Infomax Neural Joint Source-Channel Coding via Adversarial Bit Flip

Yuxuan Song[1], **Minkai Xu**[1], **Lantao Yu**[2], **Hao Zhou**[3], **Shuo Shao**[1], **Yong Yu**[1]
[1] Shanghai Jiao Tong University, [2] Stanford University [3] Bytedance AI lab
{songyuxuan,mkxu,yyu}@apex.sjtu.edu.cn, lantaoyu@cs.stanford.edu,
zhouhao.nlp@bytedance.com, shuoshao@sjtu.edu.cn

Abstract

Although Shannon theory states that it is asymptotically optimal to separate the source and channel coding as two independent processes, in many practical communication scenarios this decomposition is limited by the finite bit-length and computational power for decoding. Recently, neural joint source-channel coding (NECST) (Choi et al. 2018) is proposed to sidestep this problem. While it leverages the advancements of amortized inference and deep learning (Kingma and Welling 2013; Grover and Ermon 2018) to improve the encoding and decoding process, it still cannot always achieve compelling results in terms of compression and error correction performance due to the limited robustness of its learned coding networks. In this paper, motivated by the inherent connections between neural joint source-channel coding and discrete representation learning, we propose a novel regularization method called Infomax Adversarial-Bit-Flip (IABF) to improve the stability and robustness of the neural joint source-channel coding scheme. More specifically, on the encoder side, we propose to explicitly maximize the mutual information between the codeword and data; while on the decoder side, the amortized reconstruction is regularized within an adversarial framework. Extensive experiments conducted on various real-world datasets evidence that our IABF can achieve state-of-the-art performances on both compression and error correction benchmarks and outperform the baselines by a significant margin.

Introduction

Shannon source-channel separation theorem (Shannon 1948) is one of the most fundamental theorems in information theory. It states that when the source and channel coding procedures are conducted separately there is an asymptotically negligible loss in the point-to-point communication system, which indicates the system can be simplified as a series of two subsystems without any interference to each other. However, although the separation approach is optimal in many scenarios (Tian et al. 2013), it can be proved to be sub-optimal when the source generates finite-length data blocks (Kostina and Verdú 2013), which is common in practical communication scenarios. Furthermore, in this case, maximum likelihood decoding was proven to be an NP-hard problem (Berlekamp, McEliece, and Van Tilborg 1978), and relaxation approaches of this problem (Feldman, Wainwright, and Karger 2005; Vontobel and Koetter 2007) also suffer from high distortion rate issues.

To address the above-mentioned challenges, the machine learning community has cast the encoding process of joint source-channel coding as a binary representation learning problem. Recently, inspired by variational mutual information maximization, (Choi et al. 2018) proposed Neural Error Correcting and Source Trimming (NECST), which is a deep learning framework for joint source-channel coding. To be more specific, the data is encoded with a network into a bit-string representation without the need of manually designing the coding scheme. Additionally, based on amortized optimization, the framework provides an extremely fast decoder after training.

However, while NECST leverages the advancements of amortized inference and deep learning to improve the compression and reconstruction process, it still cannot always achieve compelling results when concerning about compression and error correction performance, which is mainly due to the limited informativeness and robustness of learned coding networks. In this paper, we propose a novel approach to solve the previous problems in NECST via mutual information maximization and amortized regularization (Shu et al. 2018). On the encoder side, we employ the idea of mutual information maximization for the better informativeness, which is predominant in representation learning (Hu et al. 2017; Zhao, Song, and Ermon 2017; Rolfe 2016). However, we note that the previous mutual information maximization methods in neural discrete representation learning are intractable when the dimension (*i.e.* length of bit-string) is high. To address this issue, we further propose a novel loss function which can be estimated and optimized with deep neural networks more effectively. On the decoder side, we theoretically demonstrate that the objective in (Choi et al. 2018) is essentially regularizing the amortized decoder and the strength of regularization is determined by the noise level of the channel. The amortized optimization tends to encourage the decoder to capture more global structures during the decoding, and it is reasonable to be suitably regularized. Based on this intuition, we further proposed "Adversarial Bits Flip", a new regularization mechanism for discrete input neural network to improve the robustness of coding scheme.

More specifically, our contributions can be summarized as follows:

- We propose a novel mutual information maximization framework, which is scalable for high-dimensional discrete representation learning;
- Theoretical analysis of the learning paradigm in neural joint source-channel coding is conducted. Based on the theoretical understanding, we propose a novel virtual adversarial regularization method to improve the robustness of the discrete-input decoder networks;
- We conduct extensive experiments on various benchmark datasets and different tasks. Empirical evidence demonstrates the effectiveness of our methods on both reducing the distortion with finite bit-length and learning useful representation for downstream tasks.

Related Work

Generative models have been regarded as the backbones of both lossless and lossy compression methods. Recently, there are a surge of studies on applying deep generative models to lossy compression. (Ballé et al. 2018) proposed to substitute the fixed prior in Variational Auto-encoder (VAE) with a learnable scale hyper-prior such that the structure information can be better presented for effective compression. (Theis et al. 2017) utilized autoencoding framework to obtain the optimal number of bits for compressing images. Likelihood-free methods based on adversarial learning are also proposed to learn neural codes for better compression (Santurkar, Budden, and Shavit 2018).

The robustness of the learned coding scheme is another important topic under the communication setting. Here we discuss two recent progress in the aspect of the involvement of channel noise. (Grover and Ermon 2018) proposed Uncertainty Autoencoder (UAE) to learn the compressed representation of the original inputs, and (Choi et al. 2018) can be seen as a discrete version of UAE, *i.e.*, both the codewords and the noise model is no longer continuous. Both (Grover and Ermon 2018) and (Choi et al. 2018) maximize the mutual information within a variational framework, while in our work, we tend to enhance the mutual information between the codewords and the input data without introducing any variational approximation. In the literature of information-theoretic approach for representation learning, (Chen et al. 2016; Hjelm et al. 2018; van den Oord, Vinyals, and others 2017; Zhao, Song, and Ermon 2018) also proposed to utilize information maximization to improve representation learning. However, in their settings, the discrete latent noise is not involved. Specifically, (Hu et al. 2017) introduced a component for regularizing the encoder function with Virtual Adversarial Training (VAT) (Miyato et al. 2015). Although both inspired by VAT, our method instead regularizes the decoder function. The underlying motivations are also different: (Hu et al. 2017) tend to impose intended invariance on discrete representations, while our method aims to enhance the robustness of coding scheme by stimulating and improving the worst-case performance for some channel noise level.

Background and Notations

Joint Source and Channel Coding

To begin with, we firstly formulate the problem of communicating data across a noisy channel. Following the notations in (Choi et al. 2018), we denote the input space as $\mathcal{X} \subseteq \mathbb{R}^n$, and the source distribution on the input space as $p_{\text{data}}(\boldsymbol{x})$. The communication system encodes a block of data $\boldsymbol{x}_{1:n}$ *i.i.d.* drawn from $p_{\text{data}}(\boldsymbol{x})$ into codewords. Specifically, for each data instance $\boldsymbol{x}$, the corresponding codeword $\boldsymbol{y}$ is a binary code of length m, *i.e.*, $\boldsymbol{y}_{1:m} \in \mathcal{Y} = \{0,1\}^m$. The codeword will then be transmitted through a noisy channel, which results in a *corrupted* codeword $\hat{\boldsymbol{y}}_{1:m} \in \mathcal{Y}$. After receiving the noisy codes, the decoder will produce a reconstructed version $\hat{\boldsymbol{x}} \in \mathcal{X}$. The ultimate goal is to minimize the overall distortion, *i.e.*, minimizing the ℓ_p norm $\|x - \hat{x}\|_p$(typically $p = 1$ or 2).

Source encoder tries to compress source message into a bit-string with as less number of bits as possible. While channel encoder tends to re-introduces redundancies for error correction after the transmission through noisy channel. Shannon proved that the above scheme is optimal for infinitely long messages in the separation theorem (Shannon 1948). While in practice, when the bit-length is finite, a joint source-channel coding algorithm can have a better performance than the separated source-channel coding (Pilc 1967).

Autoencoder and Variational Information Maximization

Autoencoder (Ballard 1987) consists of a pair of parameterized functions, an encoder (f_θ) and a decoder (f_ϕ). The encoder maps sample $\boldsymbol{x}$ from the n-dimensional data space $\mathcal{X}$ to a codeword $\boldsymbol{y}$ in the m-dimensional latent space $\mathcal{Y}$, and the decoder defines a function from the latent space to the data space. Typically, an autoencoder seeks to minimize the l_2 reconstruction error over a dataset $\mathcal{D}$:

$$\min_{f_\theta, f_\phi} \sum_{x \in \mathcal{D}} \|x - f_\phi(f_\theta(x))\|_2^2. \tag{1}$$

Usually both f_ϕ and f_θ are parameterized with neural networks. When probabilistic encoder and decoder are utilized, the encoding part and the decoding part essentially imply corresponding conditional distributions, *i.e.* $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and $q_\phi(\boldsymbol{x}|\boldsymbol{y})$. And the corresponding joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{y})$ between the two random variables is also defined by the factorization $p_\theta(\boldsymbol{x}, \boldsymbol{y}) = p_\theta(\boldsymbol{y}|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x})$. When the objective is to obtain informative measurements to reconstruct the original signal effectively, it is reasonable to maximize the mutual information $I_\theta(X, Y)$ between these two random variables X and Y:

$$\begin{aligned} \max_\theta I_\theta(X, Y) &= \int p_\theta(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{y})}{p_{\text{data}}(\boldsymbol{x}) p_\theta(\boldsymbol{y})} \mathrm{d}x \mathrm{d}y \\ &= H(X) - H_\theta(X|Y) \end{aligned} \tag{2}$$

Here H stands for differential entropy. Estimating and optimizing the mutual information between high-dimensional random variables is intractable. However, the mutual information can actually be lower bounded by introducing a variational approximation of the posterior $p_\theta(\boldsymbol{x}|\boldsymbol{y})$. With $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ representing the variational distribution, the lower bound can be written as:

$$H(X) + \mathbb{E}_{p_\theta(\boldsymbol{x},\boldsymbol{y})}[\log q_\phi(\boldsymbol{x}|\boldsymbol{y})] \leq I_\theta(X,Y) \tag{3}$$

The bound is tight when the variational distribution $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ matches the true posterior $p_\theta(\boldsymbol{x}|\boldsymbol{y})$. Since $H(X)$ is the entropy of data distribution, it can be treated as a constant during the optimization of θ and ϕ. The final objective of the stochastic optimization can be concluded as (Grover and Ermon 2018):

$$\max_{\theta,\phi} \mathbb{E}_{p_\theta(\boldsymbol{x},\boldsymbol{y})} \left[\log q_\phi(\boldsymbol{x}|\boldsymbol{y})\right] \tag{4}$$

Neural Error Correcting and Source Trimming Codes

Let X,Y,$\hat{Y}$,$\hat{X}$ the random variables for the inputs, codewords, noisy codewords after channel corruption, and the reconstructed data by decoder. (Choi et al. 2018) modeled a coding process with the following graphical model $X \rightarrow Y \rightarrow \hat{Y} \rightarrow \hat{X}$:

$$\begin{aligned} &p(\boldsymbol{x},\boldsymbol{y},\hat{\boldsymbol{y}},\hat{\boldsymbol{x}}) = \\ &p_{\text{data}}(\boldsymbol{x})p_\theta(\boldsymbol{y}|\boldsymbol{x})p_{\text{channel}}(\hat{\boldsymbol{y}}|\boldsymbol{y};\epsilon)q_\phi(\hat{\boldsymbol{x}}|\hat{\boldsymbol{y}}) \end{aligned} \tag{5}$$

Here $p_{\text{channel}}(\hat{\boldsymbol{y}}|\boldsymbol{y};\epsilon)$ denotes the probabilistic model of the noisy channel, *i.e.*, flipping each bit with probability ϵ. Inspired by recent advancements of latent-variable generative models, (Choi et al. 2018) proposed a variational information maximization method based on Eq. 4 to improve the information theoretic dependency between the input X and the noisy codeword $\hat{Y}$ in joint source-channel coding.

Methodology

Motivation

Optimizing the variational bound as proposed in (Choi et al. 2018) can approximately maximize the corresponding mutual information between the data and noisy codewords. However, it should be noticed that previous variational approximation based method is mainly limited by the capacity of the parameterized variational distribution family (Kingma and Welling 2013) and high-variance gradient estimation. In this paper, we take an alternative perspective: the maximization of $I(X,\hat{Y})$ can be decomposed into maximizing $I(X,Y)$ and minimizing the information loss during noisy channel corruption. Based on the above decomposition, we propose a novel method to sidestep the limitations of variational approximation and further improve the error correction ability. To be more specific, $I(X,Y)$ is maximized without involving a parameterized variational distribution and the robustness of a coding scheme is strengthened in an adversarial fashion for reducing information loss.

In the following, we first introduce our method on how to directly impose mutual information maximization without involving variational distribution in joint source-channel coding. Then we conduct a theoretical study on the learning paradigm of the NECST where we demonstrate that injecting latent noise essentially smooths the amortized decoder function. Based on the analysis, we propose a regularization method "Adversarial Bit Flip", which can effectively regularize the decoder by virtually attacking the vulnerable bits during training, which further improve the robustness of the learned coding scheme.

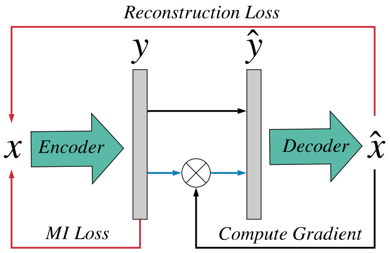


Figure 1: Overview of our proposed IABF algorithm. Green: encoder and decoder networks. $\boldsymbol{x}$ denotes the input data and $\hat{\boldsymbol{x}}$ denotes the reconstructed data from coding. $\boldsymbol{y}$ denotes the codeword before the noisy channel and $\hat{\boldsymbol{y}}$ denotes the noisy codeword. Training procedure: First (black arrows), we encode the input data $\boldsymbol{x}$ to $\boldsymbol{y}$, then we directly decode the clean code $\boldsymbol{y}$ and compute the reconstruction loss. According to the reconstruction loss, we compute the gradients of different bits. Secondly (blue arrows), we flip the bits according to the gradient norm and obtain the noisy code $\hat{\boldsymbol{y}}$. Finally (red arrow), we compute the reconstruction loss between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ and the mutual information loss between $\boldsymbol{x}$ and $\boldsymbol{y}$, and then update the parameters of encoder and decoder networks according to the two loss components.

Information Maximization with Discrete Representation

The sample space of M-bit codeword can be denoted as $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_\mathcal{M}$, where $\mathcal{Y}_m \equiv \{0,1\}$ and Y_m stands for the random variable for the m-th bit in codewords. We seek to learn a probabilistic encoder $p_\theta(\boldsymbol{y}_1,\cdots,\boldsymbol{y}_M|\boldsymbol{x})$ which implies the optimal coding strategy. Following previous works (Kingma and Welling 2013; Grover and Ermon 2018), we also leverage the mean-field assumption when modeling the conditional distribution with neural networks, *i.e.*

$$p_\theta\left(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_M|\boldsymbol{x}\right) = \prod_{d=1}^{M} p_\theta\left(\boldsymbol{y}_d|\boldsymbol{x}\right) \tag{6}$$

The mutual information $I(Y_1,\cdots,Y_M;X)$ between the codewords and inputs is intractable for large bits number M, as the summation over an exponential number of terms is involved. Following (Brown 2009), we derive the mutual information between a set of codewords and inputs as the following sum of Interaction Information terms:

$$I\left(Y_{1:M};X\right) = \sum_{T\subseteq S,|T|\geq 1} I(T\cup Y) \tag{7}$$

where $S \equiv \{Y_1,\cdots,Y_M\}$. With similar approximation by truncating all the higher order terms with $|T| > 2$ (Hu et al.

2017; Brown 2009; Erin Liong et al. 2015), the new approximated target can be illustrated as:

$$\sum_{d=1}^{M} I\left(Y_d; X\right) + \sum_{1 \leq d \neq d' \leq M} I\left(\{Y_d, Y_{d'}, X\}\right) \tag{8}$$

Note that under the mean-field assumption in Eq. 6, the pairwise interaction information can be derived as:

$$\begin{aligned} I\left(\{Y_d, Y_{d'}, X\}\right) &\equiv I\left(Y_d; Y_{d'}|X\right) - I\left(Y_d; Y_{d'}\right) \\ &= -I\left(Y_d; Y_{d'}\right) \end{aligned} \tag{9}$$

Substituting Eq. 8 into Eq. 9, we get the approximated mutual information maximization term:

$$I\left(Y_{1:M}; X\right) = \sum_{d=1}^{M} I\left(X; Y_d\right) - \sum_{1 \leq d \neq d' \leq M} I\left(Y_d; Y_{d'}\right) \tag{10}$$

The final objective consists of two terms. The first term $\sum_{d=1}^{M} I\left(X; Y_d\right) = \sum_{d=1}^{M} (H(Y_d) - H(Y_d|X))$ corresponds to maximizing the the summation of mutual information between data and each bit of codewords, which can be easily derived under the situation of binary representation:

$$\begin{aligned} H(Y_d) &\equiv h\left(p_\theta(\boldsymbol{y}_d)\right) = H\left(\frac{1}{N}\sum_{i=1}^{N} p_\theta(\boldsymbol{y}_d|\boldsymbol{x}^{(i)})\right) \\ H(Y_d|X) &\equiv \frac{1}{N}\sum_{i=1}^{N} H\left(p_\theta\left(\boldsymbol{y}_d|\boldsymbol{x}^{(i)}\right)\right) \end{aligned} \tag{11}$$

where $H(\cdot)$ denotes the entropy of a discrete distribution and N stands for the number of samples. However, the second term in Eq. 10 corresponds to the orthogonality constraint which dedicates to removing the redundancy between different dimension of codewords. To impose the orthogonality constraint on different dimensions of codewords, instead of directly optimizing the intractable target in Eq. 10, we introduce Total Correlation as an independence measure of multi-dimensional random variable:

$$TC(Y_{1:M}) = D_{\mathrm{KL}}\left(p_\theta(\boldsymbol{y}_{1:M}) \| \prod_{j=1}^{M} p_\theta\left(\boldsymbol{y}_j\right)\right) \tag{12}$$

Samples from marginal distribution p_θ can be easily obtained through the following ancestral sampling procedure: $\boldsymbol{x} \sim p(\boldsymbol{x}), \boldsymbol{y}_{1:M} \sim p_\theta(\boldsymbol{y}_{1:M}|\boldsymbol{x})$. And sampling for distribution $\prod_{j=1}^{M} p_\theta\left(\boldsymbol{y}_j\right)$ can be implemented through randomly permuting across a batch of samples from $p_\theta(\boldsymbol{y}_{1:M})$ for each dimension. To optimize $TC(Y_{1:M})$, we leverage the density ratio estimation trick as illustrated in (Kim and Mnih 2018). We mix up the samples from $p_\theta(\boldsymbol{y}_{1:M})$ and $\prod_{j=1}^{M} p_\theta\left(\boldsymbol{y}_j\right)$, and train a MLP classifier to output the probability C_ψ of a codeword coming from $p_\theta(\boldsymbol{y}_{1:M})$ with the following objective:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{y}_{1:M} \sim p_\theta(\boldsymbol{y}_{1:M})}[\log C_\psi(\boldsymbol{y}_{1:M})] + \\ &\mathbb{E}_{\boldsymbol{y}_{1:M} \sim \prod_{j=1}^{M} p_\theta(\boldsymbol{y}_j)}[\log(1 - C_\psi(\boldsymbol{y}_{1:M}))] \end{aligned} \tag{13}$$

And the total correlation can be approximated as (Goodfellow 2014):

$$\begin{aligned} TC(\boldsymbol{y}_{1:M}) &= \mathbb{E}_{p_\theta(\boldsymbol{y}_{1:M})}\left[\log \frac{p_\theta(\boldsymbol{y}_{1:M})}{\prod_{j=1}^{M} p_\theta\left(\boldsymbol{y}_j\right)}\right] \\ &\approx \mathbb{E}_{p_\theta(\boldsymbol{y}_{1:M})}\left[\log \frac{C_\psi(\boldsymbol{y}_{1:M})}{1 - C_\psi(\boldsymbol{y}_{1:M})}\right] = T\hat{C}(\boldsymbol{y}_{1:M}) \end{aligned} \tag{14}$$

Amortized Decoder Regularization with Adversarial Bit Flip

Given variational family $\mathcal{Q}$ and the joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{y})$ implicitly defined by the encoder function $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and $p_{\text{data}}(\boldsymbol{x})$, the variational mutual information maximization can be formalized as:

$$\begin{aligned} \max_\theta \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})p_\theta(\boldsymbol{y}|\boldsymbol{x})}[H(p_\theta(\cdot|\boldsymbol{y})) - \\ \min_{q \in \mathcal{Q}} D_{\mathrm{KL}}\left(p_\theta(\boldsymbol{x}|\boldsymbol{y}) \| q(\boldsymbol{x})\right))] \end{aligned} \tag{15}$$

where $D_{\mathrm{KL}}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence. It can be shown that variational bound approximates the original objective best when we use the best decoding distribution $q^*_{\boldsymbol{y}}(x)$ for each $\boldsymbol{y} \sim p_\theta(\boldsymbol{y})$. The amortized optimization (Choi et al. 2018) turns the individual optimization procedure for finding the optimal decoding distribution for each codeword $\boldsymbol{y}$ into a single regression problem by using a recognition model $f_\phi : \mathcal{Y} \rightarrow \mathcal{Q}$ to predict $q^*_{\boldsymbol{y}}(x)$. And the function f_ϕ can be concisely represented as the conditional distribution $q_\phi(x|\boldsymbol{y})$ which results in the form of objective in Eq. 4. The amortization is appealing in neural joint source-channel coding as the amortized decoder can very efficiently map the transmitted code into its best reconstruction at test time. However, amortization with the neural network as the variational function family is over-expressive in many cases and prone to overfitting (Shu et al. 2018), while the objective of joint source-channel coding is to find the coding scheme which can generalize to the unseen test data and achieve the desired compression performance. Hence regularizing the capacity of amortization family can be seen as the way to refine the decoder module to match the expected desiderata.

NECST as an Amortized Decoder Regularization In NECST, the final objective is to maximize the variational bound of mutual information between the corrupted codewords and the input data:

$$\max_{\theta, \phi} \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{y} \sim p_{\text{noisy}}(\boldsymbol{y}|\boldsymbol{x}; \epsilon, \theta)} \left[\log q_\phi(\boldsymbol{x}|\boldsymbol{y})\right] \tag{16}$$

where

$$p_{\text{noisy}}(\boldsymbol{y}|\boldsymbol{x}; \epsilon, \theta) = \sum_{\hat{\boldsymbol{y}} \in \hat{\mathcal{Y}}} p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}) p_{\text{channel}}(\boldsymbol{y}|\hat{\boldsymbol{y}}; \epsilon) \tag{17}$$

is defined according to a different noise channel. Following the typical setting of joint source-channel coding, we briefly introduce two widely used discrete channel models: (1) the binary erasure channel (BEC); and (2) the binary symmetric channel (BSC). In BEC, each bit may be i.i.d. erased into an unrecognized symbol with some probability ϵ, and the uncorrupted bits will be transmitted faithfully. In BSC, each bit will be flipped independently with probability ϵ, (*e.g.* $0 \to 1$). It is noted that BSC is widely recognized as a more difficult communication channel than BEC (Richardson and Urbanke 2008). Hence in the following sections and experiments, we mainly focus on the scenario of BSC. And BEC can be easily adapted from the discussion on BSC. In the BSC channel coding, the whole encoding distribution can be formulated as following (Choi et al. 2018):

$$\begin{aligned} & p_{\text{noisy}}(\boldsymbol{y}|\boldsymbol{x};\theta,\epsilon) \\ = & \prod_{i=1}^{m} \left(\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right) - 2\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right)\epsilon + \epsilon\right)^{\boldsymbol{y}_i} \cdot \\ & \left(1 - \sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right) + 2\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right)\epsilon - \epsilon\right)^{(1-\boldsymbol{y}_i)} \end{aligned} \tag{18}$$

where each bit $\boldsymbol{y}_i$ is modeled as an independent Bernoulli distribution, the parameter of the Bernoulli distribution is modeled through a neural network f_θ, and σ denotes the Sigmoid function. While from the perspective of the decoder module, the injected noise defines a new learning objective for amortized learning:

$$\begin{aligned} \max_{\theta} \min_{\phi} \; & I_\theta(X,Y) - \\ & \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}) p_{\text{noisy}}(\boldsymbol{y}|\boldsymbol{x};\theta,\epsilon)}[D_{\text{KL}}(p_\theta(\boldsymbol{x}|\boldsymbol{y}) \| f_\phi(\boldsymbol{y}))] \end{aligned} \tag{19}$$

We then show that the optimal amortized decoder with the channel noise is the following kernel function. And the noise level ϵ actually determine the smoothness of the optimal decoder.

Definition 1 *The univariate kernel of two discrete random variable is defined as following:*

$$K_1^{(U)}\left(y_i, y_j\right) = \begin{cases} P_Y\left(y_i\right) & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases} \tag{20}$$

where P_Y is the probability mass function of a discrete random variable Y. And the corresponding multi-variate kernel is:

$$K\left(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}\right) = \frac{1}{m} \sum_{k=1}^{m} k_1^{(U)}\left(\boldsymbol{y}_k^{(i)}, \boldsymbol{y}_k^{(j)}\right), \tag{21}$$

In our case the perturbed distribution $P_Y = \sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right) - 2\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right)\epsilon + \epsilon$ for $Y = 1$ and $P_Y = 1 - \sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right) + 2\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right)\epsilon - \epsilon$ for $Y = 0$. We refer to the corresponding kernel with noise level ϵ as K_ϵ.

Theorem 1 *The optimal amortized decoder given a fixed encoder in Eq. 19 is a kernel regression model, which can be illustrated as:*

$$f^*_{\epsilon,\phi}(\boldsymbol{y}) = \underset{f_\phi \in \mathcal{F}(q)}{\arg\min} \sum_{i=1}^{n} w_\epsilon\left(\boldsymbol{y}, \boldsymbol{y}^{(i)}\right) \cdot D_{\text{KL}}\left(p_\theta\left(\boldsymbol{x}|\boldsymbol{y}^{(i)}\right) \| f_\phi(\boldsymbol{y})\right)$$

where $w_\epsilon\left(\boldsymbol{y}, \boldsymbol{y}^{(i)}\right) = \frac{K_\epsilon\left(\boldsymbol{y}, \boldsymbol{y}^{(i)}\right)}{\sum_j K_\epsilon\left(\boldsymbol{y}, \boldsymbol{y}^{(j)}\right)}$; K_ϵ denote the kernel function defined in Definition 1 with ϵ as the noise level; n denotes the number of training samples.

Algorithm 1 Infomax Adversarial Bits Flip

```
Input: Dataset($\mathcal{X}$) to be compressed. Channel noise level $\epsilon$.
Initialize the parameters of encoder $p_\theta(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M|\boldsymbol{x})$, the parameter of decoder $q_\phi(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M)$ and a Classifier $C_\psi$.
repeat
    Sample a batch of samples from Dataset: $\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})$
    Sample the corresponding latent codes $\boldsymbol{y} \sim p_\theta(\boldsymbol{y}|\boldsymbol{x})$.
    Permute $\boldsymbol{y}$ across the batch of samples for each dimension to get $\boldsymbol{y}^*$.
    Update $\psi$ according to the classification loss:
        $\log C_\psi(\boldsymbol{y}) + \log(1 - C_\psi(\boldsymbol{y}^*))$
    Update $\theta$, $\phi$ according to the objective in Eq. 29:
        $\mathcal{L}_{rec}(\phi, \theta; \boldsymbol{x}, \epsilon) + \lambda \mathcal{L}_{info}(\theta; \boldsymbol{x})$
until Convergence
Output: Learned joint source-channel coding scheme $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and amortized decoder $q_\phi(\boldsymbol{x}|\boldsymbol{y})$
```

Theorem. 1 states that the optimal amortized decoder is related to the noise level ϵ. And the optimal $f^*_{\epsilon,\phi}(\boldsymbol{y})$ corresponds to the decoding procedure which minimizes the weighted Kullback-Leibler(KL) divergence from $f^*_{\epsilon,\phi}(\boldsymbol{y})$ to each $p_\theta(\boldsymbol{x}|\boldsymbol{y})$. The weighting function $w_\epsilon\left(\boldsymbol{y}, \boldsymbol{y}^{(i)}\right)$ depends on the Bernoulli parameter of decoder output $\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right)$ and the noise level ϵ. It can be found that the noise level ϵ forces the decoder to get similar reconstruction outputs with similar codewords under a predefined similarity measure. When the channel noise reach the highest level ,*i.e.* $\epsilon = 0.5$, the weighted function will then turn into constant $w_\epsilon\left(\boldsymbol{y}, \boldsymbol{y}^{(i)}\right) = \frac{1}{n}$. Intuitively, the noise level is related to the smoothness of the decoder function: how much a single decoding procedure will be influenced by the other decoding procedures.

Formally, we have the following proposition:

Proposition 1 *With a regularized amortization objective as:*

$$\begin{aligned} & R(\phi;\epsilon) = \\ & \min_{f_\phi \in \mathcal{F}(q)} \mathbb{E}_{p(\boldsymbol{x}) p_{noisy}(\boldsymbol{y}|\boldsymbol{x};\theta,\epsilon)}[D_{\text{KL}}(p_\theta(\boldsymbol{x}|\boldsymbol{y}) \| f_\phi(\boldsymbol{y}))] \end{aligned} \tag{22}$$

The following condition is satisfied if the input is closed under the permutation: if $f_\phi(\boldsymbol{y}) \in \mathcal{F}(q)$ then the noised version $f_\phi(\hat{\boldsymbol{y}};\epsilon) \in \mathcal{F}(q)$, where $f_\phi(\hat{y};\epsilon)$ denotes perturb the input $\hat{\boldsymbol{y}}$ with noise channel $p_{channel}(\hat{\boldsymbol{y}}|\boldsymbol{y};\epsilon)$. Then it is satisfied that when $0 \leq \epsilon_1 < \epsilon_2 < 0.5$, $R(\phi;\epsilon_1) \leq R(\phi;\epsilon_2)$ for all $\phi \in \Phi$.

Theorem. 1 and Proposition. 1 show that with a larger noise level, the decoder is further regularized and forced to be smoother. And the objective with smaller ϵ in Eq. 19 is necessarily bounded by the objective with larger ϵ. So the regularized objectives are valid lower bounds of the original variational bound of mutual information in Eq. 4, where the ϵ can be seen as 0.

Adversarial Bit Flip Above theoretical analysis helps us build connections between neural joint source-channel coding and regularized amortized optimization. Hence there is strong motivation to design effective regularization strategy for the amortized decoder and improve the stability and generalization performance of the coding scheme. In the continuous setting, the intended regularization of neural networks can be imposed through local perturbations (Miyato et al. 2015; Bachman, Alsharif, and Precup 2014), where the neural network is encouraged to be invariant towards the perturbation. More specifically, in Virtual Adversarial Training (VAT) (Miyato et al. 2015), the perturbation is selected as the adversarial direction based on first-order gradient. However, in our scenario, we tend to inject noise on the binary codewords which makes the gradient-based perturbation strategy no longer applicable. Inspired by VAT, we introduce the following "adversarial bit flip" procedure. Given a M-bit codeword $\boldsymbol{y} \sim p_\theta(\boldsymbol{y}|\boldsymbol{x})$, we first calculate the gradient with respect to the reconstruct loss $\mathcal{L} = \log q_\phi(\boldsymbol{x}|\boldsymbol{y})$:

$$\nabla_{\boldsymbol{y}} \mathcal{L} = \left[\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_1}, \ldots, \frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_M} \right] \tag{23}$$

The naive perturbation is to get the corrupted bits $\hat{\boldsymbol{y}}$ by:

$$\hat{\boldsymbol{y}} = \boldsymbol{y} + \operatorname{sign}\left(\nabla_{\boldsymbol{y}} \mathcal{L}\right) \tag{24}$$

where the $\operatorname{sign}\left(\nabla_{\boldsymbol{y}} \mathcal{L}\right) \in \{-1, +1\}^M$. Since the codewords is constrained to be binary, perturbation following Eq. 24 may output numbers not in $\{0, 1\}$. Note that the bit flip should only happen in two situations, when the original bits is 0 and the corresponding gradient is positive or the original bit is 1 and the corresponding gradient is negative. Then we refine the perturbation procedure as following:

$$\boldsymbol{m} = \boldsymbol{y} \oplus \left(\operatorname{sign}\left(\nabla_{\boldsymbol{y}} \mathcal{L}\right)/2 + 0.5\right) \tag{25}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{y} \oplus \boldsymbol{m} \tag{26}$$

where $\oplus$ denotes the XOR operation. The key motivation of above procedure lies in that we leverage the gradient information as guidance to find the most *vulnerable* bits and virtually attack these bits, which will implicitly force the information uniformly-distributed along different dimensions. From another perspective, the Adversarial Bit Flip can be seen as simulating the worst possible case given noise level fixed, which can provide more informative signal to improve the robustness. The whole training procedure is summarized in Algorithm 1.

Experiments

In this section, we firstly introduce the implementation and optimization in details. Then following the evaluation framework in (Choi et al. 2018), results on several benchmark datasets: BinaryMNIST, MNIST (LeCun 1998), Omniglot (Lake, Salakhutdinov, and Tenenbaum 2015) and CIFAR10 (Krizhevsky, Hinton, and others 2009) are provided to demonstrate the effectiveness of our methods on compression and error correction.[1].

Objective and Optimization Procedure

With the definition in Eq. 11 and Eq. 14, the mutual information maximization term for increasing $I(X, Y)$ is:

$$\sum_{d=1}^{M} (H(Y_d) - H(Y_d|X)) - \hat{TC}(Y_{1:M}) \equiv \mathcal{L}_{info}(\theta; \boldsymbol{x}) \tag{27}$$

And the adversarial bits flip term can be illustrated as:

$$\sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{E}_{\boldsymbol{y} \sim p_{adv}(\boldsymbol{y}|\boldsymbol{x};\epsilon,\theta)} \left[\log q_\phi(\boldsymbol{x}|\boldsymbol{y})\right] \equiv \mathcal{L}_{rec}(\phi, \theta; \boldsymbol{x}, \epsilon) \tag{28}$$

Combining the above two components, we derive the final objective of our Infomax Adversarial Bits Flip(IABF) as following:

$$\max_{\theta,\phi} \mathcal{L}_{rec}(\phi, \theta; \boldsymbol{x}, \epsilon) + \lambda \mathcal{L}_{info}(\theta; \boldsymbol{x}) \tag{29}$$

where λ is the only hyperparameter and selected from a small candidate set $\{0.1, 0.01, 0.001\}$ during the experiments. The perturbed distribution implied by adversarial bit flip $p_{adv}(\boldsymbol{y}|\boldsymbol{x}; \epsilon, \theta)$ is approximated with a continuous relaxation of Eq. 26:

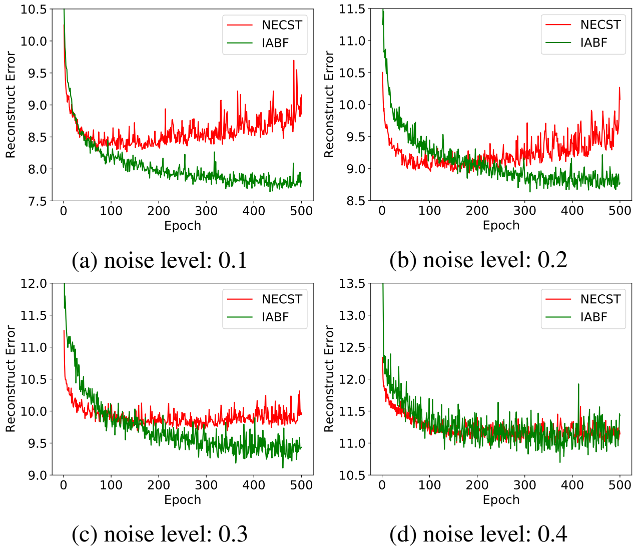
$$\begin{aligned} & p_{adv}(\boldsymbol{y}|\boldsymbol{x}; \theta, \epsilon) \\ = & \prod_{i=1}^{m} \left(\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right) - 2\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right)\epsilon_i^* + \epsilon_i^{\cdot}\right)^{\boldsymbol{y}_i} \cdot \\ & \left(1 - \sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right) + 2\sigma\left(f_\theta\left(\boldsymbol{x}\right)_i\right)\epsilon_i^* - \epsilon_i^*\right)^{(1-\boldsymbol{y}_i)} \end{aligned} \tag{30}$$

Instead of directly applying the same ϵ level noise on all the dimensions, here the noise level on i-dimension $\epsilon_i \propto \|\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i}\|$ satisfying that $\sum_{d=1}^{M} \epsilon_i = M\epsilon, 0 \leq \epsilon_i \leq 1$. And ϵ_i^* indicates the modified noise level with with the mask code $\boldsymbol{m}$ as shown in Eq. 25, *i.e* $\epsilon_i^* = \epsilon_i \cdot m_i$.

Binary MNIST	0.1	0.2	0.3	0.4
100-bit NECST	0.116	0.150	0.193	0.249
100-bit ABF	0.114	0.149	0.191	0.244
100-bit IABF	**0.113**	**0.147**	**0.189**	**0.241**
Omniglot	0.1	0.2	0.3	0.4
200-bit NECST	24.693	31.538	39.657	47.888
200-bit ABF	24.192	31.430	39.039	47.827
200-bit IABF	**24.132**	**31.373**	**38.936**	**47.608**
CIFAR10	0.1	0.2	0.3	0.4
500-bit NECST	63.238	74.402	89.619	117.803
500-bit ABF	58.992	71.338	**83.192**	116.874
500-bit IABF	**55.351**	**70.525**	83.193	**116.281**
MNIST	0.1	0.2	0.3	0.4
100-bit NECST	14.439	22.556	34.377	48.291
100-bit ABF	13.369	**22.013**	33.319	48.041
100-bit IABF	**13.251**	22.026	**33.176**	**46.555**

Table 1: $\mathcal{L}^2$ squared reconstruction error loss (per image) of NECST vs. IABF. The error is calculated on test set.

[1] The codebase for this work can be found at https://github.com/MinkaiXu/neural-coding-IABF.



(a) noise level: 0.1

(b) noise level: 0.2

(c) noise level: 0.3

(d) noise level: 0.4

Figure 2: $\mathcal{L}^2$ reconstruction error (per image) of IABF vs. NECST on CIFAR-10 dataset. The error is calculated on validation set during training. Different figure correspond to different noise level. Red: NECST. Green: IABF.

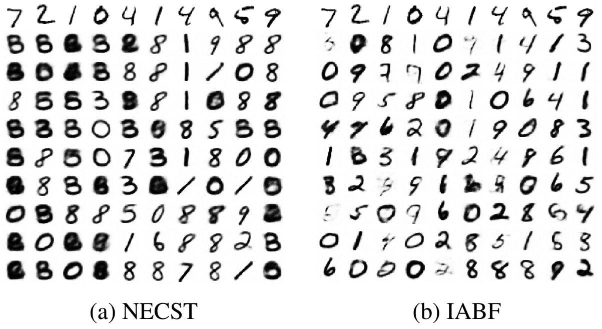
Another practical challenge lies in the optimization procedure of $\mathcal{L}_{rec}(\phi, \theta; \boldsymbol{x}, \epsilon)$, the gradient of the encoder parameters θ is non-trivial to estimate. Hence we adapted VIMCO (Mnih and Rezende 2016; Choi et al. 2018), a multi-sample variational lower bound objective for obtaining low-variance gradients. Following (Choi et al. 2018), we also use the multi-sample objective with $K = 5$:

$$\mathcal{L}^K_{rec}(\phi, \theta; \boldsymbol{x}, \epsilon) = \sum_{\boldsymbol{x}\in\mathcal{D}} \mathbb{E}_{\boldsymbol{y}^{1:K}\sim p_{adv}(\boldsymbol{y}|\boldsymbol{x};\epsilon,\theta)} \left[\log \frac{1}{K} \sum_{i=1}^{K} q_\phi(\boldsymbol{x}|\boldsymbol{y}^{(i)})\right] \tag{31}$$

Compression and Error Correction

To validate our proposed method, we demonstrate the effectiveness of IABF on compression and error correction and conduct comparison with NECST, which is currently the state-of-the-art joint source-channel coding methods within the finite bit-length setting.

The evaluation setting is directly adapted from NECST (Choi et al. 2018). With the number of the bits m fixed, we report the corresponding distortion levels(reconstruction errors) on test sets with respect to various noise levels ϵ in Table. 1. The test result is reported according to the model with the lowest distortion on the validation set. And as observed, IABF can stably outperform the NECST on all binary and RGB datasets within all noise levels from $\epsilon = 0.1$ to $\epsilon = 0.4$, which demonstrates the stability of IABF on dealing with different data complexity and different channel noise level. To verify the effectiveness of different components, we conduct ablation study on MNIST dataset. ABF stands for simply applying Adversarial Bit Flip without mutual information maximization term. It is shown that the ABF itself can already stably outperform NECST, and the with the information maximization term IABF can further decrease the distortion.



(a) NECST

(b) IABF

Figure 3: Markov chain samples from NECST and IABF. Models are trained on MNIST dataset with 0.1 noise level.

Figure. 2 shows the changes of validation reconstruction error with respect to training timesteps for IABF and NECST. We find that the overall training procedure of IABF is more stable than NECST. It is worth noting that at the very beginning of training, the reconstruct error of IABF is higher than NECST. This is due to the fact that at the early stage the encoder is not well trained and hence there is not enough information encoded into the codewords. IABF can more effectively regularize the decoder; however, at this period, the capacity of the well-regularized decoder is not enough to produce good reconstruction. As the training goes on, NECST shows obvious instability and overfitting phenomenon while the advantages of IABF will emerge. The information maximization part enables the model to achieve less distortion, and the adversarial bits flip helps stable the training and avoid over-fitting.

Implicit Generative Modeling

As shown in (Grover and Ermon 2018; Choi et al. 2018), the latent variable model in IABF and NECST specifies an implicit generative model(Mohamed and Lakshminarayanan 2016) under certain conditions. We directly adapt their results here: starting from any data point $x^{(0)} \sim \mathcal{X}$, we define a Markov chain over $\mathcal{X} \times \mathcal{Y}$ with the following transitions:

$$\boldsymbol{y}^{(t)} \sim p_{\text{noisy}}\left(\boldsymbol{y}|\boldsymbol{x}^{(t)}; \theta, \epsilon\right), \boldsymbol{x}^{(t+1)} \sim q_\phi(\boldsymbol{x}|\boldsymbol{y}^{(t)}) \tag{32}$$

We show the samples obtained by running the chain for the model trained with IABF and NECST in Figure. 3. It can be found that the IABF is able to learn a generative model with both better sample quality and diversity than NECST. More specifically, the generative model implied by NECST shows severe model collapse phenomenon due to fact that the information on codewords may be concentrated on several dimensions, which will results in several clusters in the marginal codeword distribution. Hence during sampling, the sampled data may fall into some clusters. While in IABF, both the information maximization term and the adversarial bit flip tend to encourage codewords uniformly distributed, hence there will be fewer clusters and barriers in the latent codeword space which will make the ergodic condition for Markov sampling well satisfied.

Conclusion

We propose IABF to improve neural joint source-channel coding, where the information theoretic dependency between codewords and data is enhanced without the involvement of parameterized variational distribution and the amortized decoder is also regularized in an adversarial fashion. Experimental results demonstrate that IABF is able to stably improve both the compression and error correction ability of the coding scheme within various kinds of data and noise levels. Active learning may be another option to implement adversarial bits flip, we leave it as future direction.

Acknowledgement

The corresponding author Shuo Shao thanks the support of National Natural Foundation of China (61872149, 61901261), Shanghai Sailing Program (19YF1424200).

References

Bachman, P.; Alsharif, O.; and Precup, D. 2014. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, 3365–3373.

Ballard, D. H. 1987. Modular learning in neural networks. In *AAAI*, 279–284.

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*.

Berlekamp, E.; McEliece, R.; and Van Tilborg, H. 1978. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory* 24(3):384–386.

Brown, G. 2009. A new perspective for information theoretic feature selection. In *Artificial intelligence and statistics*, 49–56.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, 2172–2180.

Choi, K.; Tatwawadi, K.; Weissman, T.; and Ermon, S. 2018. Necst: Neural joint source-channel coding. *arXiv preprint arXiv:1811.07557*.

Erin Liong, V.; Lu, J.; Wang, G.; Moulin, P.; and Zhou, J. 2015. Deep hashing for compact binary codes learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2475–2483.

Feldman, J.; Wainwright, M. J.; and Karger, D. R. 2005. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory* 51(3):954–972.

Goodfellow, I. J. 2014. On distinguishability criteria for estimating generative models. *arXiv preprint arXiv:1412.6515*.

Grover, A., and Ermon, S. 2018. Uncertainty autoencoders: Learning compressed representations via variational information maximization. *arXiv preprint arXiv:1812.10539*.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; and Sugiyama, M. 2017. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1558–1567. JMLR. org.

Kim, H., and Mnih, A. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kostina, V., and Verdú, S. 2013. Lossy joint source-channel coding in the finite blocklength regime. *IEEE Transactions on Information Theory* 59(5):2545–2575.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338.

LeCun, Y. 1998. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Miyato, T.; Maeda, S.-i.; Koyama, M.; Nakae, K.; and Ishii, S. 2015. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*.

Mnih, A., and Rezende, D. J. 2016. Variational inference for monte carlo objectives. *arXiv preprint arXiv:1602.06725*.

Mohamed, S., and Lakshminarayanan, B. 2016. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*.

Pilc, R. J. 1967. *Coding theorems for discrete source-channel pairs.* Ph.D. Dissertation, Massachusetts Institute of Technology.

Richardson, T., and Urbanke, R. 2008. *Modern coding theory*. Cambridge university press.

Rolfe, J. T. 2016. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*.

Santurkar, S.; Budden, D.; and Shavit, N. 2018. Generative compression. In *2018 Picture Coding Symposium (PCS)*, 258–262. IEEE.

Shannon, C. E. 1948. A mathematical theory of communication. *Bell system technical journal* 27(3):379–423.

Shu, R.; Bui, H. H.; Zhao, S.; Kochenderfer, M. J.; and Ermon, S. 2018. Amortized inference regularization. In *Advances in Neural Information Processing Systems*, 4393–4402.

Theis, L.; Shi, W.; Cunningham, A.; and Huszár, F. 2017. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*.

Tian, C.; Chen, J.; Diggavi, S. N.; and Shamai, S. 2013. Optimality and approximate optimality of source-channel separation in networks. *IEEE Transactions on Information Theory* 60(2):904–918.

van den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 6306–6315.

Vontobel, P. O., and Koetter, R. 2007. On low-complexity linear-programming decoding of ldpc codes. *European transactions on telecommunications* 18(5):509–517.

Zhao, S.; Song, J.; and Ermon, S. 2017. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*.

Zhao, S.; Song, J.; and Ermon, S. 2018. The information autoencoding family: A lagrangian perspective on latent variable generative models. *arXiv preprint arXiv:1806.06514*.