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 coding and modulation (JSCM) framework was enabled by recent developments in deep learning, which allows to automatically learn from data, and in an end-to-end fashion, the best compression codes and modulation schemes. In this paper, we show the existence of a strict tradeoff between channel rate, distortion, perception, and classification accuracy in a JSCM scenario. We then propose two image compression methods to navigate that tradeoff: an inverse-domain generative adversarial network (ID-GAN), which achieves extreme compression, and a simpler, heuristic method that reveals insights about the performance of ID-GAN. Experiment results not only corroborate the theoretical findings, but also demonstrate that the proposed ID-GAN algorithm significantly improves system performance compared to traditional separation-based methods and recent deep JSCM architectures.**

***Index Terms*—Image compression, 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 coding rarely work in such an environment, especially for long source bit sequences like images.

Junli Fang, Weicheng Zhang, Wenhui Hua, and Xuemin Hong are with the School of Informatics, Xiamen University, Xiamen, China. (e-mail: {junlifang,zhangweicheng,huawenhui,}@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)

*equal contribution

Joint source coding and modulation. The above problem can be addressed by designing the source coding and modulation schemes jointly, a framework known as *joint source coding and modulation* (JSCM). Enabled by deep learning, JSCM leverages training data to automatically learn the best compression codes and modulation schemes in an end-to-end fashion. Signals are thus directly mapped onto a complex-valued constellation space, and the resulting representation is transmitted through the noisy channel.

In the context of complex signals (like images), and under extreme compression requirements (as in underwater communication), the selection of the features to be compressed strikes different 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 [3], 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.

The RDPC function. 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 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., $p_{\widehat{\boldsymbol{Y}}|\boldsymbol{Y}}(\widehat{\boldsymbol{y}}|\boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{y}\,, \boldsymbol{\Sigma}\big)$, 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.

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:

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

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 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 $i \ne j$. 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).

Our approach. As existing characterizations of tradeoffs between, for example, distortion and perception [3], rate, distortion, and perception [4], or classification error, distortion, and perception [5], 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 [3]–[5], as we consider vector signals (not necessarily scalar) and their compression in terms of dimensionality. Furthermore, channel capacity is evaluated at test time by injecting different levels of channel noise, obtaining different rates. 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, leveraging the capacity of generative adversarial networks (GANs) to model probability distributions [6], we propose to use inverse-domain GAN (ID-GAN) [7] 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 [6], ID-GAN [7] learns how to map not only a latent code to an image, but also an image to a latent code. To understand the behavior of this algorithm, we also propose a method that attempts to directly solve (3) under simple assumptions, e.g., that there are only two classes ($L = 2$) and that the encoder and decoder are linear maps. Despite these simplifications, experimental results show the same type of behavior for both algorithms.

Contributions. 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: one based on inverse-domain GAN (ID-GAN) [7], and a simpler one to understand its behavior. The proposed inverse-domain GAN algorithm can transmit images under extreme compression rates, handling low-capacity channels and preserving semantic information, perception quality, and reconstruction fidelity.
- We upper bound the optimal value in (3) when the input signal is a Gaussian mixture model (GMM). To achieve this, we obtain a new bound on the Wasserstein-1 distance between GMMs in terms of their parameters.
- Experiment results show that the proposed ID-GAN algorithm outperforms both a traditional method with source coding and modulation designed separately (JPEG+LDPC+BPSK) and recent deep JSCM algorithms like D-JSCC [2] and AE+GAN [8].

Organization. We overview related work in Section II and characterize the tradeoff problem (3) in Section III. Section IV then describes the proposed JSCM algorithms, whose performance is assessed in Section V. We conclude in Section VI.

II. Related Work

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

A. Joint Source-Channel Coding and Modulation (JSCM)

As mentioned, JSCM schemes achieve better reconstruction than the classical source-channel separation method. They can be divided into two categories: methods that focus on channel transmission, and methods that focus on source coding.

JSCM for channel transmission. The main goal of JSCM for channel transmission is to mitigate the effects of channel noise. 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.[3]

Inspired by D-JSCC, subsequent work has designed retrieval-oriented image compression schemes [9], used channel feedback to improve the quality of transmission [10], and considered an adaptive bandwidth to transmit information progressively [11]. Furthermore, [12] designed an end-to-end approach for D-JSCC with channel state information (CSI) feedback. The main idea is to apply a non-linear transform network to compress both the data and the CSI. Finally, [13] designed a scheme for orthogonal frequency division multiplexing (OFDM) transmission that directly maps the source images onto complex-valued baseband samples.

JSCM for source coding. Techniques used in JSCM for source coding vary according to the domain of the data, e.g., text, image, video, or multimodal data. For example, the JSCM system designed in [14] used a recurrent neural network for transmitting text. Also focusing on text transmission, [15] proposed a semantic communication system (DeepSC) based

[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 $p = q$.

[3] Although the method in [2] is named JSCC, it maps the compressed images directly onto the constellation space (IQ domain). So it performs modulation rather than source-coding, and is therefore an JSCM method.

on a transformer and, to evaluate performance, also a novel metric to measure sentence similarity. DeepCS was extended in [16] for speech transmission. JSCM has also been applied to the transmission of multimodal data. For instance, [17] 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, [18] designed a coarse-to-fine multitask semantic model using an attention mechanism.

GAN-based compression. Most algorithms for image transmission are based on autoencoders [19], e.g., [2], [9]–[11], [20]. 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 generative adversarial networks (GANs) [6]. A GAN is a generative model that learns, without supervision, both a low-dimensional representation of the data and its distribution [21]. This gives them the potential to achieve extreme compression without undermining image perception quality. For example, [8] 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. The algorithm we propose, ID-GAN, requires less training (as encoder and decoder are trained separately), but attains a performance similar or better than AE+GAN [8].

Also related to our work, [22] 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 [23], as a regularizer together with a distortion loss that aligns with human perception, LPIPS [24]. 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 [22] 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 [22] uses a pre-trained GAN for the decoder. Finally, while training StyleGAN-2 in [22] (on a database of faces) requires tremendous computational resources, training our networks takes more modest resources.

B. Tradeoff Analyses

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

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

where $I(\widehat{\boldsymbol{X}}, \boldsymbol{X})$ 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, [3] 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 [3], [4] 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 $I(\widehat{\boldsymbol{X}}, \boldsymbol{X})$ 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 [4] 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 [25] further improved on [4] 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 [5] analyzed instead the classification-distortion-perception (CDP) tradeoff, i.e., a modification of problem (3) in which $\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}})$ is minimized under the first two constraints (and 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 in distortion, perception, and classification error. Furthermore, we show the existence of a strict tradeoff, rather than just a simple tradeoff (as in [3]–[5]) between rate and all the constraints of (3).

III. The RDPC Tradeoff

Recall our multiclass signal model in (1) and the channel model in (2). Recall also that we assume a Gaussian channel $p_{\widehat{\boldsymbol{Y}}|\boldsymbol{Y}}(\widehat{\boldsymbol{y}}|\boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{y}\,, \boldsymbol{\Sigma}\big)$, 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 $c_0 : \mathbb{R}^n \to \{1, \ldots, L\}$ which, for $l = 1, \ldots, L$, decides $c_0(\widehat{\boldsymbol{X}}) = l$ whenever $\widehat{\boldsymbol{X}}$ belongs to a fixed region $\mathcal{R}_l \subset \mathbb{R}^n$. The probability of classification error is then

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

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 the Supplementary Material. □

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 [26], [27]. The same assumption was used in [3]–[5]. The theorem says that if we optimize the channel for the smallest possible rate, the encoding-decoding system cannot achieve arbitrarily small distortion, semantic 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 in (3). Note that while prior work [3]–[5] 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.

IV. Algorithms for Joint Source Coding and Modulation

As solving the RDPC problem in (3) in full generality (i.e., non-parametrically) is difficult, we propose two algorithms that approximately solve that problem. Our main algorithm is described in Section IV-A and is inspired on inverse-domain GAN (ID-GAN) [7]. As will be shown in Section V, it can handle extreme compression ratios while preserving distortion, perception, and classification metrics. Then, in Section IV-B, we propose an algorithm that tries to directly solve (3) under a set of simplifying assumptions. This will enable to draw some insights on the performance of the former algorithm.

A. Inverse-domain GAN compression

As formulated in (3), our goal is to design an encoder-decoder pair and channel noise levels that preserve image reconstruction quality, perception quality, and classification accuracy. Neural networks, and GANs [6] in particular, are a powerful tool to achieve this. 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 [7] which, however, was proposed for a task different from JSCM. Specifically, given an (adversarially-trained) image generator, the goal in [7] 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.

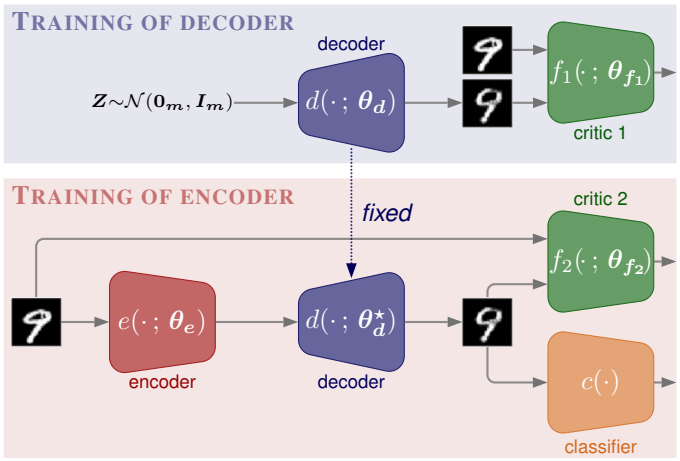


Fig. 1. Proposed ID-GAN framework for solving the RDPC problem in JSCM. The generator/decoder is first trained adversarially with critic 1 in the first step. The decoder is then fixed and coupled with an encoder, which is in turn trained with critic 2 in order to preserve both reconstruction quality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

Proposed scheme. Fig. 1 shows our framework for JSCM. As in ID-GAN [7], 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 [6], [28]. 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). This will enable us to compute semantically-meaningful latent codes to be transmitted along a communication channel. In other words, in (2), $p_{\boldsymbol{Y}|\boldsymbol{X}}$ is implemented by $e(\cdot\,;\boldsymbol{\theta}_e)$, and $p_{\widehat{\boldsymbol{Y}}|\widehat{\boldsymbol{X}}}$ is implemented by $d(\cdot\,;\boldsymbol{\theta}_d)$. To apply the scheme in a JSCM scenario, we normalize the power in the last layer of encoder and evaluate the system for different transmission rates. Next, we explain the training processes of the decoder and encoder in more detail.

Training the decoder. To train the decoder $d(\cdot\,;\boldsymbol{\theta}_d)$ and critic f_1 as in Fig. 1 (top), we leverage the Wasserstein GAN (WGAN) [28] 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 $\boldsymbol{Z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$:

$$W_1(p_r, p_{\boldsymbol{\theta}_d}) = \sup_{\|f_1\|_L\le 1} \mathbb{E}_{\boldsymbol{X}\sim p_r}[f_1(\boldsymbol{X})] - \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{\theta}_d}}[f_1(\boldsymbol{X})]\,, \tag{6}$$

where the supremum is over the functions $f_1 : \mathbb{R}^n \to \mathbb{R}$ that are 1-Lipschitz continuous. Notice that (6) is given in Kantarovich-Rubinstein dual form.[4] While the critic f_1 is

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

found by maximizing the argument in (6), 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 [6], which used instead the Jensen-Shannon divergence. To enforce Lipschitz-continuity of the critic f_1, [28] proposed to limit its parameters to a small box around the origin. The work in [29], 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 [29] for finding critic f_1:

$$L_{f_1} = \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{\theta}_d}}[f_1(\boldsymbol{X})] - \mathbb{E}_{\boldsymbol{X}\sim p_r}[f_1(\boldsymbol{X})] + \lambda_g \, \mathbb{E}_{\widetilde{\boldsymbol{X}}\sim p_{\boldsymbol{\theta}_d, r}}\Big[\big(\|\nabla_{\widetilde{\boldsymbol{x}}} f_1(\widetilde{\boldsymbol{X}})\|_2 - 1\big)^2\Big], \quad (7)$$

where $\lambda_g \geq 0$, and $\widetilde{\boldsymbol{X}}$ is a point sampled uniformly in 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. Notice that the third term in (7) eliminates the need to constrain the critic to be Lipschitz-continuous [constraint in (6)]; see [29] for more details. In turn, the parameters $\boldsymbol{\theta}_d$ of the decoder are found by minimizing (6) which, when f_1 is fixed, is equivalent to minimizing

$$L_{\boldsymbol{\theta}_d} = -\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{\theta}_d}}[f_1(\boldsymbol{X})] = -\mathbb{E}_{\boldsymbol{Z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\big[f_1\big(d(\boldsymbol{Z}\,;\,\boldsymbol{\theta}_d)\big)\big]\,. \quad (8)$$

During training, we update the parameters of the critic n_{critic} times while keeping the parameters of the decoder fixed. This guarantees that the supremum in (6) is reasonably well computed. See [28], [29] for details.

Training the encoder. After training the decoder $d(\cdot\,;\,\boldsymbol{\theta}_d)$, we fix it 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, but now we take into account the quality of reconstruction and its semantic meaning. 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. The full loss is associated to the encoder is

$$L_e = -\mathbb{E}\big[f_2(d(e(\boldsymbol{X})))\big] + \mathbb{E}\Big[\lambda_d \big\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\big\|_2^2 + \lambda_c \text{CE}\Big(c\big(\widehat{\boldsymbol{X}}\big), \text{class}\big(\boldsymbol{X}\big)\Big)\Big], \quad (9)$$

where the expectations are with respect to $\boldsymbol{X}$. In (9), $\text{CE}(\cdot,\cdot)$ is the cross-entropy loss, and $\lambda_d, \lambda_c \geq 0$ are hyperparameters that balance the MSE and cross-entropy losses. Note that $\widehat{\boldsymbol{X}}$ is a deterministic function of $\boldsymbol{X}$, i.e., $\widehat{\boldsymbol{X}}(\boldsymbol{X})$. For simplicity, we omit this fact as well as that $\boldsymbol{X} \sim p_r$. The loss associated to the critic f_2 is, in turn, akin to (7):

$$L_{f_2} = \mathbb{E}\Big[f_2\big(d(e(\boldsymbol{X}))\big)\Big] - \mathbb{E}\big[f_2(\boldsymbol{X})\big] + \lambda_g \, \mathbb{E}_{\boldsymbol{X},\epsilon}\Big[\big(\big\|\nabla_{\widetilde{\boldsymbol{x}}} f_2\big(\widetilde{\boldsymbol{X}}\big)\big\|_2 - 1\big)^2\Big], \quad (10)$$

where $\widetilde{\boldsymbol{X}} = (1-\epsilon)\boldsymbol{X} + \epsilon\, d(e(\boldsymbol{X}))$ and $\epsilon \sim \mathcal{U}(0,1)$ is uniformly distributed in $[0,1]$. Note that we omitted the parameters of the encoder and decoder for simplicity.

Algorithm 1 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}, loss hyperparameters λ_g, λ_d, λ_p, λ_c

Initialization: Set encoder $\boldsymbol{\theta}_e^{(0)}$ and critic $\boldsymbol{\theta}_f^{(0)}$ parameters 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: $L_{f_2}^{(1)} = \frac{1}{S}\sum_{s\in\mathcal{S}_j} f_2\Big(d\big(e(\boldsymbol{x}^{(s)}\,;\,\boldsymbol{\theta}_e^{(j)})\big)\,;\,\boldsymbol{\theta}_f^{(k)}\Big)$
6: $L_{f_2}^{(2)} = \frac{1}{S}\sum_{s\in\mathcal{S}_j} f_2\Big(\boldsymbol{x}^{(s)}\,;\,\boldsymbol{\theta}_f^{(k)}\Big)$
7: **for** $s \in \mathcal{S}_j$ **do**
8: Draw $\epsilon \sim \mathcal{U}(0,1)$ randomly
9: $\widetilde{\boldsymbol{x}}^{(s)} = (1-\epsilon)\cdot\boldsymbol{x}^{(s)} + \epsilon \cdot d\big(e\big(\boldsymbol{x}^{(s)}\,;\,\boldsymbol{\theta}_e^{(j)}\big)\big)$
10: **end for**
11: $L_{f_2}^{(3)} = \frac{1}{S}\sum_{s\in\mathcal{S}_j}\Big(\big\|\nabla_{\widetilde{\boldsymbol{x}}} f_2(\widetilde{\boldsymbol{x}}^{(s)})\big\|_2 - 1\Big)^2$
12: $L_{f_2} = L_{f_2}^{(1)} - L_{f_2}^{(2)} + \lambda_g L_{f_2}^{(3)}$
13: $\boldsymbol{\theta}_f^{(k+1)} = \text{Adam}\Big(\boldsymbol{\theta}_f^{(k)},\, L_{f_2},\, \alpha,\, \beta_1,\, \beta_2\Big)$
14: **end for**
15: $\boldsymbol{\theta}_f^{(j)} = \boldsymbol{\theta}_f^{(n_{\text{critic}})}$
16: **for** $s \in \mathcal{S}_j$ **do**
17: $\widehat{\boldsymbol{x}}^{(s)} = d\Big(e\big(\boldsymbol{x}^{(s)}\,;\,\boldsymbol{\theta}_e^{(j)}\big)\Big)$
18: **end for**
19: $L_e = \frac{1}{S}\sum_{s\in\mathcal{S}_j} -f_2\Big(\widehat{\boldsymbol{x}}^{(s)}\,;\,\boldsymbol{\theta}_f^{(j)}\Big) + \lambda_d\Big\|\boldsymbol{x}^{(s)} - \widehat{\boldsymbol{x}}^{(s)}\Big\|_2^2 + \lambda_c \text{CE}\big(c(\widehat{\boldsymbol{x}}^{(s)}), \boldsymbol{\ell}^{(s)}\big)$
20: $\boldsymbol{\theta}_e^{(j+1)} = \text{Adam}\Big(\boldsymbol{\theta}_e^{(j)},\, L_e,\, \alpha,\, \beta_1,\, \beta_2\Big)$
21: **end for**

The complete training procedure of the encoder is shown in Algorithm 1. 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 and associated critic, 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 4-14 performs n_{critic} iterations of Adam to minimize the critic loss in (10) 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 (6) between the real data $\boldsymbol{X}$ and the reconstructed one $\widehat{\boldsymbol{X}} = d\big(e\big(\boldsymbol{X}\big)\big)$. The terms in (10) are computed separately, with the last term requiring the creation of the intermediate variables $\widetilde{\boldsymbol{x}}^{(s)}$ in steps 7-10. 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 (9) and thus to update the encoder parameters $\boldsymbol{\theta}_e$. This requires passing each image in the batch through the encoder and decoder to create $\widehat{\boldsymbol{x}}^{(t)}$, as in steps 16-18.

Connection with (3). Although minimizing (9) constitutes a problem different from (3), the three terms in (9) capture the quantities constrained in (3). That is, $\mathbb{E}\big[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_2^2\big]$ in (9) is a particular instance of $\mathbb{E}[\Delta(\boldsymbol{X}, \widehat{\boldsymbol{X}})]$ in the distortion constraint

of (3); the term L_{f_2} in (9) approximates the Wasserstein distance between the distributions of $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$, a particular instance of $d(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}})$ in the perception constraint of (3); and the cross-entropy term in (9) is a proxy of the classification error term in the last constraint of (3). However, rather than minimizing the rate as in (3), here we fix the dimension m of the latent variable and evaluate the performance of the resulting scheme for different values of m. Next, we attempt to solve (3) directly under some simplifying assumptions.

B. Solving RDPC: Problem Formulation

To develop insight on how the proposed framework strikes a tradeoff between achievable rate, distortion, perception, and classification error, we now analyze a simplified version of problem (3). We make the following assumptions.

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

1) *The source signal $\boldsymbol{X} \in \mathbb{R}^n$ is drawn from a two-class Gaussian mixture model, i.e.,*

$$\boldsymbol{X}|H_0 \sim \mathcal{N}(\boldsymbol{0}_n, \boldsymbol{I}_n) \tag{11a}$$
$$\boldsymbol{X}|H_1 \sim \mathcal{N}(\boldsymbol{c}_n, \boldsymbol{I}_n)\,, \tag{11b}$$

   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, i.e., they are implemented by full-rank matrices $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{D} \in \mathbb{R}^{n\times m}$, respectively. In particular, $p_{\boldsymbol{Y}|\boldsymbol{X}}$ and $p_{\widehat{X}|\widehat{Y}}$ in (2) are deterministic.*
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 $W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}})$ as a metric for perception, where $\boldsymbol{X} \sim p_{\boldsymbol{X}}$ and $\widehat{\boldsymbol{X}} \sim p_{\widehat{\boldsymbol{X}}}$.*
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.*

Note that a linear encoder and decoder as in 2) can model a linearization of a fully trained neural network around an optimal point; see, e.g., work on the neural tangent kernel [30], [31]. Assumptions 1) and 2) imply that the reconstructed signal $\widehat{\boldsymbol{X}}$ is also a Gaussian mixture model. To see this, first note that the output signal is $\widehat{\boldsymbol{Y}} = \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{X} + \boldsymbol{N}\big)$, where $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{\Sigma})$. Since the sum of two Gaussian random variables is also Gaussian, we obtain

$$\widehat{\boldsymbol{X}}\,|\,H_0 \;\sim\; \mathcal{N}\Big(\boldsymbol{0}_n\,,\, \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma}\big)\boldsymbol{D}^\top\Big), \tag{12a}$$
$$\widehat{\boldsymbol{X}}\,|\,H_1 \;\sim\; \mathcal{N}\Big(\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n\,,\, \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma}\big)\boldsymbol{D}^\top\Big)\,. \tag{12b}$$

Note that the covariance matrix $\widehat{\boldsymbol{\Sigma}} := \boldsymbol{D}(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma})\boldsymbol{D}^\top$ of both distributions in (12) is rank deficient, since $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ with $n > m$, making the distributions 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, $|\widehat{\boldsymbol{\Sigma}}|$ is the product of the positive entries of $\boldsymbol{\Lambda}$.

Problem formulation. Under Assumption 2, (3) becomes

$$\begin{aligned} R(D,P,C) = \min_{\boldsymbol{E},\,\boldsymbol{\Sigma},\,\boldsymbol{D}} \quad & \textstyle\sum_{i=1}^m \log\big(1 + \frac{1}{\boldsymbol{\Sigma}_{ii}}\big) \\ \text{s.t.} \quad & \mathbb{E}\big[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_2^2\big] \leq D \\ & W_1(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}}) \leq P \\ & \epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq C\,, \end{aligned} \tag{13}$$

where we omitted the dependence of $\widehat{\boldsymbol{X}}$ on $\boldsymbol{E}$ and $\boldsymbol{D}$ for simplicity. Despite the simplifications under Assumption 2, problem (13) 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 (13), we first condition the expected values:

$$\begin{aligned} \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\big\|_2^2\Big] = \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\big\|_2^2 \,\big|\, H_0\Big] \cdot p_0 \\ + \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\big\|_2^2 \,\big|\, H_1\Big] \cdot p_1\,, \end{aligned} \tag{14}$$

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

$$\begin{aligned} \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\big\|_2^2 \,\big|\, H_l\Big] = \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}}\big\|_2^2 \,\big|\, H_l\Big] \\ - 2\,\mathbb{E}\Big[\widehat{\boldsymbol{X}}^\top \boldsymbol{X} \,\Big|\, H_l\Big] + \mathbb{E}\Big[\big\|\boldsymbol{X}\big\|_2^2 \,\big|\, H_l\Big]\,. \end{aligned} \tag{15}$$

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

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

where we used the linearity of the trace $\mathrm{tr}(\cdot)$ in the second equality, and (11a) in the third equality. Similarly, under H_1,

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

due to (11b). Similar reasoning applies to the first term of (15):

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

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 (15) can be rewritten for $l = 0, 1$ as

$$\begin{aligned} \mathbb{E}\Big[\widehat{\boldsymbol{X}}^\top \boldsymbol{X} \,\Big|\, H_l\Big] &= \mathbb{E}\Big[\boldsymbol{X}^\top \boldsymbol{D}\big(\boldsymbol{E}\boldsymbol{X} + \boldsymbol{N}\big) \,\Big|\, H_l\Big] \\ &= \mathbb{E}\big[\mathrm{tr}\big(\boldsymbol{E}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{D}\big) \,\big|\, H_l\big] + \mathbb{E}\Big[\boldsymbol{X}^\top \boldsymbol{D}\boldsymbol{N} \,\Big|\, H_l\Big] \\ &= \mathrm{tr}\Big(\boldsymbol{E}\,\mathbb{E}\big[\boldsymbol{X}\boldsymbol{X}^\top \,|\, H_l\big]\boldsymbol{D}\Big)\,, \end{aligned} \tag{16}$$

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 (15)-(16) into (14),

$$\begin{aligned} \mathbb{E}\Big[\big\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\big\|_2^2\Big] = \Big[\mathrm{tr}\big(\widehat{\boldsymbol{\Sigma}}\big) - 2\,\mathrm{tr}\big(\boldsymbol{E}\boldsymbol{D}\big) + n\Big]p_0 + \Big[\mathrm{tr}\big(\widehat{\boldsymbol{\Sigma}}\big) \\ + \boldsymbol{c}_n^\top \boldsymbol{E}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n - 2\mathrm{tr}\Big(\boldsymbol{E}(\boldsymbol{I}_n + \boldsymbol{c}_n\boldsymbol{c}_n^\top)\boldsymbol{D}\Big) + n + \big\|\boldsymbol{c}_n\big\|_2^2\Big]p_1\,. \end{aligned} \tag{17}$$

Perception constraint. We now consider the perception constraint in (13), 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 $\boldsymbol{X} \sim p_{\boldsymbol{X}} = \mathcal{N}(\boldsymbol{\mu_X}, \boldsymbol{\Sigma_X})$ and $\boldsymbol{Y} \sim p_{\boldsymbol{Y}} = \mathcal{N}(\boldsymbol{\mu_Y}, \boldsymbol{\Sigma_Y})$ be two Gaussian random vectors with means $\boldsymbol{\mu_X}, \boldsymbol{\mu_Y} \in \mathbb{R}^n$ and positive semidefinite covariance matrices $\boldsymbol{\Sigma_X}, \boldsymbol{\Sigma_Y} \succeq \mathbf{0_{n\times n}}$. It can be shown that the squared Wasserstein-2 distance between them is [32], [33]

$$\|\boldsymbol{\mu_X} - \boldsymbol{\mu_Y}\|_2^2 + \operatorname{tr}\left(\boldsymbol{\Sigma_X} + \boldsymbol{\Sigma_Y} - 2\left(\boldsymbol{\Sigma_Y}^{\frac{1}{2}}\boldsymbol{\Sigma_X}\boldsymbol{\Sigma_Y}^{\frac{1}{2}}\right)^{\frac{1}{2}}\right).$$

In the case where $\boldsymbol{\Sigma_X}$ and $\boldsymbol{\Sigma_Y}$ commute, i.e., $\boldsymbol{\Sigma_X}\boldsymbol{\Sigma_Y} = \boldsymbol{\Sigma_Y}\boldsymbol{\Sigma_X}$, the expression simplifies to

$$W_2^2(p_{\boldsymbol{X}}, p_{\boldsymbol{Y}}) = \left\|\boldsymbol{\mu_X} - \boldsymbol{\mu_Y}\right\|_2^2 + \left\|\boldsymbol{\Sigma_X}^{\frac{1}{2}} - \boldsymbol{\Sigma_Y}^{\frac{1}{2}}\right\|_F^2, \tag{18}$$

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 4 [or (6)] 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 Gaussian mixture model following (11) [resp. (12)], in which the probability of hypothesis H_0 is p_0 and of hypothesis H_1 is $p_1 = 1 - p_0$. Then,*

$$W_1(p_{\boldsymbol{X}}, p_{\widehat{\boldsymbol{X}}}) \leq \left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F + \left\|\boldsymbol{D}\boldsymbol{E}\boldsymbol{c_n} - \boldsymbol{c_n}\right\|_2 \cdot p_1\,. \tag{19}$$

Proof. See the Supplementary Material. □

To enforce the second constraint in (13), we will thus impose the right-hand side of (19) to be bounded by P.

Classification constraint. We now address the last constraint of (13). 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 Gaussian mixture model $\boldsymbol{X} \sim p_0\,\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) + p_1\,\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, the bound is

$$\begin{aligned}\mathbb{P}(\text{error}^\star) &\leq \sqrt{p_0\,p_1}\int_{\mathbb{R}^n}\sqrt{p_{\boldsymbol{X}|H_0}(\boldsymbol{x})\,p_{\boldsymbol{X}|H_1}(\boldsymbol{x})}\;\mathrm{d}\boldsymbol{x}\\ &= \sqrt{p_0\,p_1}\,\exp\left[-\frac{1}{8}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^\top\left[\frac{\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1}{2}\right]^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\right.\\ &\qquad\left.-\frac{1}{2}\log\frac{\left|(\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1)/2\right|}{\sqrt{|\boldsymbol{\Sigma}_0||\boldsymbol{\Sigma}_1|}}\right],\end{aligned} \tag{20}$$

where $|\cdot|$ is the determinant of a matrix, and error* is the classification error achieved by a Bayes classifier. We apply (20) to $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$, whose models are in (11) and (12). Thus, $\boldsymbol{\mu}_0 = \mathbf{0}_n$, $\boldsymbol{\mu}_1 = \boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n$, and $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{D}(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma})\boldsymbol{D}^\top$. Hence,

$$\begin{aligned}\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) &= \mathbb{P}\big(\text{class}(\widehat{\boldsymbol{X}}) \neq c_0(\widehat{\boldsymbol{X}})\big)\\ &\leq \sqrt{p_0 p_1}\,\exp\left[-\frac{1}{8}\boldsymbol{c}_n^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n\right].\end{aligned} \tag{21}$$

So, in (13), rather than bounding $\epsilon_{c_0}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq C$, we impose instead that the right-hand side of (21) is upper bounded by C, which is equivalent to

$$\boldsymbol{c}_n^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n \geq -8\log\frac{C}{\sqrt{p_0 p_1}}\,. \tag{22}$$

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

Bound on RDPC. Instead of solving (13), we will aim to solve a problem that upper bounds its optimal value:

$$\begin{aligned}R(D,P,C) \leq \min_{\boldsymbol{E},\boldsymbol{\Sigma},\boldsymbol{D}}\quad & \textstyle\sum_{i=1}^m \log\left(1+\frac{1}{\boldsymbol{\Sigma}_{ii}}\right)\\ \text{s.t.}\quad & (17) \leq D\\ & (19) \leq P\\ & (22)\,,\end{aligned} \tag{23}$$

where (17) and (19) 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 (13). The resulting problem, however, is still nonconvex and will require approximation techniques.

C. Solving RDPC: Heuristic Algorithm

Solving (23) is difficult, as it is nonconvex and has an infinite number of solutions. Indeed, $\boldsymbol{E}$ and $\boldsymbol{D}$ appear in the constraints of (23) always as the product $\boldsymbol{D}\boldsymbol{E}$. Thus, if $(\boldsymbol{E}^\star, \boldsymbol{\Sigma}^\star, \boldsymbol{D}^\star)$ is a solution of (23) 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 (23).

Design of $\widehat{\boldsymbol{\Sigma}}$. While the original signals in (11) have non-degenerate distributions, the decoded signals in (12) 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 (12) 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 $\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n = \mathbf{0}_n$), 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 (19) requires $\|\boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n - \boldsymbol{c}_n\|_2$ to be small].

To avoid this, we first generate the (degenerate) covariance matrix $\widehat{\boldsymbol{\Sigma}} := \boldsymbol{D}(\boldsymbol{E}\boldsymbol{E}^\top + \boldsymbol{\Sigma})\boldsymbol{D}^\top$ 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

$$\nabla_{\boldsymbol{\sigma}} h_p(\boldsymbol{\Sigma}) = \left[\operatorname{diag}\left(\boldsymbol{D}^\top\left(\boldsymbol{I_m} - 2\left(\boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma D}^\top\right)^{-1}\right)\boldsymbol{D}\right)\right] \Big/ \left(P_k - \left\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F^2\right) \tag{29a}$$

$$\nabla_{\boldsymbol{\sigma}} h_c(\boldsymbol{\Sigma}) = \frac{\operatorname{diag}\left(\boldsymbol{D}^\top\left(\boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma D}^\top\right)^{-1}\boldsymbol{DEcc}^\top\boldsymbol{E}^\top\boldsymbol{D}^\top\left(\boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma D}^\top\right)^{-1}\boldsymbol{D}\right)}{\boldsymbol{c_n}^\top\boldsymbol{E_k}^\top\boldsymbol{D_k}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{c_n} + 8\log\frac{C}{\sqrt{p_0 p_1}}} \tag{29b}$$

$\widehat{\boldsymbol{\Sigma}} = \boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma_{k-1}D}^\top$. 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 a optimization problem:

$$\begin{aligned}\min_{\boldsymbol{E},\boldsymbol{D}} \quad & \tfrac{1}{2}\left\|\boldsymbol{I}_n - \boldsymbol{DEE}^\top\boldsymbol{D}^\top\right\|_F^2 + \tfrac{1}{2}\left\|\boldsymbol{D\Sigma_{k-1}D}^\top\right\|_F^2 \\ & \qquad + \tfrac{1}{2}\left\|\boldsymbol{c_n} - \boldsymbol{DEc_n}\right\|_2^2 \\ \text{s.t.} \quad & \widehat{\boldsymbol{\Sigma}} = \boldsymbol{DEE}^\top\boldsymbol{D}^\top + \boldsymbol{D\Sigma_{k-1}D}^\top,\end{aligned} \tag{24}$$

which, eliminating the constraint, can be written as

$$\begin{aligned}\min_{\boldsymbol{E},\boldsymbol{D}} \ & \frac{1}{2}\left\|\boldsymbol{I}_n - \widehat{\boldsymbol{\Sigma}} + \boldsymbol{D\Sigma_{k-1}D}^\top\right\|_F^2 + \frac{1}{2}\left\|\boldsymbol{D\Sigma_{k-1}D}^\top\right\|_F^2 \\ & \qquad + \frac{1}{2}\left\|\boldsymbol{c_n} - \boldsymbol{DEc_n}\right\|_2^2.\end{aligned} \tag{25}$$

We apply gradient descent to (25) 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 (25) are

$$\frac{\partial g(\boldsymbol{E},\boldsymbol{D})}{\partial \boldsymbol{E}} = \boldsymbol{D}^\top(\boldsymbol{DEc_n} - \boldsymbol{c_n})\boldsymbol{c_n}^\top \tag{26a}$$

$$\begin{aligned}\frac{\partial g(\boldsymbol{E},\boldsymbol{D})}{\partial \boldsymbol{D}} = {} & 4\boldsymbol{D\Sigma_k D}^\top\boldsymbol{D\Sigma_{k-1}} + 2(\boldsymbol{I_n} - \widehat{\boldsymbol{\Sigma}})\boldsymbol{D\Sigma_{k-1}} \\ & + (\boldsymbol{DEc_n} - \boldsymbol{c_n})\boldsymbol{c_n}^\top\boldsymbol{E}^\top.\end{aligned} \tag{26b}$$

Finding $\boldsymbol{\Sigma}$. Once the encoder-decoder pair is fixed at $(\boldsymbol{E_k}, \boldsymbol{D_k})$, we find the diagonal rate matrix $\boldsymbol{\Sigma} := \operatorname{Diag}(\boldsymbol{\sigma}) := \operatorname{Diag}(\sigma_1, \ldots, \sigma_m)$ by applying a barrier method [35] to (23), i.e., we solve a sequence of problems in t, each of which is

$$\min_{\boldsymbol{\Sigma}=\operatorname{Diag}(\boldsymbol{\sigma})} \ t\, h_r(\boldsymbol{\sigma}) - \lambda_D h_D(\boldsymbol{\Sigma}) - \lambda_P h_P(\boldsymbol{\Sigma}) - \lambda_C h_C(\boldsymbol{\Sigma}), \tag{27}$$

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

$$h_r(\boldsymbol{\sigma}) = \sum_{i=1}^{m} \log\left(1 + \frac{1}{\sigma_i}\right) \tag{28a}$$

$$h_d(\boldsymbol{\Sigma}) = \log\left[D_k - \operatorname{tr}\left(\boldsymbol{D_k\Sigma D_k}^\top\right)\right] \tag{28b}$$

$$h_p(\boldsymbol{\Sigma}) = \log\left[P_k - \left\|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F^2\right] \tag{28c}$$

$$h_c(\boldsymbol{\Sigma}) = \log\left[\boldsymbol{c_n}^\top\boldsymbol{E_k}^\top\boldsymbol{D_k}^\top\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{-1}\boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{c_n} + 8\log\frac{C}{\sqrt{p_0 p_1}}\right]. \tag{28d}$$

where $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}} = \boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{E_k}^\top\boldsymbol{D_k}^\top + \boldsymbol{D_k\Sigma D_k}^\top$. In (28b), 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 (17) [including D]. To obtain (28c), note that imposing the right-hand side of (19) to be smaller than P is equivalent to $\left\|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{\frac{1}{2}} - \boldsymbol{I_n}\right\|_F^2 \leq \left(P - \|\boldsymbol{D_k}\boldsymbol{E_k}\boldsymbol{c_n} - \boldsymbol{c_n}\|_2 \cdot p_1\right)^2 =: P_k$. And $h_p(\boldsymbol{\Sigma})$ depends on $\boldsymbol{\Sigma}$ via $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}$. Finally, (28d) is the direct application of the log-barrier function to (22).

To solve each instance of (27), we apply again gradient descent. While the gradients of h_r in (28a) and h_d in (28b) can be computed directly, namely $\mathrm{d}h_r(\boldsymbol{\sigma})/\,\mathrm{d}\sigma_i = -1/(\sigma_i^2 + \sigma_i)$, for $i = 1, \ldots, m$, and $\nabla_{\boldsymbol{\sigma}} h_d(\boldsymbol{\Sigma}) = -\operatorname{diag}(\boldsymbol{D_k}^\top\boldsymbol{D_k})/[D_k - \operatorname{tr}(\boldsymbol{D_k\Sigma}, \boldsymbol{D_k}^\top)]$, where $\operatorname{diag}(\cdot)$ extracts the diagonal entries of a matrix into a vector, computing the gradients of h_p in (28c) and h_c in (28d) is more laborious. Their expressions are shown in (29) where, for simplicity, we omitted the iteration index.

RDPCO algorithm. We summarize all the above steps in Algorithm 2, which we name RDPCO for RDPC optimization. In our implementation and experiments, we set $p_0 = p_1 = 1/2$ and, in step 6, employ a constant learning rate of 10^{-4} for 2×10^4 iterations. Regarding the barrier method in steps 8-11, we initialize t as $t_0 = 0.01$ and update it with $\mu = 2$. That algorithm stops either when the number of iterations reaches $m/100$ (determined experimentally) or when the duality gap is below 0.01. The parameter ϵ in step 13 is set to 10^{-5}. To balance the terms in (27), we set $\lambda_D = 1/\log(D)$, $\lambda_P = 1/\log(P)$, and $\lambda_C = -1/\log(\sqrt{p_0\, p_1})$.

Algorithm 2 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 $\lambda_D, \lambda_P, \lambda_C$.

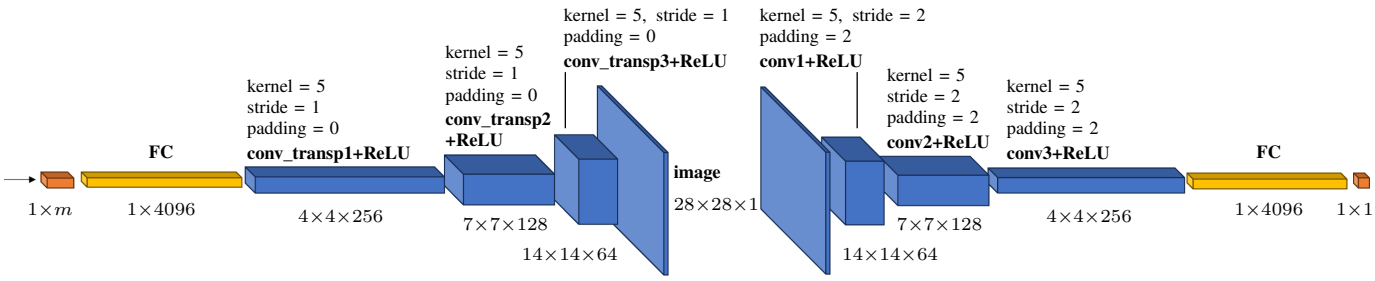
Initialization: $\boldsymbol{\Sigma_0} = \boldsymbol{I_m}$

Generate $\widehat{\boldsymbol{\Sigma}}$

1: Set $\widetilde{\boldsymbol{Q}} = \begin{bmatrix}\boldsymbol{c_n} & \boldsymbol{R}\end{bmatrix}$, 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 $\lambda_i \in [0,1]$, $i = 2, \ldots, m$ randomly and build $\boldsymbol{\Lambda} = \operatorname{Diag}(1, \lambda_2, \ldots, \lambda_m, 0, \ldots, 0) \in \mathbb{R}^{n \times n}$
4: Set $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{Q\Lambda Q}^\top$

Find $\boldsymbol{E}, \boldsymbol{D}, \boldsymbol{\Sigma}$

5: **for** $k = 1, \ldots, K$ **do**
6: Find $(\boldsymbol{E_k}, \boldsymbol{D_k})$ via gradient descent applied to (25) [cf. (26)]
7: Set $t = t_0$
8: **for** $r = 1, \ldots, \lceil m/100 \rceil$ **do**
9: Find $\boldsymbol{\Sigma_r}$ via gradient descent applied to (27)
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**



(a) Decoder (b) Critics

Fig. 2. 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.

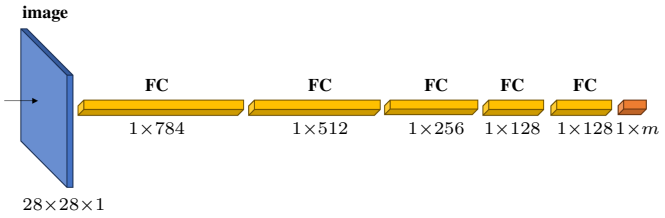


Fig. 3. Architecture of the encoder $e(\cdot\,;\boldsymbol{\theta}_e)$, consisting of fully connected layers of indicated dimensions.

V. Experimental Results

We now present our experiments to evaluate the performance of the proposed ID-GAN method (Algorithm 1, for the training of the encoder), and also to show how RDPCO (Algorithm 2) unveils insights about encoder unveils insights about the RDPC problem (3).

A. ID-GAN-based JSCM framework

Before the experimental setup, we explain how all the functions in the ID-GAN framework in Fig. 1 were implemented.

Network architectures. Fig. 2 shows the architectures of the decoder $d(\cdot\,;\boldsymbol{\theta}_d)$ [Fig. 2(a)] and of the critics f_1 and f_2 [Fig. 2(b)]. The latter have the same architecture, but they are initialized independently, with different seeds. The decoder in Fig. 2(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. 2(b)] is symmetric to that of the decoder. The input image is compressed via a convolutional network, whose output at the last layer is mapped to a probability vector with a sigmoid function. The architecture of the encoder, on the hand, downsamples the input image using only fully connected layers, as shown in Fig. 3. The network architecture of the classifier [cf. Fig. 1, bottom] is similar to the one of the encoder, except that we apply batch normalization after each fully-connected layer and use a leaky ReLU as activation. Also, 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. To illustrate the performance of the algorithms, we use the popular MNIST dataset [36], which has $60,000$ training images and $10,000$ test images of size 28×28. In Algorithm 1, 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 $S = 50$, $n_{\text{critic}} = 5$ iterations for the inner loop of the critic, and loss hyperparameters $\lambda_g = \lambda_d = \lambda_p = \lambda_c = 1$. Also, we ran the algorithm for just 5 epochs. The reason for such a small number is that, as described in Section IV-A, the decoder has already been trained when we run Algorithm 1. And the decoder required also only 8 epochs to train.

Algorithms. We compare the proposed method for JSCM, i.e., ID-GAN in Algorithm 1, against D-JSCC [2], the parallel autoencoder-GAN (AE+GAN) training method [8], and a traditional approach in which source coding and modulation are designed separately using JPEG and Huffman codes for source coding, $3/4$ LDPC codes for channel coding, and BPSK for modulation [37].

Metrics and comparison. For comparison metrics, we selected the mean squared error (MSE), the Fréchet inception distance (FID) [38], and classification error. FID captures perception quality by measuring the similarity between distributions of real and generated images. The smaller the FID, the closer both distributions are. In summary, the smaller all the metrics, the better the performance of the algorithm. 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 by the objective of (3), 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. To compare the different algorithms in a fair way, we thus fix rate, which is independent of the latent/compressed dimension m.

Results. Figs. 4(a)-4(c) show how the above metrics vary with the rate for the proposed ID-GAN with $\lambda_c = 30$, $\lambda_c = 100$, and $\lambda_d = 10$ (when one parameter is set, the remaining are equal to 1), as well as for D-JSCC [2] and AE+GAN [8]. Fig. 4(a) indicates that image distortion, measured by the MSE, decreases with transmission rate for all the algorithms. D-JSCC, however, outperforms all other methods, as it was designed to minimize image distortion. Indeed, according to the RDPC tradeoff (Theorem 1), if perception and

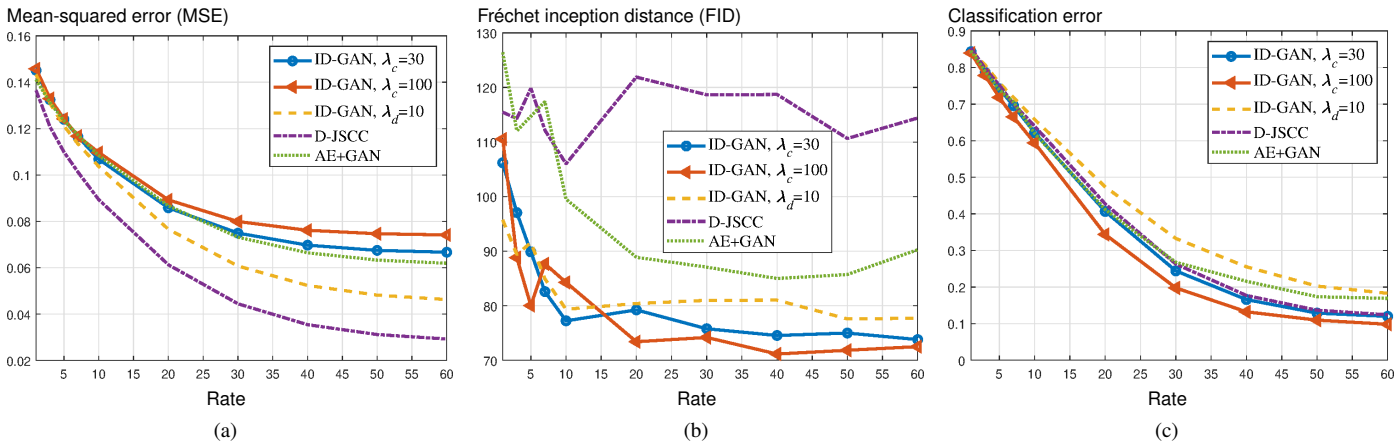


Fig. 4. Results for proposed ID-GAN (under different hyperparameters), D-JSCC [2], and AE+GAN training [8]. (a) mean-squared error (MSE), (b) Fréchet inception distance (FID), and (c) classification error as a function of the rate.

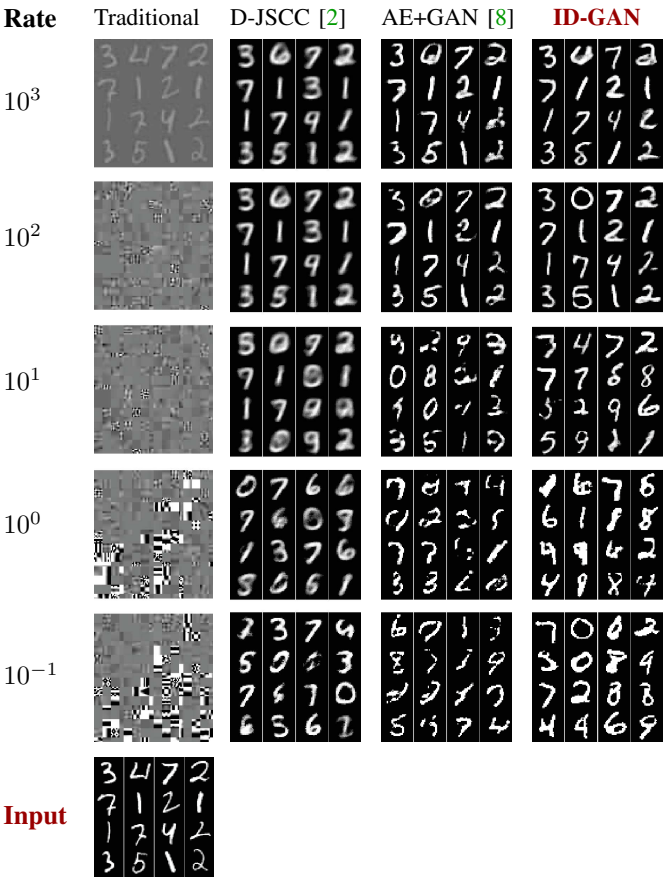
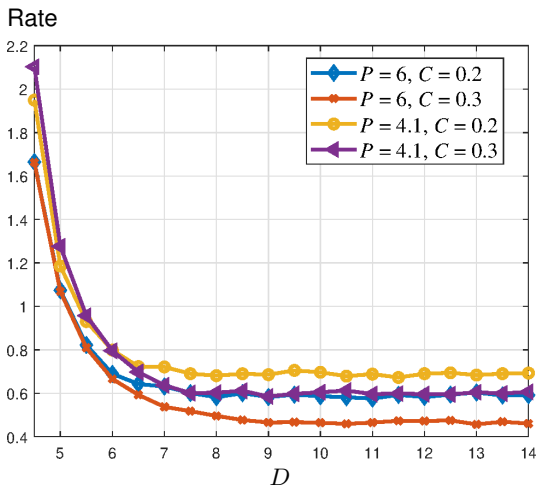


Fig. 5. Example images reconstructed by a traditional method (JPEG+LDPC+BPSK), D-JSCC, AE+GAN training, and proposed ID-GAN.

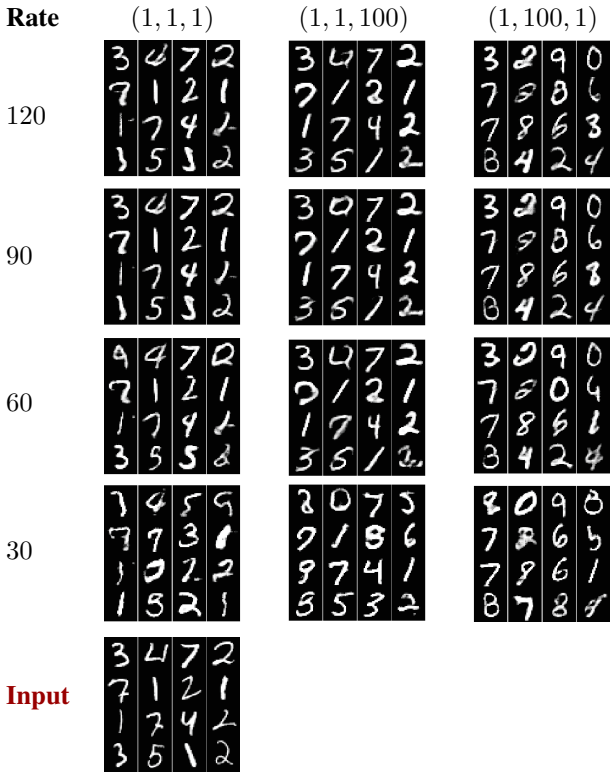
classification metrics are ignored, an algorithm can achieve a given level of distortion with a smaller transmission rate. The RDPC tradeoff also explains why increasing λ_d in ID-GAN yields a better MSE [the weight associated to the MSE loss in (9) increases], while increasing λ_c leads to a worse one. Fig. 4(b) shows that our algorithm outperforms D-JSCC and AE+GAN in terms of FID for all the values of the tested hyperparameters. This is because we use a GAN to match the distributions of real and generated images. Notice also that in our algorithm FID is essentially invariant to different hyperparameters, since the decoder/generator is fixed in the second stage of training [cf. Fig. 1]. However, when the transmission rate is small, from 0 to 10, FID in ID-GAN decreases drastically from 110 to 80. We attribute this behavior to a relatively large noise in the channel when the rate is small. As the decoder is fixed, a large noise in the output of the encoder $\boldsymbol{y}$ translates into images with low-quality in terms of perception. Fig. 4(c) illustrates again the RDPC tradeoff: when we increase λ_d in our algorithm, its classification performance decreases. On the other hand, as expected, increasing λ_c improves its classification performance.

In conclusion, these results indicate that by adopting a two-stage training process whereby the decoder is first trained and then kept fixed, we effectively stabilize the perception quality of the images. Compared to D-JSCC [2], this, however, comes at the expense of a slight increase in MSE.

Visual illustration on MNIST. Fig. 5 shows image examples from MNIST reconstructed by all the algorithms. In this experiment, the latent dimension m was fixed to 8 in D-JSCC [2], AE+GAN [8], and ID-GAN. The rate was then computed according to the expression in the objective of (3). For ID-GAN, we set $\lambda_d = \lambda_p = \lambda_c = 1$. The images that the traditional system in unable to reconstruct the images even when the rate is high, likely due to the quantization and compression in JPEG. When the rate decreases, we observe a *cliff effect*, with the image quality degrading abruptly. D-JSCC [2], on the other hand, is based on an autoencoder which, given the high bandwidth ratio, outputs blurry images at all rates. Yet, despite the degraded perceptual quality, it still preserves semantic information. The images reconstructed by AE+GAN training [8], on the other hand, have poor perceptual quality for low rates (1 and 0.1). This is likely because it is difficult to balance weights between multiple loss terms The proposed ID-GAN scheme, 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.



(a)



(b)

Fig. 6. (a) Rate-distortion curves of RDPCO for varying P and C, (b) Corresponding behavior of ID-GAN on MNIST images. The triples on top of each column represent the hyperparameters $(\lambda_d, \lambda_p, \lambda_c)$.

B. RDPCO framework

We now describe experiments using the RDPCO algorithm (Algorithm 2) to solve (23). Since the channel rate in JSCM depends on the latent dimension m and noise levels in $\mathbf{\Sigma}$, we evaluate how these two factors influence the JSCM system. First, we fix m and compute rate-distortion curves under different perception and classification error levels (P and C); we visualize the effect on ID-GAN (Algorithm 1). Second, we investigate how the selection of m influences the RDPC tradeoff for both RDPCO and ID-GAN.

Rate-distortion analysis. In this experiment, we vary both P and C in the constraints of (23). Fig. 6(a) shows rate-distortion curves for different pairs (P, C). 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. 6(b) shows, for different rates, a similar behavior of ID-GAN when we modify its hyperparameters accordingly. Note that the hyperparameters apply only during training; during testing, we inject different levels of channel noise, obtaining different rates. The first column weighs all the metrics equally. The second column imposes a stringer requirement on classification performance by setting λ_c two orders of magnitude larger than λ_d and λ_p [akin to decreasing C in (23)]. 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.

Metrics as a function of D. Fig. 7 shows how distortion, perception, and classification error metrics vary with D in (23). Here the input dimension is $n = 5$ and the latent one is $m = 2$. The metrics displayed in Fig. 7 are different from the ones in Fig. 4, namely the right-hand side of (17) for distortion, (19) for perception, and (21) for classification error. The reason we display these quantities is that RDPCO attempts to minimize them directly. In Figs. 7(a)- 7(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 the RDPC tradeoff. In Fig. 7(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.

Effect of m on the rate-distortion tradeoff. Fig. 8 is similar to Fig. 7, but P and C are fixed, while the latent dimension m varies. When $m = 1$, all metrics are invariant to D. Indeed, in this case (cf. Fig. 8(b)), the perception constraint is active and dominates the two other constraints. For $m = 2, 3$, their classification error behaves similarly, but $m = 3$ achieves better perception and worse distortion.

Fig. 9(a) shows the rate-distortion curves under the same parameters as Fig. 8. We can see again that the rate is invariant to D for $m = 1$, since the perception constraint is the 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. 8(b), the perception constraint is the most stringent of three constraints, and $m = 2$ leads to a better perception-distortion tradeoff. This translates into a better rate-distortion tradeoff in Fig. 9(a).

Finally, Fig. 9(b) shows how m and different rates affect the output images of ID-GAN. Images in the first column 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. These results corroborate the RDPC tradeoff we derived. Indeed, they point to the existence of an optimal

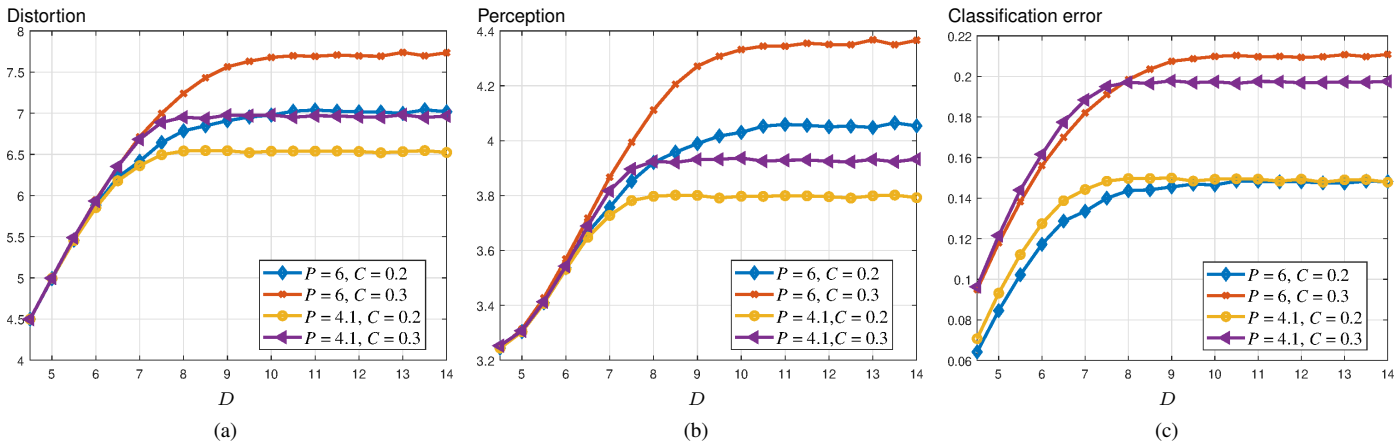


Fig. 7. Values of (a) distortion, (b) perception, and (c) classification error for RDPCO for varying distortion parameter D. These metrics are computed by the right-hand side of the expressions in (17), (19), and (21), respectively.

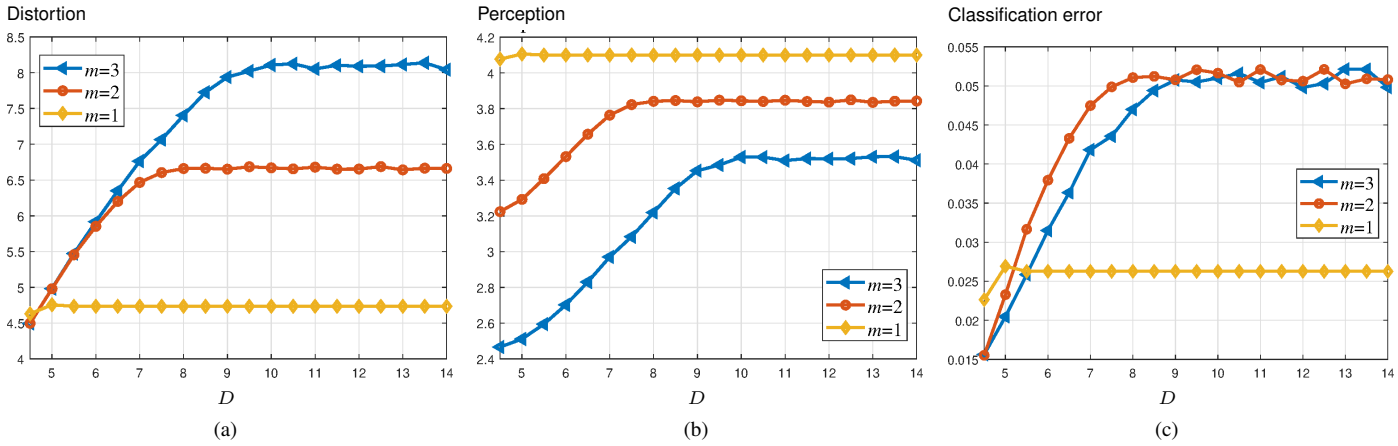


Fig. 8. Values of (a) distortion, (b) perception, and (c) classification error for RDPCO for varying latent dimension m and hyperparameters $(P, C) = (4.1, 0.1)$.

latent dimension m that minimizes the rate while satisfying the distortion, perception, and classification constraints.

VI. 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 based on inverse-domain GAN (ID-GAN) and works under a general scenario; the other algorithm is heuristic and was designed, under a series of simplifying assumptions, to minimize an upper bound on the RDPC function. Experimental results showed the superior performance of ID-GAN in comparison to a traditional method, in which source coding and modulation are designed separately, and to deep joint source-channel coding schemes. Experiments revealed that improving perception quality and classification accuracy require higher rates, and the existence of an optimal compressed/latent dimension that minimizes rate while satisfying constraints on distortion, perception, and classification.

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

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

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

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

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

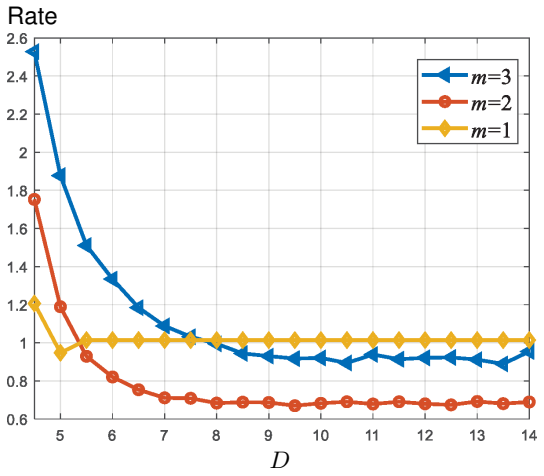
[8] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *ICCV*, 2019, pp. 221–231.

[9] 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.

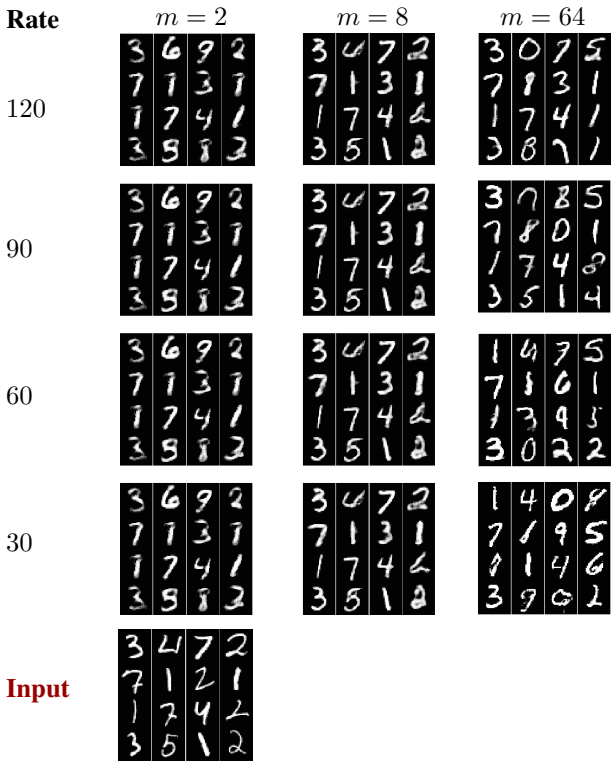
[10] D. B. Kurka and D. Gündüz, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE J. Selected Areas in Inf. Th.*, vol. 1, no. 1, pp. 178–193, 2020.

[11] 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.

[12] J. Xu, B. Ai, N. Wang, and W. Chen, "Deep joint source-channel coding for CSI feedback: An end-to-end approach,"



(a)



(b)

Fig. 9. (a) Rate-distortion curves of RDPCO for varying compressed dimension m; hyperparameters are $(P, C) = (4.1, 0.1)$. (b) corresponding behavior of ID-GAN on MNIST images.

IEEE J. Selected Areas in Communications, vol. 41, no. 1, pp. 260–273, 2023.

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

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

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

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

[30] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *NeurIPS*, 2018, pp. 1–10.

[31] J. Lee, L. Xiao, S. S. Schoenholz, *et al.*, "Wide neural networks of any depth evolve as linear models under gradient descent," in *NeurIPS*, 2019, pp. 1–10.

[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. Wiley, 2001.

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

[36] 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.

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

[38] 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.

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

APPENDIX A
PROOF OF THEOREM 1

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

$$\begin{aligned}&(1-\alpha)R(D_1,P_1,C_1)+\alpha R(D_2,P_2,C_2)\\ &> R\big((1-\alpha)D_1+\alpha D_2,\,(1-\alpha)P_1+\alpha P_2,\,(1-\alpha)C_1+\alpha C_2\big).\end{aligned} \tag{29}$$

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

$$\begin{aligned}\Big(p^{(j)}_{\boldsymbol{Y}|\boldsymbol{X}},\,p^{(j)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}},\,\boldsymbol{\Sigma}^{(j)}\Big) := \underset{p_{\boldsymbol{Y}|\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}},\,\boldsymbol{\Sigma}}{\operatorname{argmin}} \quad & \textstyle\sum_{i=1}^m \log\big(1+\frac{1}{\boldsymbol{\Sigma}_{ii}}\big)\\ \text{s.t.} \quad & \mathbb{E}\big[\Delta(\boldsymbol{X},\widehat{\boldsymbol{X}})\big]\leq D_j\\ & d(p_{\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}})\leq P_j\\ & \epsilon_{c_0}(\boldsymbol{X},\widehat{\boldsymbol{X}})\leq C_j\,,\end{aligned} \tag{30}$$

and denote by $\widehat{\boldsymbol{X}}^{(j)}$ the output of (2) with the parameters computed in (30). Using the strict convexity of the function $x\mapsto\log(1+1/x)$ for $x>0$, the left-hand side of (29) equals

$$\begin{aligned}&(1-\alpha)R(D_1,P_1,C_1)+\alpha R(D_2,P_2,C_2)\\ &=\sum_i\left[(1-\alpha)\log\Big(1+\frac{1}{\boldsymbol{\Sigma}^{(1)}_{ii}}\Big)+\alpha\log\Big(1+\frac{1}{\boldsymbol{\Sigma}^{(2)}_{ii}}\Big)\right]\\ &>\sum_i\log\left(1+\frac{1}{(1-\alpha)\boldsymbol{\Sigma}^{(1)}_{ii}+\alpha\boldsymbol{\Sigma}^{(2)}_{ii}}\right) &(31)\\ &\geq \underset{p_{\boldsymbol{Y}|\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}},\,\boldsymbol{\Sigma}}{\min} \quad \textstyle\sum_{i=1}^m\log\big(1+\frac{1}{\boldsymbol{\Sigma}_{ii}}\big)\\ &\qquad\text{s.t.} \quad \mathbb{E}\big[\Delta(\boldsymbol{X},\widehat{\boldsymbol{X}})\big]\leq(1-\alpha)D_1+\alpha D_2\\ &\qquad\qquad d(p_{\boldsymbol{X}},\,p_{\widehat{\boldsymbol{X}}})\leq(1-\alpha)P_1+\alpha P_2\\ &\qquad\qquad \epsilon_{c_0}(\boldsymbol{X},\widehat{\boldsymbol{X}})\leq(1-\alpha)C_1+\alpha C_2 &(32)\\ &=R\big((1-\alpha)D_1+\alpha D_2,\,(1-\alpha)P_1+\alpha P_2,\\ &\qquad\qquad (1-\alpha)C_1+\alpha C_2\big)\,. &(33)\end{aligned}$$

Step (32) to (33) follows from the definition of $R(D,P,C)$ in (3). The rest of the proof will consist of showing that the step from (31) to (32) holds. Indeed, this will follow if we show that the triple

$$\begin{aligned}\Big((1-\alpha)p^{(1)}_{\boldsymbol{Y}|\boldsymbol{X}}+\alpha p^{(2)}_{\boldsymbol{Y}|\boldsymbol{X}}\,,\;(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}\,,\\ (1-\alpha)\boldsymbol{\Sigma}^{(1)}+\alpha\boldsymbol{\Sigma}^{(2)}\Big)\end{aligned} \tag{34}$$

satisfies the constraints of the optimization problem in (32).

First, notice that (34) 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 (34). Notice that

$$p_{\widehat{\boldsymbol{X}}^{(\alpha)}|\widehat{\boldsymbol{Y}}}=(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}|\widehat{\boldsymbol{Y}}}\,. \tag{35}$$

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

$$p_{\widehat{\boldsymbol{X}}^{(\alpha)}}=(1-\alpha)p^{(1)}_{\widehat{\boldsymbol{X}}}+\alpha p^{(2)}_{\widehat{\boldsymbol{X}}}\,, \tag{36}$$

where $p^{(1)}_{\widehat{\boldsymbol{X}}}$ and $p^{(2)}_{\widehat{\boldsymbol{X}}}$ are the distributions of the output of (2) with the parameters in (30), satisfy the constraints in (32). Indeed, for the first constraint, conditioning on $\widehat{\boldsymbol{Y}}$,

$$\begin{aligned}&\mathbb{E}\big[\Delta(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(\alpha)})\big]\\ &=\mathbb{E}_{\widehat{\boldsymbol{Y}}}\Big[\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(\alpha)}\big)\,|\,\widehat{\boldsymbol{Y}}\big]\Big] &(37)\\ &=\mathbb{E}_{\widehat{\boldsymbol{Y}}}\Big[(1-\alpha)\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(1)}\big)\,|\,\widehat{\boldsymbol{Y}}\big]\\ &\qquad\qquad+\alpha\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(2)}\big)\,|\,\widehat{\boldsymbol{Y}}\big]\Big] &(38)\\ &=(1-\alpha)\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(1)}\big)\big]+\alpha\mathbb{E}\big[\Delta\big(\boldsymbol{X},\widehat{\boldsymbol{X}}^{(2)}\big)\big] &(39)\\ &\leq(1-\alpha)D_1+\alpha D_2\,. &(40)\end{aligned}$$

In (37) and (39), we applied the tower property of expectation. From (37) to (38), we used (35). And from (39) to (40), we used (30). For the second constraint, we use the assumption that $d(\cdot,\cdot)$ is convex in its second argument and, again, (35) and (30):

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

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

$$\begin{aligned}\epsilon_{c_0}(\boldsymbol{X},\,\widehat{\boldsymbol{X}}^{(\alpha)})&=\sum_{i<j}p_j\cdot\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(\alpha)}|H_j} &(41)\\ &=\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(\alpha)}|\widehat{\boldsymbol{Y}},\,H_j}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}} &(42)\\ &=\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(\alpha)}|\widehat{\boldsymbol{Y}}}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}} &(43)\\ &=(1-\alpha)\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(1)}|\widehat{\boldsymbol{Y}}}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}}\\ &\qquad+\alpha\sum_{i<j}p_j\cdot\int\int_{\mathcal{R}_i}\mathrm{d}\,p_{\widehat{\boldsymbol{X}}^{(2)}|\widehat{\boldsymbol{Y}}}\,\mathrm{d}\,p_{\widehat{\boldsymbol{Y}}} &(44)\\ &=(1-\alpha)\epsilon_{c_0}\big(\boldsymbol{X},\,\widehat{\boldsymbol{X}}^{(1)}\big)+\alpha\epsilon_{c_0}\big(\boldsymbol{X},\,\widehat{\boldsymbol{X}}^{(2)}\big) &(45)\\ &\leq(1-\alpha)C_1+\alpha C_2\,. &(46)\end{aligned}$$

From (41) to (42), we conditioned on $\widehat{\boldsymbol{Y}}$. From (42) to (43), we used the Markov property of (2). From (43) to (44), we used (35). From (44) to (45), we applied the same steps as in (5), but in reverse order. And, finally, from (45) to (46), we used (30). □

APPENDIX B
PROOF OF LEMMA 3

Proof. We use the dual form of Wasserstein-1 distance in (6):

$$W_1\big(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}}\big) = \sup_{\|f\|_L \le 1} \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}\big[f(\boldsymbol{X})\big] - \mathbb{E}_{\boldsymbol{X}\sim p_{\widehat{\boldsymbol{X}}}}\big[f(\boldsymbol{X})\big]$$

$$\le \sup_{\|f\|_L \le 1} \Big(\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}\Big[f(\boldsymbol{X}) \,|\, H_0\Big] - \mathbb{E}_{\boldsymbol{X}\sim p_{\widehat{\boldsymbol{X}}}}\Big[f(\boldsymbol{X}) \,|\, H_0\Big]\Big) p_0$$

$$+ \sup_{\|\widehat{f}\|_L \le 1} \Big(\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}\Big[\widehat{f}(\boldsymbol{X}) \,|\, H_1\Big] - \mathbb{E}_{\boldsymbol{X}\sim p_{\widehat{\boldsymbol{X}}}}\Big[\widehat{f}(\boldsymbol{X}) \,|\, H_1\Big]\Big) p_1 \tag{47}$$

$$=: W_1\Big(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}} \,|\, H_0\Big)\cdot p_0 + W_1\Big(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}} \,|\, H_1\Big)\cdot p_1 \tag{48}$$

$$\le W_2\Big(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}} \,|\, H_0\Big)\cdot p_0 + W_2\Big(p_{\boldsymbol{X}},\, p_{\widehat{\boldsymbol{X}}} \,|\, H_1\Big)\cdot p_1 \tag{49}$$

$$= \big\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\big\|_2 p_0 + p_1\sqrt{\big\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\big\|_F^2 + \big\|\boldsymbol{c}_n - \boldsymbol{D}\boldsymbol{E}\boldsymbol{c}_n\big\|_2^2} \tag{50}$$

$$\le \big\|\widehat{\boldsymbol{\Sigma}}^{\frac{1}{2}} - \boldsymbol{I_n}\big\|_F + \big\|\boldsymbol{c}_n - \boldsymbol{D}\boldsymbol{E}\boldsymbol{c_n}\big\|_2 \cdot p_1\,.$$

In (47), 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 (47) to (48), we defined the Wasserstein-1 conditional on an event. From (48) to (49), we used the fact that $W_p(\cdot,\,\cdot) \le W_q(\cdot,\,\cdot)$ whenever $p \le q$; see [39, Remark 6.6]. From (49) to (50), we applied (18) to the models in (11) and (12). And in the last step, we used the triangular inequality. □