d(\cdot\,;\boldsymbol{\theta}_d)$, respectively. The encoder, in particular, normalizes its output signal to unit power. According to the channel model (2), the output of the decoder is then $\widehat{\boldsymbol{X}} = d(e(\boldsymbol{X}) + \boldsymbol{N})$, where we omitted dependencies on $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_d$ for simplicity. Under these assumptions and further assuming that the different channels have equal variance [i.e., (24)], (28) becomes

$$\underset{\boldsymbol{\theta}_e, \sigma_t}{\text{minimize}}\; m \log_2\Big(1 + \frac{1}{\sigma_t^2}\Big) + \mathbb{E}\Big[\lambda_d \big\|\boldsymbol{X} - d(e(\boldsymbol{X}) + \sigma_t \boldsymbol{Z})\big\|_2^2 + \lambda_c \text{CE}\Big(c\big(d(e(\boldsymbol{X}) + \sigma_t \boldsymbol{Z})\big),\, \ell(\boldsymbol{X})\Big)\Big] + \lambda_p W_1\big(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}}\big)\,, \quad (29)$$

where, akin to the reparameterization trick [38], we replaced $\boldsymbol{N}$ by $\sigma_t \boldsymbol{Z}$, with $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)$. This makes the dependency of $\boldsymbol{N}$ on σ_t explicit and enables computing derivatives with respect to σ_t. Note that the expectation in (29) is with respect to $\boldsymbol{X}$ and $\boldsymbol{Z}$. Note also that $\boldsymbol{\theta}_d$ is not included in the optimization variables of (29), as the decoder has already been trained. Adopting the approximations for W_1 described in Section V-B, we find the parameters of the encoder and channel noise level by solving

$$\underset{\boldsymbol{\theta}_e, \sigma_t}{\text{minimize}}\; m \log_2\Big(1 + \frac{1}{\sigma_t^2}\Big) + \mathbb{E}\Big[\lambda_d \big\|\boldsymbol{X} - d(e(\boldsymbol{X}) + \sigma_t \boldsymbol{Z})\big\|_2^2 + \lambda_c \text{CE}\Big(c\big(d(e(\boldsymbol{X}) + \sigma_t \boldsymbol{Z})\big),\, \ell(\boldsymbol{X})\Big) - \lambda_p f_2(d(e(\boldsymbol{X}) + \sigma_t \boldsymbol{Z}))\Big]\,. \quad (30)$$

As remarked in Section I-A, we design the encoder and channel noise level σ_t jointly. In practice, one would instead estimate the channel noise level and design the power of the encoder. The advantage of doing as we do is that the first term of (30) is convex in σ_t. The parameters $\boldsymbol{\theta}_{f_2}$ of critic f_2, in turn, are computed like in (26), by minimizing the loss

$$L_{f_2} = \lambda_p \Big\{\mathbb{E}\Big[f_2\big(d(e(\boldsymbol{X}) + \sigma_t \boldsymbol{Z})\big) - f_2(\boldsymbol{X})\Big] + \lambda_g\, \mathbb{E}\Big[\big(\big\|\nabla_{\widetilde{\boldsymbol{x}}} f_2\big(\widetilde{\boldsymbol{X}}\big)\big\|_2 - 1\big)^2\Big]\Big\}\,, \quad (31)$$

where $\widetilde{\boldsymbol{X}} = (1-\epsilon)\boldsymbol{X} + \epsilon\, d(e(\boldsymbol{X}) + \sigma_t \boldsymbol{Z})$ and $\epsilon \sim \mathcal{U}(0,1)$ is uniformly distributed in $[0,1]$. The first expectation is with respect to $\boldsymbol{X}$ and $\boldsymbol{Z}$, and the second with respect to $\boldsymbol{X}$ and ϵ.

Training algorithm. The complete training procedure of the encoder is shown in Algorithm 2. Its inputs include training images $\boldsymbol{x}^{(t)}$ and corresponding labels $\boldsymbol{\ell}^{(t)}$, and a pretrained decoder $d(\cdot)$ and classifier $c(\cdot)$. After initializing the parameters of the encoder, associated critic, and channel noise level, in each epoch we randomly permute the indices of the training data (step 1) and visit all the training data in batches of size S. This takes $\lceil T/S \rceil$ iterations, where T is the number of data points. The loop in steps 4-15 performs n_{critic} iterations of Adam to minimize the critic loss in (31) and thus to update the critic f_2 parameters $\boldsymbol{\theta}_f$ (where we omit the index 2 for simplicity). This corresponds to computing the supremum in the Wasserstein distance (25) between the real data $\boldsymbol{X}$ and the reconstructed one $\widehat{\boldsymbol{X}} = d\big(e\big(\boldsymbol{X}\big) + \boldsymbol{N}\big)$. The terms in (26) are computed separately, with the last term requiring the creation of the intermediate variables $\widetilde{\boldsymbol{x}}^{(s)}$ in steps 8-11. As usual, expected values were replaced by sample averages over the batch. After having updated the parameters of critic f_2, we perform one iteration of Adam to minimize the encoder loss in (30) and thus to update the encoder parameters $\boldsymbol{\theta}_e$ and channel noise level σ_t. This requires passing each image in the batch through the encoder and decoder to create $\widehat{\boldsymbol{x}}^{(t)}$, as in steps 17-19. In step 21, the parameters of the encoder and the channel noise level are updated simultaneously. In our experiments, reported in Section VI, we run two versions of Algorithm 2: one exactly as described, where the noise level σ_t, and thus $\text{SNR}_t = -10 \log_{10} \sigma_t^2$, is learned during training; and another where SNR_t is fixed to a predefined value.

VI. Experimental Results

We now present our experiments to evaluate the performance of the proposed algorithms, RDPCO (Algorithm 1)

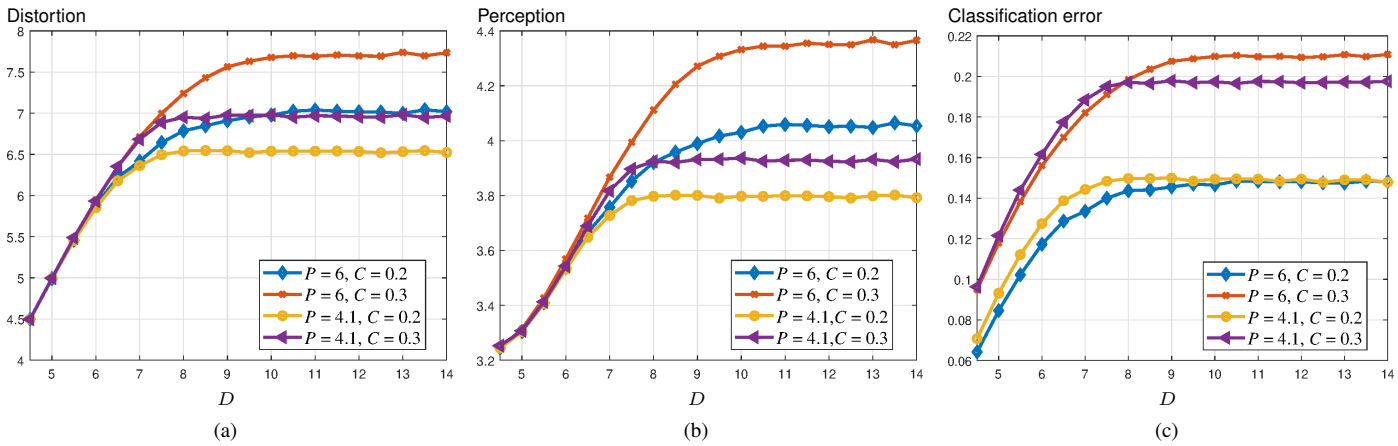


Fig. 2. Values of (a) distortion, (b) perception, and (c) classification error for RDPCO for varying distortion parameter D. These metrics are computed by the right-hand side of the expressions in (12), (14), and (16), respectively.

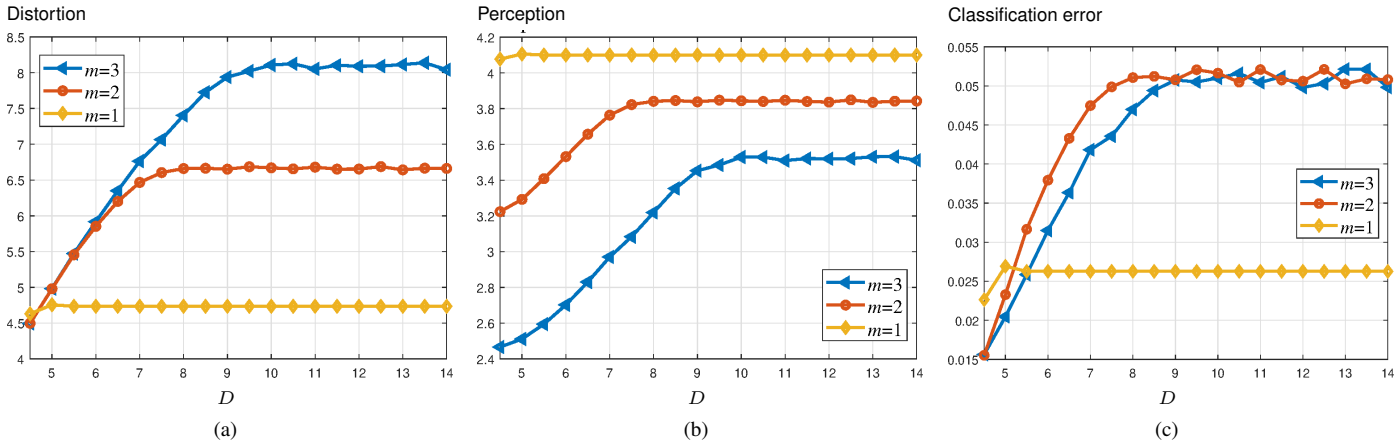


Fig. 3. Values of (a) distortion, (b) perception, and (c) classification error for RDPCO for varying latent dimension m and hyperparameters $(P, C) = (4.1, 0.1)$.

and ID-GAN (Algorithm 2). We start with RDPCO and then consider ID-GAN.

A. RDPCO algorithm

Recall that RDPCO (Algorithm 1) solves the approximation (18) of (3). Before explaining the experiments, we describe how we set the parameters of the algorithm.

Experimental setup. In Algorithm 1, we generate $\boldsymbol{c_n} \in \mathbb{R}^n$ randomly with i.i.d. Gaussian entries with zero mean and variance 4. The classes are always equiprobable, i.e., $p_0 = p_1 = 1/2$. In the gradient descent method in step 6, we employ a constant learning rate of 10^{-4} for 20k iterations. In the barrier method in steps 8-11, we initialize t as $t_0 = 0.01$ and update it with a factor of $\mu = 2$. The parameter ϵ in step 13 is set to 10^{-5}. To balance the terms in (22), we set $\lambda_D = 1/\log(D)$, $\lambda_P = 1/\log(P)$, and $\lambda_C = -1/\log(\sqrt{p_0\, p_1})$. During the experiments, we vary D, P, C, and R [which depends on the latent dimension m and noise level σ_t; cf. (24)]. To evaluate the performance of the algorithm, we visualize how two metrics vary, e.g., rate and distortion, while the remaining parameters are fixed.

Metrics as a function of D. Here, we fix the input dimension to $n = 5$ and the latent one to $m = 2$. Fig. 2 shows how distortion, perception, and classification error metrics vary with D in (18). These metrics are, respectively, the right-hand side of (12), of (14), and of (16). In Figs. 2(a)- 2(b), we see that when C (resp. P) is fixed, increasing P (resp. C) increases either the distortion or perception metrics. This behavior is as expected according to Theorem 1. In Fig. 2(c), we observe that for a fixed C, modifying P produces no significant effect on the classification error, indicating that the classification constraint becomes active before the perception one.

Fig. 3 is similar to Fig. 2, but P and C are fixed to 4.1 and 0.1, respectively, while the latent dimension m varies. When $m = 1$, all metrics are invariant to D. This is because, as seen in Fig. 3(b), the perception constraint is active (it equals its limit of 4.1), dominating the two other constraints. For $m = 2, 3$, their classification error behaves similarly, but $m = 3$ achieves better perception and worse distortion.

Rate-distortion analysis. In this experiment, as in Fig. 2, we vary both P and C in the constraints of (18). Fig. 4(a) shows the resulting rate-distortion curves. For a fixed P, decreasing C increases the rate; similarly, for a fixed C, decreasing P increases the rate as well. This validates the tradeoff established in Theorem 1. Fig. 4(b) shows the rate-distortion curves under the same parameters as Fig. 3, i.e., $(P, C) = (4.1, 0.1)$. We can see again that, for $m = 1$, the rate is invariant to D, since the perception constraint is the

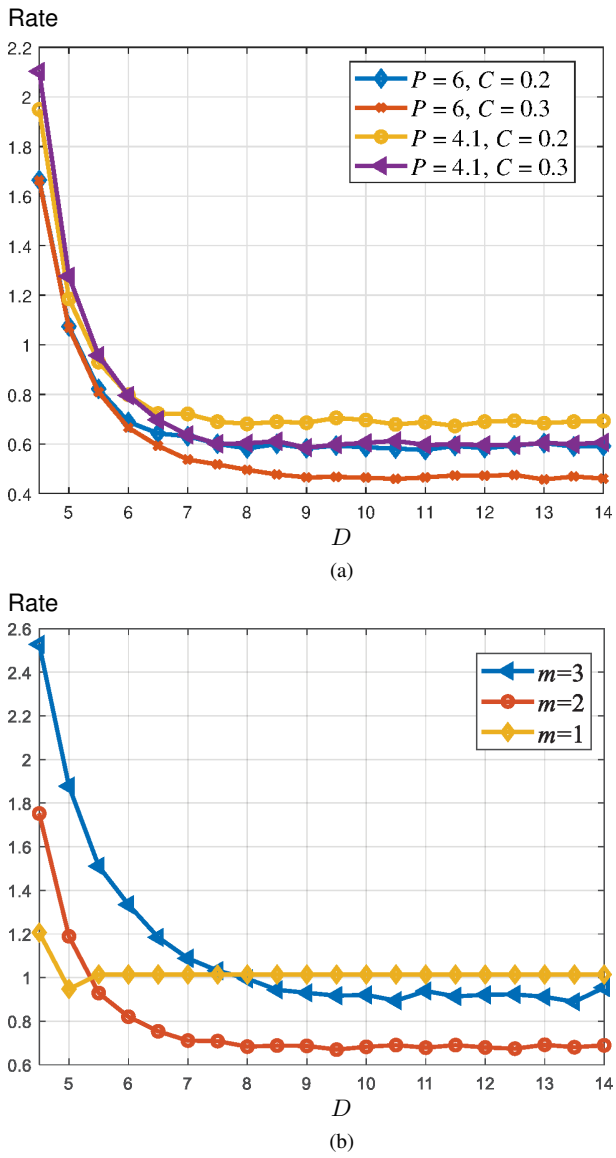


Fig. 4. Rate-distortion curves of RDPCO for (a) varying P and C, and (b) varying compressed dimension m with hyperparameters $(P, C) = (4.1, 0.1)$.

only active one. For $m = 2, 3$, the curves have the familiar tradeoff shape. In this case, $m = 2$ yields a rate-distortion curve better than $m = 3$. The reason is that, as we saw in Fig. 3(b), the perception constraint is the most stringent of three constraints, and $m = 2$ achieves a perception value closer to the limit of 4.1, leading to a better rate-distortion tradeoff in Fig 4(b). These results corroborate the RDPC tradeoff we derived. Indeed, they point to the existence of an optimal latent dimension m that minimizes the rate while satisfying the distortion, perception, and classification constraints.

B. ID-GAN algorithm

We now consider ID-GAN (Algorithm 2) applied to the popular MNIST dataset [39], which has 60k training images and 10k test images of size 28×28 depicting digits from 0 to 9 (10 classes). Before describing the experimental setup in detail, we explain how all the functions in the ID-GAN framework in Fig. 1 were implemented.

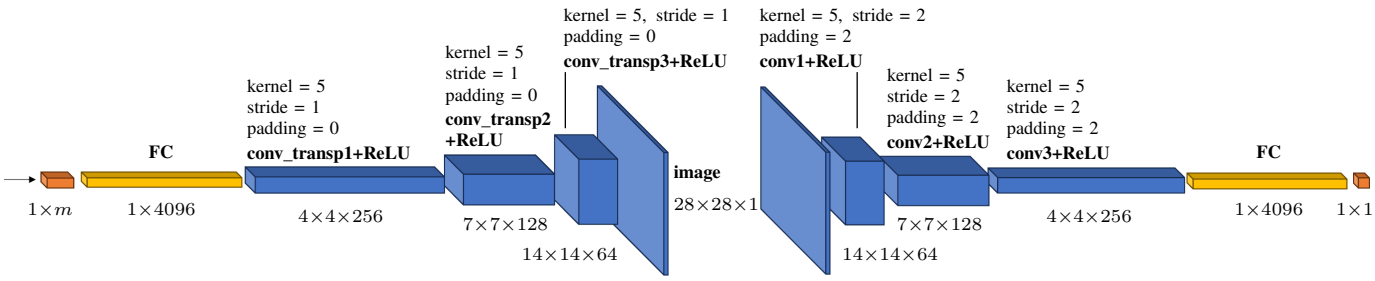
Network architectures. Fig. 5 shows the architectures of the decoder $d(\cdot\,; \boldsymbol{\theta}_d)$ [Fig. 5(a)] and of the critics f_1 and f_2 [Fig. 5(b)]. The latter have the same architecture, but they are initialized independently, with different seeds. The decoder in Fig. 5(a) increases the dimensions of the data by first using a fully connected network, whose output is reshaped to a $4 \times 4 \times 256$ tensor, and then by upsampling along the channel. The upsampling is performed via transposed convolutions with ReLU activations. The architecture of the critics [Fig. 5(b)] is symmetric to that of the decoder. The input image is compressed via a convolutional network, whose output in the last layer is mapped to a probability vector with a sigmoid function. The architecture of the encoder, on the other hand, downsamples the input image using only fully connected layers, as shown in Fig. 6. The network architecture of the classifier [cf. Fig. 1, bottom] is similar to the one of the encoder, except that the last layer is mapped to a 10-dimensional vector (coinciding with the number of classes of MNIST) and a sigmoid is applied to each entry. We train the classifier beforehand and fix it when training the encoder.

Experimental setup. In Algorithm 2, we used a learning rate $\alpha = 10^{-5}$ and acceleration parameters $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for Adam, a batch size of $S = 50$, and $n_{\text{critic}} = 5$ iterations for the inner loop of the critic. The loss hyperparameters and training noise level SNR_t will be reported for each experiment. Also, we ran the algorithm for just 5 epochs. The reason for such a small number is that, as described in Section V-C, the decoder has already been trained when we run Algorithm 2. In fact, the decoder was also trained with few epochs, 8.

Algorithms. We compare the proposed ID-GAN algorithm against D-JSCC [2], the parallel autoencoder-GAN (AE+GAN) [12], and a traditional approach in which source coding, channel coding, and modulation are designed separately: JPEG and Huffman codes for source coding, $3/4$ LDPC codes for channel coding, and BPSK for modulation [40].

Metrics and comparison. For comparison metrics, we selected the mean squared error (MSE), the Fréchet inception distance (FID) [41], and the classification error $(1/|\mathcal{V}|)\sum_{v\in\mathcal{V}} \mathbb{1}_{c_0(\widehat{\boldsymbol{x}}^{(v)})\neq \boldsymbol{\ell}^{(v)}}$, where $\mathcal{V}$ is the test set, and $\mathbb{1}$ the 0-1 loss. FID captures perception quality by measuring the similarity between distributions of real and generated images. The smaller the FID, the closer the distributions. In summary, the smaller all the metrics, the better the performance. Comparing the performance of a JSCM system against a traditional system, however, is challenging. For example, [2] proposed to use the ratio of bandwidth compression. Yet, in a traditional system, it is not obvious how to accurately determine the ratio between the size of images and the corresponding vectors in IQ-domain. Instead, we will use rate as defined in (24), i.e., channel rate, which quantifies the amount of information that a symbol can convey through a channel. In particular, the dimension m of the latent variable $\boldsymbol{y}$ corresponds to the number of constellations in a traditional system. Thus, to compare the different algorithms in a fair way, we fix during testing the channel rate (24), making the results independent from the latent dimension m.

Results. Fig. 7 shows how the above metrics vary with the rate (24), measured in bits/image, for D-JSCC [2],



(a) Decoder (b) Critics

Fig. 5. Architectures of the decoder $d(\cdot\,;\boldsymbol{\theta}_d)$ and of the critics f_1 and f_2 in ID-GAN [cf. Fig. 1]. **FC** stands for fully connected layer, **conv** for convolutional layer, and **conv_transp** for transposed convolutional layer. We indicate the dimensions of the layer as well as the size of the kernels, stride, and padding.

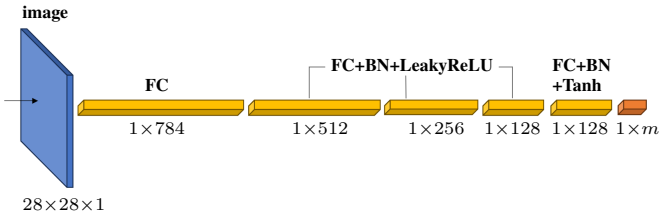


Fig. 6. Architecture of the encoder $e(\cdot\,;\boldsymbol{\theta}_e)$, consisting of fully connected layers of indicated dimensions. Layers 2-4 contain a batch normalization (BN) layer and use LeakyReLU as activation. Layer 5 uses $\tanh$ instead.

AE+GAN [12], and the proposed ID-GAN. The curves represent the average values over the 10k test images of MNIST. For ID-GAN, we fixed the hyperparameters as $(\lambda_g, \lambda_d, \lambda_p, \lambda_c) = (1, 10^3, 1, 10^3)$ and considered four different *fixed* training noise levels, $\text{SNR}_t \in \{-10, 5, 20, \infty\}$ dB [$\text{SNR}_t = \infty$ corresponds to no noise], and also a value of SNR_t that is *learned* during training. The vertical lines in Figs. 7(a)-7(c) indicate the rates corresponding to $\text{SNR}_t = -10$ dB, 5 dB and 20 dB.

Fig. 7(a) indicates that image distortion, measured by the MSE, decreases with the rate for all the algorithms. D-JSCC outperforms all the other methods, as it was designed to minimize image distortion (MSE). Indeed, according to the RDPC tradeoff (Theorem 1), if the perception and classification metrics are ignored, an algorithm can achieve a smaller distortion for a given rate. When ID-GAN is trained with fixed SNR_t, we observe that the larger the value of SNR_t, the lower the achieved MSE for all rates. As we need to take into account other metrics (FID, classification error), this indicates the difficulty of manually selecting SNR_t. The version of ID-GAN with learned SNR_t found an optimal value of $\text{SNR}_t = 16.5$ dB during training, and its MSE performance follows closely the performance of ID-GAN versions with $\text{SNR}_t = 20$ dB and $\text{SNR}_t = \infty$ (the lines of latter practically coincide).

In Fig. 7(b), however, D-JSCC is the algorithm with the worst FID performance, with an optimal point around a rate of 5 bits/image. For AE+GAN, the FID metric decreases monotonically, eventually surpassing ID-GAN trained with $\text{SNR}_t = 20$ dB and ∞. Indeed, these versions of ID-GAN were the best overall in terms of FID performance, again closely followed by ID-GAN with learned SNR_t. When ID-GAN is trained with $\text{SNR}_t = -10$ dB or 5 dB, FID reaches a minimum around a rate of 10 bits/image and then it increases. When SNR_t is learned during training, its FID value decreases until a rate of 20 bits/image, and then it increases gradually. This implies that when SNR_t is learned, the perception quality of ID-GAN is stable for a certain range of the rate. The reason why FID values increase for larger rates in ID-GAN is likely due to the mismatch between the noise levels during testing and training: indeed, the smaller the training noise level, the earlier the FID curve starts to increase. This happens only for FID likely because the decoder, which is largely responsible for the perception quality of the output, is fixed during the training of the encoder. This decoupling may make perception quality more sensitive to discrepancies between training/testing setups.

Fig. 7(c) shows how the classification error of the algorithms varies with the rate. All the versions of ID-GAN [except for $\text{SNR}_t = -10$ dB, for rate > 40 bits/image] outperform both D-JSCC and AE+GAN, which have almost overlapping lines. Within ID-GAN versions with fixed SNR_t, we see that $\text{SNR}_t = 5$ dB yields the best classification error across all the rates. Decreasing it to -10 dB or increasing it to 20 dB or ∞ results in a worse classification error. The downside of such a good performance in terms of classification error is the poor performance in terms of distortion [Fig. 7(a)] and perception [Fig. 7(b)] for large rates. ID-GAN with learned SNR_t, on the other hand, achieves a good performance across all metrics [Figs. 7(a)-7(c)], effectively balancing rate and all the metrics. We conclude that ID-GAN achieves a tradeoff better than AE+GAN, outperforming it in all metrics for rates $\lesssim 40$ bits/image. It also achieves perception and classification performance better than D-JSCC, at the cost of worse distortion.

Visual illustration. Fig. 8 shows concrete image examples reconstructed by all the algorithms. In this experiment, the latent dimension m in D-JSCC [2], AE+GAN [12], and ID-GAN was fixed to 8 and the respective rate was then computed via (24). The parameters of ID-GAN were set to $(\lambda_g, \lambda_d, \lambda_p, \lambda_c) = (1, 500, 1, 10^3)$, and SNR_t was learned. The first column depicts the images reconstructed by a traditional JPEG+LDPC+BPSK system. Even when the rate is high, the reconstructed images fail to match the input one, likely due to quantization and compression artifacts in JPEG. As the rate decreases, we observe a *cliff effect*, with the image quality degrading abruptly. D-JSCC [2] in the second column, on the other hand, is based on an autoencoder which, given

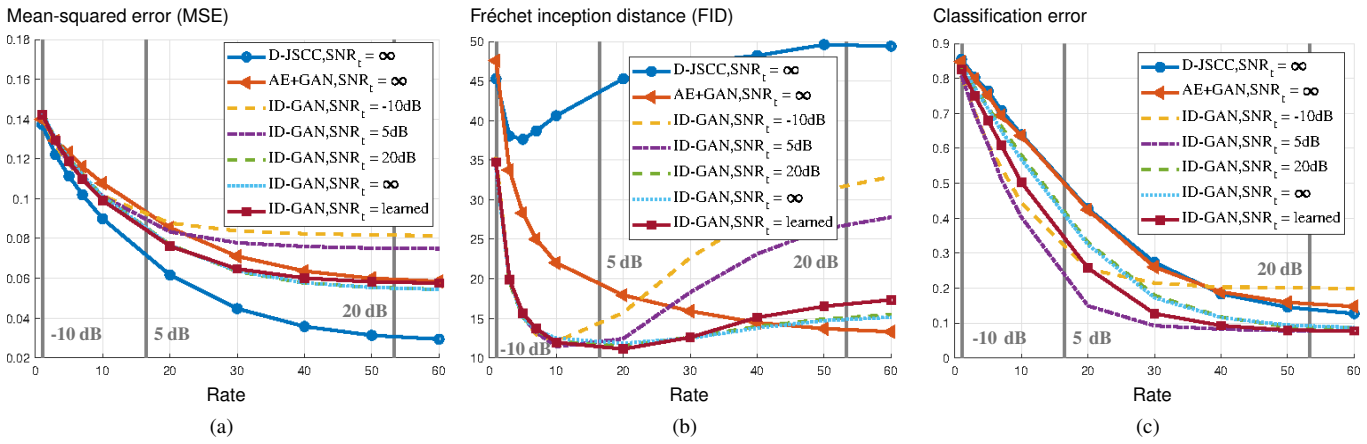


Fig. 7. Different metrics versus rate (24) for the proposed ID-GAN, D-JSCC [2], and AE+GAN [12]. In ID-GAN, we set $(\lambda_g, \lambda_d, \lambda_p, \lambda_c) = (1, 10^3, 1, 10^3)$ and, during training, we either learned $\text{SNR}_t := -10 \log_{10} \sigma_t^2$ or fixed it to -10 dB, 5 dB, 20 dB (vertical lines depicting the corresponding rates), or ∞ (no noise). For all the metrics, the lower the better. (a) mean-squared error (MSE), (b) Fréchet inception distance (FID), and (c) classification error.

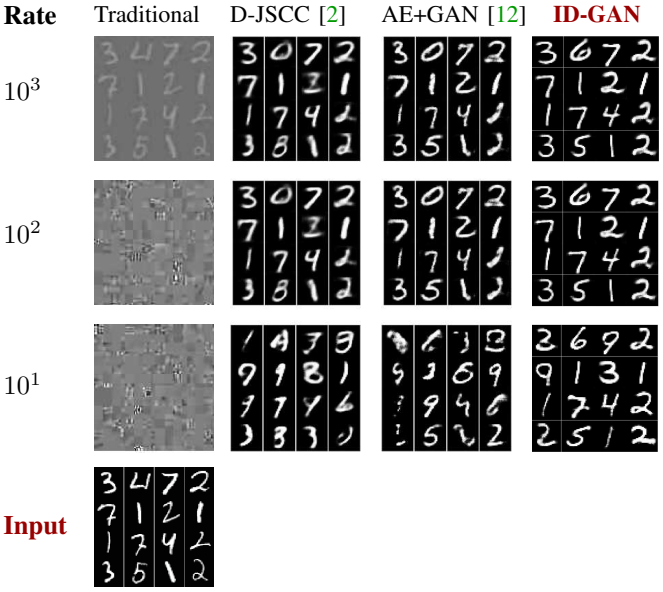


Fig. 8. Example images reconstructed by a traditional method (JPEG+LDPC+BPSK), D-JSCC, AE+GAN training, and proposed ID-GAN with learned SNR_t and parameters $(\lambda_g, \lambda_d, \lambda_p, \lambda_c) = (1, 500, 1, 10^3)$.

the high bandwidth ratio, outputs blurry images at all rates. As in Fig. 7, even though D-JSCC outperforms the other algorithms in MSE, it has a relatively poor perception quality (FID) and classification error. In the third column, the images reconstructed by AE+GAN [12] have poor perceptual quality for a low rate, i.e., 10. We noticed that the quality of its output images collapses when its training parameters λ_d, λ_p, and λ_c are not carefully set, revealing the difficulty in balancing different loss terms. The proposed ID-GAN scheme shown in the fourth column, in contrast, not only preserves semantic information in images, but also reconstructs them with high perceptual quality. Compared to D-JSCC and AE+GAN, ID-GAN generates images with various styles, including different angles and thickness, even at extremely high bandwidth compression ratios.

Rate-distortion analysis. To illustrate how ID-GAN and RDPCO behave similarly, Fig. 9 adapts to MNIST the setup of Fig. 4, which shows rate-distortion curves of RDPCO with varying (P, C) or m. We fixed the training noise level of ID-GAN to $\text{SNR}_t = \infty$ so that we can visualize the effect of modifying its hyperparameters. Fig. 9(a) shows, for different rates, a behavior similar to RDPCO in Fig. 4(a) when we modify its hyperparameters $(\lambda_d, \lambda_p, \lambda_c)$ accordingly. Note that these hyperparameters apply only during training; during testing, we inject different levels of channel noise, obtaining different rates. The first column of Fig. 9(a) weighs all the metrics equally. The second column imposes a more stringent requirement on classification performance by setting λ_c two orders of magnitude larger than λ_d and λ_p [akin to decreasing C in (18)]. While classification accuracy is maintained, we observe loss of details, like digit orientation and line thickness. The third column imposes a larger weight on perception. At low rates, even though semantic information is altered, our algorithm still generates meaningful digits.

Fig. 9(b), akin to Fig. 4(b), shows how different values of m affect the output images of ID-GAN for different rates. Images in the first column, for $m = 2$, are blurry and discontinuous, indicating that they may disregard the perception and classification losses. However, when $m = 8$, the output images seem to be more suitable for transmission at all rates, preserving perception, but with a few semantic mistakes, e.g., a digit 2 becoming a 3. When $m = 64$, the algorithm requires higher rates to reconstruct the images accurately.

Ablation study. We performed a small ablation study to assess whether all the terms in the loss associated to the encoder (30) are necessary for good performance. To do so, we fixed the noise level at $\text{SNR}_t = 30$ dB, corresponding to a rate of 50, which is large enough to enable seeing visual differences in the reconstructed images. First, we trained the encoder with just the MSE loss, i.e., $(\lambda_d, \lambda_p, \lambda_c) = (1, 0, 0)$ in (30). A set of reconstructed digits from the test set is shown in the second column of Fig. 10. The digits are blurry, as the encoder learns just to compress pixel-level information and disregards any semantic information. Then, we trained the encoder with the MSE and classification loss, i.e., $(\lambda_d, \lambda_p, \lambda_c) = (1, 0, 40)$. As shown in the third column of Fig. 10, the algorithm preserved semantic information better, e.g., correctly depicting a 4 in

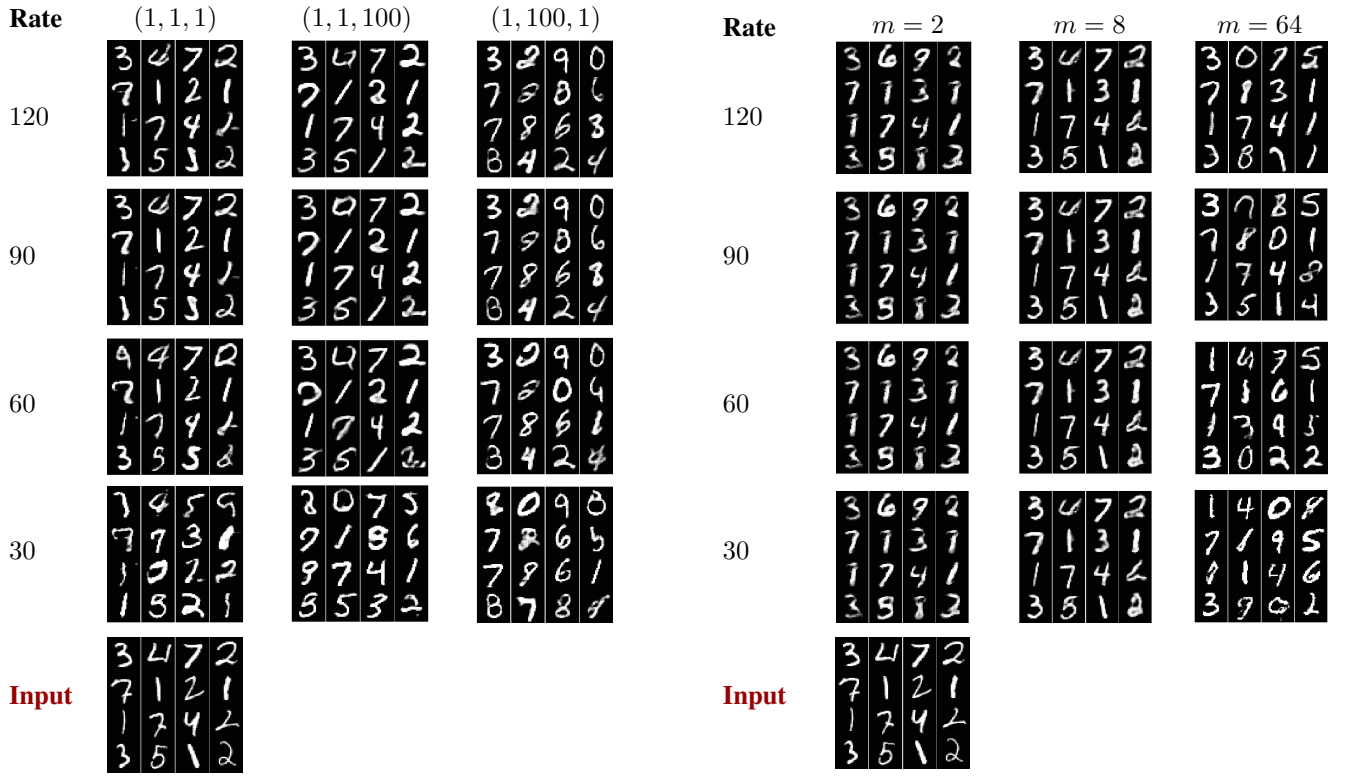


Fig. 9. (a) Corresponding behavior of ID-GAN on MNIST images. The triples on top of each column represent hyperparameters $(\lambda_d, \lambda_p, \lambda_c)$. We fixed $\text{SNR}_t = \infty$. (b) corresponding behavior of ID-GAN on MNIST images with fixed $\text{SNR}_t = \infty$.

Fig. 10. Visualization of the results of the ablation study. In the second column, only the MSE term of (30) was considered during training. In the third, both the MSE and the cross-entropy terms were considered. And in the fourth, the full loss (30) was considered.

the 3rd row and column, but the digits exhibit poor diversity. For example, all the 2's look similar. Finally, we trained the encoder using the full loss, with $(\lambda_d, \lambda_p, \lambda_c) = (1, 1, 40)$. The reconstructed digits in the last column of Fig. 10 are not only sharp, of the correct class (except for the 4 in row 1, column 2) but also exhibit more diversity, thus looking more realistic.

VII. Conclusions

We formulated and analyzed the tradeoff between rate, distortion, perception, and rate (RDPC) in a joint source coding and modulation (JSCM) framework. We showed the existence of a tradeoff and proposed two algorithms to achieve it. One algorithm is heuristic and was designed under simplifying assumptions to minimize an upper bound on the RDPC function; the other was based on inverse-domain GAN (ID-GAN) and works under a general scenario. Experimental results showed that ID-GAN achieves a better tradeoff than a traditional method, in which source coding and modulation are designed separately, and a tradeoff better or similar to recent deep joint source-channel coding schemes. Experiments revealed that improving perception quality and classification accuracy require higher rates, and also showed the existence of an optimal compressed/latent dimension that minimizes rate while satisfying constraints on distortion, perception, and classification.

Acknowledgments

We thank the two reviewers for their insightful suggestions, which significantly improved the quality of the paper.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley & Sons, 1991.

[2] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cog. Comms. Network.*, vol. 39, no. 1, pp. 89–100, 2019.

[3] M. Fresia, F. Peréz-Cruz, H. V. Poor, and S. Verdú, "Joint source and channel coding," *IEEE SP Mag*, vol. 27, no. 6, pp. 104–113, 2010.

[4] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *CVPR*, 2018, pp. 6228–6237.

[5] W. Hua, D. Chen, J. Fang, J. F. C. Mota, and X. Hong, "Semantics-guided contrastive joint source-channel coding for image transmission," in *IEEE Int. Conf. Wireless Communications and Signal Processing (WCSP)*, 2022, pp. 505–510.

[6] Z. Lei, P. Duan, X. Hong, J. F. C. Mota, J. Shi, and C.-X. Wang, "Progressive deep image compression with hybrid contexts of image classification and reconstruction," *IEEE J. Selected Areas in Communications*, vol. 41, no. 1, pp. 72–89, 2023.

[7] W. Hua, L. Xiong, S. Liu, *et al.*, "Classification-driven discrete neural representation learning for semantic communications," *IEEE Internet of Things*, vol. 1, no. 1, pp. 1–14, 2024.

[8] Y. Blau and T. Michaeli, "Rethinking lossy compression: The rate-distortion-perception tradeoff," in *ICML*, 2019, pp. 675–685.

[9] D. Liu, H. Zhang, and Z. Xiong, "On the classification-distortion-perception tradeoff," in *NeurIPS*, 2019, pp. 1–10.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *NeurIPS*, 2014, pp. 1–9.

[11] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *ECCV*, 2020, pp. 592–608.

[12] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *ICCV*, 2019, pp. 221–231.

[13] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *ICASSP*, 2018, pp. 2326–2330.

[14] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE T-SP*, vol. 69, pp. 2663–2675, 2021.

[15] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.

[16] S. Wan, Q. Yang, Z. Shi, Z. Yang, and Z. Zhang, "Cooperative task-oriented communication for multi-modal data with transmission control," in *IEEE Int. Conf. Comms. Workshops*, 2023, pp. 1635–1640.

[17] Q. Y. Z. Zhang, S. He, and et al., "Semantic communication approach for multi-task image transmission," in *IEEE VTC*, 2022, pp. 1–2.

[18] D. B. Kurka and D. Gündüz, "Bandwidth-agile image transmission with deep joint source-channel coding," *IEEE Trans. Wireless Comm.*, vol. 20, no. 12, pp. 8081–8095, 2021.

[19] D. B. Kurka and D. Gündüz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Selected Areas in Inf. Th.*, vol. 1, no. 1, pp. 178–193, 2020.

[20] J. Xu, B. Ai, N. Wang, and W. Chen, "Deep joint source-channel coding for CSI feedback: An end-to-end approach," *IEEE J. Selected Areas in Communications*, vol. 41, no. 1, pp. 260–273, 2023.

[21] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE J. Selected Areas in Communications*, vol. 39, no. 1, pp. 89–100, 2020.

[22] M. Yang, C. Bian, and H.-S. Kim, "Deep joint source channel coding for wireless image transmission with OFDM," in *IEEE Int. Conf. Comms.*, 2021, pp. 1–6.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[24] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, "Wireless image transmission using deep source channel coding with attention modules," *IEEE Trans. Circuits Sys. for Video Tech.*, vol. 32, no. 4, pp. 2315–2328, 2022.

[25] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[26] E. Erdemir, T.-Y. Tung, P. L. Dragotti, and D. Gündüz, "Generative joint source-channel coding for semantic image transmission," *IEEE J. Selected Areas in Communications*, vol. 41, no. 8, pp. 2645–2657, 2023.

[27] T. Karras, S. Laine, M. Aittala, J. Hellsten, and J. Lehtinen, "Analyzing and improving the image quality of StyleGAN," in *CVPR*, 2020, pp. 8110–8119.

[28] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[29] Z. Yan, F. Wen, R. Ying, C. Ma, and P. Liu, "On perceptual lossy compression: The cost of perceptual reconstruction and an optimal training framework," in *ICML*, 2021, pp. 11 682–11 692.

[30] I. Csiszár and P. C. P. C. Shields, "Information theory and statistics: A tutorial," *Found. and Trends in Communications and Information Theory*, vol. 1, no. 4, pp. 417–528, 2004.

[31] T. van Erven and P. Harremoës, "Rényi divergence and Kullback-Leibler divergence," *IEEE T-IT*, vol. 60, no. 7, pp. 3797–3820, 2014.

[32] I. Olkin and F. Pukelsheim, "The distance between two random vectors with given dispersion matrices," *Linear algebra and its applications*, vol. 48, pp. 257–263, 1982.

[33] D. C. Dowson and B. V. Landau, "The Fréchet distance between multivariate normal distributions," *J. Multivariate Analysis*, vol. 12, pp. 450–455, 1982.

[34] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd edition. Wiley, 2001.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017, pp. 214–223.

[37] I. Gulrajani, F. Ahmed, M. Arjovsky, and et al., "Improved training of Wasserstein GANs," in *NeurIPS*, 2017, pp. 1–11.

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014, pp. 1–14.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[40] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Th.*, vol. 8, no. 1, pp. 21–28, 1962.

[41] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *NeurIPS*, 2017, pp. 1–12.

[42] C. Villani, *Optimal Transport: Old and New*. Springer, 2009.

APPENDIX A
PROOF OF THEOREM 1

For convenience, we restate Theorem 1:

Theorem. *Let $\boldsymbol{X}$ be a multiclass model as in (1). Consider the communication scheme in (2) and the associated RDPC problem in (3). Assume the classifier c_0 is deterministic and that the perception function $d(\cdot,\cdot)$ is convex in its second argument. Then, the function $R(D,P,C)$ is strictly convex, and it is non-increasing in each argument.*

Proof. If we increase either D, P, or C in right-hand side of (3), the constraint set of the optimization problem is enlarged or remains the same. This means that $R(D,P,C)$ is non-increasing with any of these variables.

To show strict convexity, we take arbitrary pairs $(D_1,P_1,C_1) \geq 0$ and $(D_2,P_2,C_2) \geq 0$ and, for any $0<\alpha<1$, show that

$$\begin{aligned}&(1-\alpha)R(D_1,P_1,C_1)+\alpha R(D_2,P_2,C_2)\\ &> R\big((1-\alpha)D_1+\alpha D_2,\,(1-\alpha)P_1+\alpha P_2,\,(1-\alpha)C_1+\alpha C_2\big).\end{aligned} \tag{32}$$

To do so, we define, for $j=1,2$,

$$\begin{aligned}\Big(p^{(j)}_{\boldsymbol{Y}|\boldsymbol{X}},\,p^{(j)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}},\,\boldsymbol{\Sigma}^{(j)}\Big) := \underset{p_{\boldsymbol{Y}|\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}},\,\boldsymbol{\Sigma}}{\operatorname{argmin}} \quad & \textstyle\sum_{i=1}^m \log\big(1+\frac{1}{\boldsymbol{\Sigma}_{ii}}\big)\\ \text{s.t.} \quad & \mathbb{E}\big[\Delta(\boldsymbol{X},\widehat{\boldsymbol{X}})\big]\leq D_j\\ & d(p_{\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}})\leq P_j\\ & \mathbb{E}\big[\epsilon_{c_0}(\boldsymbol{X},\widehat{\boldsymbol{X}})\big]\leq C_j,\end{aligned} \tag{33}$$

and denote by $\widehat{\boldsymbol{X}}^{(j)}$ the output of (2) with the parameters computed in (33). Using the strict convexity of the function $x\mapsto\log(1+1/x)$ for $x>0$, the left-hand side of (32) equals

$$\begin{aligned}&(1-\alpha)R(D_1,P_1,C_1)+\alpha R(D_2,P_2,C_2)\\ &=\sum_i\left[(1-\alpha)\log\left(1+\frac{1}{\boldsymbol{\Sigma}^{(1)}_{ii}}\right)+\alpha\log\left(1+\frac{1}{\boldsymbol{\Sigma}^{(2)}_{ii}}\right)\right]\\ &>\sum_i\log\left(1+\frac{1}{(1-\alpha)\boldsymbol{\Sigma}^{(1)}_{ii}+\alpha\boldsymbol{\Sigma}^{(2)}_{ii}}\right) &(34)\\ &\geq \min_{p_{\boldsymbol{Y}|\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}},\,\boldsymbol{\Sigma}} \quad \textstyle\sum_{i=1}^m\log\big(1+\frac{1}{\boldsymbol{\Sigma}_{ii}}\big)\\ &\qquad\text{s.t.} \quad \mathbb{E}\big[\Delta(\boldsymbol{X},\widehat{\boldsymbol{X}})\big]\leq(1-\alpha)D_1+\alpha D_2\\ &\qquad\qquad d(p_{\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}})\leq(1-\alpha)P_1+\alpha P_2\\ &\qquad\qquad \mathbb{E}\big[\epsilon_{c_0}(\boldsymbol{X},\widehat{\boldsymbol{X}})\big]\leq(1-\alpha)C_1+\alpha C_2 &(35)\\ &=R\big((1-\alpha)D_1+\alpha D_2,\,(1-\alpha)P_1+\alpha P_2,\\ &\qquad\qquad (1-\alpha)C_1+\alpha C_2\big). &(36)\end{aligned}$$

Step (35) to (36) follows from the definition of $R(D,P,C)$ in (3). The rest of the proof will consist of showing that the step from (34) to (35) holds. Indeed, this will follow if we show that the triple

$$\begin{aligned}\Big((1-\alpha)p^{(1)}_{\boldsymbol{Y}|\boldsymbol{X}}+\alpha p^{(2)}_{\boldsymbol{Y}|\boldsymbol{X}}\,,\;(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}\,,\\ (1-\alpha)\boldsymbol{\Sigma}^{(1)}+\alpha\boldsymbol{\Sigma}^{(2)}\Big)\end{aligned} \tag{37}$$

satisfies the constraints of the optimization problem in (35).

First, notice that (37) defines valid parameters for the communication process in (2). Specifically, because convex combinations of probability distributions are also probability distributions, $(1-\alpha)p^{(1)}_{\boldsymbol{Y}|\boldsymbol{X}}+\alpha p^{(2)}_{\boldsymbol{Y}|\boldsymbol{X}}$ and $(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}$ characterize valid encoding and decoding processes. If $\boldsymbol{\Sigma}^{(1)}$ and $\boldsymbol{\Sigma}^{(2)}$ are diagonal positive definite matrices, then their convex combination also is. Let then $\widehat{\boldsymbol{X}}^{(\alpha)}$ denote the output of (2) with the parameters in (37). Notice that

$$p_{\widehat{\boldsymbol{X}}^{(\alpha)}|\widehat{\boldsymbol{Y}}}=(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}\,. \tag{38}$$

We will show that $\widehat{\boldsymbol{X}}^{(\alpha)}$ and its probability distribution

$$p_{\widehat{\boldsymbol{X}}^{(\alpha)}}=(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}}\,, \tag{39}$$

where $p^{(1)}_{\widehat{\boldsymbol{X}}}$ and $p^{(2)}_{\widehat{\boldsymbol{X}}}$ are the distributions of the output of (2) with the parameters in (33), satisfy the constraints in (35). Indeed, for the first constraint, conditioning on $\widehat{\boldsymbol{Y}}$,

$$\begin{aligned}&\mathbb{E}\big[\Delta(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(\alpha)})\big]\\ &=\mathbb{E}_{\widehat{\boldsymbol{Y}}}\Big[\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(\alpha)}\big)\,|\,\widehat{\boldsymbol{Y}}\big]\Big] &(40)\\ &=\mathbb{E}_{\widehat{\boldsymbol{Y}}}\Big[(1-\alpha)\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(1)}\big)\,|\,\widehat{\boldsymbol{Y}}\big]\\ &\qquad\qquad+\alpha\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(2)}\big)\,|\,\widehat{\boldsymbol{Y}}\big]\Big] &(41)\\ &=(1-\alpha)\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(1)}\big)\big]+\alpha\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(2)}\big)\big] &(42)\\ &\leq(1-\alpha)D_1+\alpha D_2\,. &(43)\end{aligned}$$

In (40) and (42), we applied the tower property of expectation. From (40) to (41), we used (38). And from (42) to (43), we used (33). For the second constraint, we use the assumption that $d(\cdot,\cdot)$ is convex in its second argument and, again, (38) and (33):

$$\begin{aligned}d\big(p_{\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}^{(\alpha)}}\big)&=d\big(p_{\boldsymbol{X}},\,(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}}\big)\\ &\leq(1-\alpha)d\big(p_{\boldsymbol{X}},\,p^{(1)}_{\widehat{\boldsymbol{X}}}\big)+\alpha d\big(p_{\boldsymbol{X}},\,p^{(2)}_{\widehat{\boldsymbol{X}}}\big)\,.\\ &\leq(1-\alpha)P_1+\alpha P_2\,.\end{aligned}$$

Finally, for the last constraint, we plug $\widehat{\boldsymbol{X}}^{(\alpha)}$ into (5):

$$\begin{aligned}\mathbb{E}\Big[\epsilon_{c_0}(\boldsymbol{X},\,\widehat{\boldsymbol{X}}^{(\alpha)})\Big]&=\sum_{i<j}p_j\cdot\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(\alpha)}|H_j} &(44)\\ &=\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(\alpha)}|\widehat{\boldsymbol{Y}},\,H_j}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}} &(45)\\ &=\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(\alpha)}|\widehat{\boldsymbol{Y}}}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}} &(46)\\ &=(1-\alpha)\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(1)}|\widehat{\boldsymbol{Y}}}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}}\\ &\qquad+\alpha\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(2)}|\widehat{\boldsymbol{Y}}}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}} &(47)\\ &=(1-\alpha)\mathbb{E}\big[\epsilon_{c_0}\big(\boldsymbol{X},\,\widehat{\boldsymbol{X}}^{(1)}\big)\big]+\alpha\mathbb{E}\big[\epsilon_{c_0}\big(\boldsymbol{X},\,\widehat{\boldsymbol{X}}^{(2)}\big)\big] &(48)\\ &\leq(1-\alpha)C_1+\alpha C_2\,. &(49)\end{aligned}$$

From (44) to (45), we conditioned on $\widehat{\boldsymbol{Y}}$. From (45) to (46), we used the Markov property of (2). From (46) to (47), we used (38). From (47) to (48), we applied the same steps as in (5), but in reverse order. And, finally, from (48) to (49), we used (33). □

APPENDIX B
PROOF OF LEMMA 3

For convenience, we restate Lemma 3:

Lemma. *Let $p_{\boldsymbol{X}}$ (resp. $p_{\widehat{\boldsymbol{X}}}$) be a Gaussian mixture model following (6) [resp. (7)], in which the probability of hypothesis H_0 is p_0 and of hypothesis H_1 is $p_1 = 1 - p_0$. Then,*

$$W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}) \leq \left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F + \left\|\boldsymbol{D}\boldsymbol{E}\boldsymbol{c_n} - \boldsymbol{c_n}\right\|_2 \cdot p_1 \,.$$

Proof. We use the dual form of Wasserstein-1 distance in (25):

$$\begin{aligned}
W_1\big(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}\big) &= \sup_{\|f\|_L \leq 1} \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}\big[f(\boldsymbol{X})\big] - \mathbb{E}_{\boldsymbol{X}\sim p_{\widehat{\boldsymbol{X}}}}\big[f(\boldsymbol{X})\big] \\
&\leq \sup_{\|f\|_L \leq 1} \Big(\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}\Big[f(\boldsymbol{X}) \,|\, H_0\Big] - \mathbb{E}_{\boldsymbol{X}\sim p_{\widehat{\boldsymbol{X}}}}\Big[f(\boldsymbol{X}) \,|\, H_0\Big]\Big) p_0 \\
&\quad + \sup_{\|\widehat{f}\|_L \leq 1} \Big(\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}\Big[\widehat{f}(\boldsymbol{X}) \,|\, H_1\Big] - \mathbb{E}_{\boldsymbol{X}\sim p_{\widehat{\boldsymbol{X}}}}\Big[\widehat{f}(\boldsymbol{X}) \,|\, H_1\Big]\Big) p_1 \qquad (50) \\
&=: W_1\Big(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}} \,|\, H_0\Big) \cdot p_0 + W_1\Big(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}} \,|\, H_1\Big) \cdot p_1 \qquad (51) \\
&\leq W_2\Big(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}} \,|\, H_0\Big) \cdot p_0 + W_2\Big(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}} \,|\, H_1\Big) \cdot p_1 \qquad (52) \\
&= \left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_2 p_0 + p_1 \sqrt{\left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F^2 + \left\|\boldsymbol{c}_n - \boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n\right\|_2^2} \qquad (53) \\
&\leq \left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F + \left\|\boldsymbol{c}_n - \boldsymbol{D}\boldsymbol{E}\boldsymbol{c_n}\right\|_2 \cdot p_1 \,.
\end{aligned}$$

In (50), we first conditioned on H_0 and H_1, and then used the subadditivity of the supremum. The inequality is due to using different variables f and $\widehat{f}$. From (50) to (51), we defined the Wasserstein-1 conditional on an event. From (51) to (52), we used the fact that $W_p(\cdot,\cdot) \leq W_q(\cdot,\cdot)$ whenever $p \leq q$; see [42, Remark 6.6]. From (52) to (53), we applied (13) to the models in (6) and (7). And in the last step, we used the triangular inequality. □