Improving Lossless Compression Rates via Monte Carlo Bits-Back Coding

Yangjun Ruan [*1 2] Karen Ullrich [*2 3] Daniel Severo [*1 2] James Townsend [4] Ashish Khisti [1] Arnaud Doucet [5]
Alireza Makhzani [1 2] Chris J. Maddison [1 2]

Abstract

Latent variable models have been successfully applied in lossless compression with the bits-back coding algorithm. However, bits-back suffers from an increase in the bitrate equal to the KL divergence between the approximate posterior and the true posterior. In this paper, we show how to remove this gap asymptotically by deriving bits-back coding algorithms from tighter variational bounds. The key idea is to exploit extended space representations of Monte Carlo estimators of the marginal likelihood. Naively applied, our schemes would require more initial bits than the standard bits-back coder, but we show how to drastically reduce this additional cost with couplings in the latent space. When parallel architectures can be exploited, our coders can achieve better rates than bits-back with little additional cost. We demonstrate improved lossless compression rates in a variety of settings, especially in out-of-distribution or sequential data compression.

1. Introduction

Datasets keep getting bigger; the recent CLIP model was trained on 400 million text-image pairs gathered from the internet (Radford et al., 2020). With datasets of this size coming from ever more heterogeneous sources, we need compression algorithms that can store this data efficiently.

In principle, data compression can be improved with a better approximation of the data generating distribution. Luckily, the quality of generative models is rapidly improving (van den Oord et al., 2016; Salimans et al., 2017; Razavi et al., 2019; Vahdat & Kautz, 2020). From this panoply of generative models, latent variable models are particularly attractive for compression applications, because they are typically easy to parallelize; speed is a major concern for compression methods. Indeed, some of the most successful learned compressors for large scale natural images are based on deep latent variable models (see Yang et al. (2020) for lossy, and Townsend et al. (2020) for lossless).

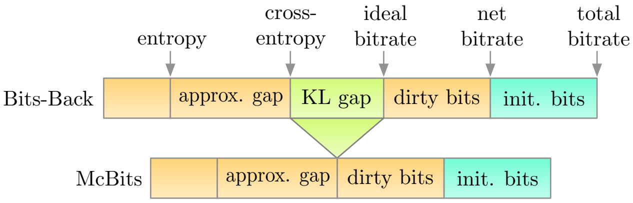


Figure 1. Monte Carlo Bits-Back coders reduce the KL gap to zero.

Lossless compression with latent variable models has a complication that must be addressed. Any model-based coder needs to evaluate the model's probability mass function $p(x)$. Latent variable models are specified in terms of a joint probability mass function $p(x, z)$, where z is a latent, unobserved variable. Computing $p(x)$ in these models requires a (typically intractable) summation, and achieving the model's optimal bitrate, $-\log p(x)$, is not always feasible. When compressing large datasets of i.i.d. data, it is possible to approximate this optimal bitrate using the bits-back coding algorithm (Hinton & Van Camp, 1993; Townsend et al., 2019). Bits-back coding is based on variational inference (Jordan et al., 1999), which uses an approximation $q(z \mid x)$ to the true posterior $p(z \mid x)$. Unfortunately, this adds roughly $D_{\mathrm{KL}}(q(z \mid x) \parallel p(z \mid x))$ bits to the bitrate. This seems unimprovable for a fixed q, which is a problem, if approximating $p(z \mid x)$ is difficult or expensive.

In this paper, we show how to remove (asymptotically) the D_{KL} gap of bits-back schemes for (just about) any fixed q. Our method is based on recent work that derives tighter variational bounds using Monte Carlo estimators of the marginal likelihood (e.g., Burda et al., 2015). The idea is that q and p can be lifted into an extended latent space (e.g., Andrieu et al., 2010) such that the D_{KL} over the extended latent space goes to zero and the overall bitrate approaches $-\log p(x)$. For example, our simplest extended bits-back method, based on importance sampling, introduces N identically distributed particles z_i and a categorical random vari-

[*] Equal contribution [1] University of Toronto [2] Vector Institute [3] Facebook AI Research [4] University College London [5] University of Oxford. Correspondence to: Yangjun Ruan, Daniel Severo, Chris Maddison <yjruan@cs.toronto.edu, d.severo@mail.utoronto.ca, cmaddis@cs.toronto.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

able that picks from z_i to approximate $p(z \,|\, x)$. We also define extended bits-back schemes based on more advanced Monte Carlo methods (AIS, Neal, 2001; SMC, Doucet et al., 2001).

Adding $\mathcal{O}(N)$ latent variables introduces another challenge that we show how to address. Bits-back requires an initial source of bits, and, naively applied, our methods increase the initial bit cost by $\mathcal{O}(N)$. One of our key contributions is to show that this cost can be reduced to $\mathcal{O}(\log N)$ for some of our coders using couplings in latent space, a novel technique that may be applicable in other settings to reduce initial bit costs. Most of our coders can be parallelized over the number of particles, which significantly reduces the computation overhead. So, our coders extend bits-back with little additional cost.

We test our methods in various lossless compression settings, including compression of natural images and musical pieces using deep latent variable models. We report between 2% - 19% rate savings in our experiments, and we see our most significant improvements when compressing out-of-distribution data or sequential data. We also show that our methods can be used to improve the entropy coding step of learned transform coders with hyperpriors (Ballé et al., 2018) for lossy compression. We explore the factors that affect the rate savings in our setting.

2. Background

2.1. Asymmetric Numeral Systems

The goal of lossless compression is to find a short binary representation of the outcome of a discrete random variable $x \sim p_d(x)$ in a finite symbol space $x \in \mathbb{S}$. Achieving the best possible expected length, i.e., the entropy $H(p_d)$ of p_d, requires access to p_d, and typically a *model* probability mass function (PMF) $p(x)$ is used instead. In this case, the smallest achievable length is the *cross-entropy* of p relative to p_d, $H(p_d, p) = -\sum_x p_d(x) \log p(x)$[1]. See MacKay (2003); Cover & Thomas (1991) for more detail.

Asymmetric numeral systems (ANS) are model-based coders that achieve near optimal compression rates on sequences of symbols (Duda, 2009). ANS stores data in a stack-like 'message' data structure, which we denote m, and provides an inverse pair of functions, encode_p and decode_p, which each process one symbol $x \in \mathbb{S}$ at a time:

$$\begin{aligned} \mathrm{encode}_p &: m, x \mapsto m' \\ \mathrm{decode}_p &: m' \mapsto (m, x). \end{aligned} \tag{1}$$

Encode pushes x onto m, and decode pops x from m'. Both functions require access to routines for computing the cumulative distribution function (CDF) and inverse CDF of p.

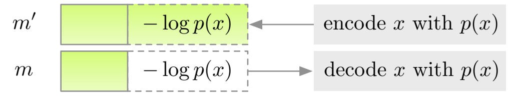


Figure 2. ANS is a last-in-first-out lossless coder. We adopt the visualizations of (Kingma et al., 2019): the green bars represent the message m, a stack that stores symbols x.

If n symbols, drawn i.i.d. from p_d, are pushed onto m with p , the *bitrate* (bits/symbol) required to store the ANS message approaches $H(p_d, p) + \epsilon$ for some small error ϵ (Duda, 2009; Townsend, 2020). The exact sequence is recovered by popping n symbols off the final m with p.

For our purposes, the ANS message can be thought of as a store of randomness. Given a message m with enough bits, regardless of m's provenance, we can decode from m using any distribution p. This will return a random symbol x and remove roughly $-\log p(x)$ bits from m. Conversely, we can encode a symbol x onto m with p, which will increase m's length by roughly $-\log p(x)$. If a sender produces a message m through some sequence of encode or decode steps using distributions p_i, then a receiver, who has access to the p_i, can recover the sequence of encoded symbols and the initial message by reversing the order of operations and switching encodes with decodes. When describing algorithms, we often leave out explicit references to m, instead writing steps like 'encode x with $p(x)$'. See Fig. 2.

2.2. Bits-Back Compression with ANS

The class of latent variable models is highly flexible, and most operations required to compute with such models can be parallelized, making them an attractive choice for model-based coders. Bits-back coders, in particular Bits-Back with ANS (BB-ANS, Townsend et al., 2019), specialize in compression using latent variable models.

A latent variable model is specified in terms of a joint distribution $p(x, z)$ between x and a latent discrete[2] random variable taking value in a symbol space $z \in \mathbb{S}'$. We assume that the joint distribution of latent variable models factorizes as $p(z)p(x \,|\, z)$ and that the PMFs, CDFs, and inverse CDFs, under $p(z)$ and $p(x \,|\, z)$ are tractable. However, computing the marginal $p(x) = \sum_z p(z)p(x \,|\, z)$ is often intractable. This fact means that we cannot directly encode x onto m. A naive strategy would be for the sender to pick some $z \in \mathbb{S}'$, and encode (x, z) using p, which would require $-\log p(x, z)$ bits; however, this involves communicating the symbol z, which is redundant information.

BB-ANS gets a better bitrate, by compressing sequences of symbols in a chain and by having the sender *decode* latents

[1] All logarithms in this paper are base 2.

[2] BB-ANS can easily be extended to continuous z, with negligible cost, by quantizing; see Townsend et al. (2019).

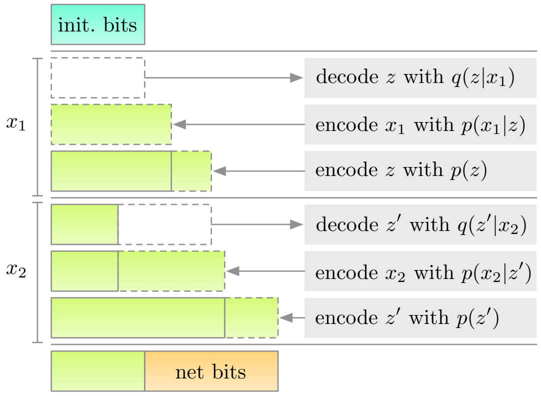

init. bits
x_1
decode z with $q(z|x_1)$
encode x_1 with $p(x_1|z)$
encode z with $p(z)$
x_2
decode z' with $q(z'|x_2)$
encode x_2 with $p(x_2|z')$
encode z' with $p(z')$
net bits

Figure 3. Encoding (x_1, x_2) requires an initial source of bits (light blue), but bits-back reduces its total bit consumption by using intermediate messages as the initial source of bits for encoding x_2. The net bits used (light orange) is close to the negative ELBO.

z from the intermediate message state, rather than picking z; see Fig. 3. Suppose that we have already pushed some symbols onto a message m. BB-ANS uses an approximate posterior $q(z \mid x)$ such that if $p(x, z) = 0$ then $q(z \mid x) = 0$. To encode a symbol x onto m, the sender first pops z from m using $q(z \mid x)$. Then they push (x, z) onto m using $p(x, z)$. The new message m' has approximately $-\log p(x, z) + \log q(z \mid x)$ more bits than m. m' is then used in exactly the same way for the next symbol. The per-symbol rate saving over the naive method is $-\log q(z \mid x)$ bits. However, for the first symbol, an initial message is needed, causing a one-time overhead.

2.3. The Bitrate of Bits-Back Coders

When encoding a sequence of symbols, we define the *total bitrate* to be the number of bits in the final message per symbol encoded; the *initial bits* to be the number of bits needed to initialize the message; and the *net bitrate* to be the total bitrate minus the initial bits per symbol, which is equal to the expected increase in message length per symbol. As the number of encoded symbols grows, the total bitrate of BB-ANS will converge to the net bitrate.

One subtlety is that the BB-ANS bitrate depends on the distribution of the latent z, popped with $q(z \mid x)$. In an ideal scenario, where m contains i.i.d. uniform Bernoulli distributed bits, z will be an exact sample from q. Unfortunately, in practical situations, the exact distribution of z is difficult to characterize. Nevertheless, Townsend et al. (2019) found the 'evidence lower bound' (ELBO)

$$\begin{aligned}&\mathbb{E}_{z \sim q(z \mid x)}\left[-\log p(x, z) + \log q(z \mid x)\right] \\ &= -\log p(x) + D_{\mathrm{KL}}(q(z \mid x) \parallel p(z \mid x)),\end{aligned} \tag{2}$$

which assumes $z \sim q(z \mid x)$, to be an accurate predictor of BB-ANS's empirical compression rate; the effect of inaccurate samples (which they refer to as 'dirty bits') is typically less than 1%. So, in this paper we mostly elide the dirty bits issue, regarding (2) to be the net bitrate of BB-ANS, and hereafter we refer to vanilla BB-ANS as BB-ELBO. Taking the expectation under p_d, BB-ELBO achieves a net bitrate of approximately $H(p_d, p) + \mathbb{E}_{x \sim p_d}[D_{\mathrm{KL}}(q(z \mid x) \parallel p(z \mid x))]$.

3. Monte Carlo Bits-Back Coding

The net bitrate of bits-back is ideally the negative ELBO. This rate seems difficult to improve without finding a better q. However, the ELBO may be a loose bound on the marginal log-likelihood. Recent work in variational inference shows how to bridge the gap from the ELBO to the marginal log-likelihood with tighter variational bounds (e.g., Burda et al., 2015; Domke & Sheldon, 2018), motivating the question: *can we derive bits-back coders from those tighter bounds and approach the cross-entropy?*

In this section we provide an affirmative answer to this question with a framework called Monte Carlo bits-back coding (McBits). We point out that the extended space constructions of Monte Carlo estimators can be reinterpreted as bits-back coders. One of our key contributions is deriving variants whose net bitrates improve over BB-ELBO, while being nearly as efficient with initial bits. We begin by motivating our framework with two worked examples. Implementation details are in Appendix A.

3.1. Bits-Back Importance Sampling

The simplest of our McBits coders is based on importance sampling (IS). IS samples N particles $z_i \sim q(z_i \mid x)$ i.i.d. and uses the importance weights $p(x, z_i)/q(z_i \mid x)$ to estimate $p(x)$. The corresponding variational bound (IWAE, Burda et al., 2015) is the log-average importance weight:

$$-\mathbb{E}_{\{z_i\}_{i=1}^N}\left[\log\left(\sum_{i=1}^N \frac{1}{N}\frac{p(x, z_i)}{q(z_i \mid x)}\right)\right] \geq -\log p(x). \tag{3}$$

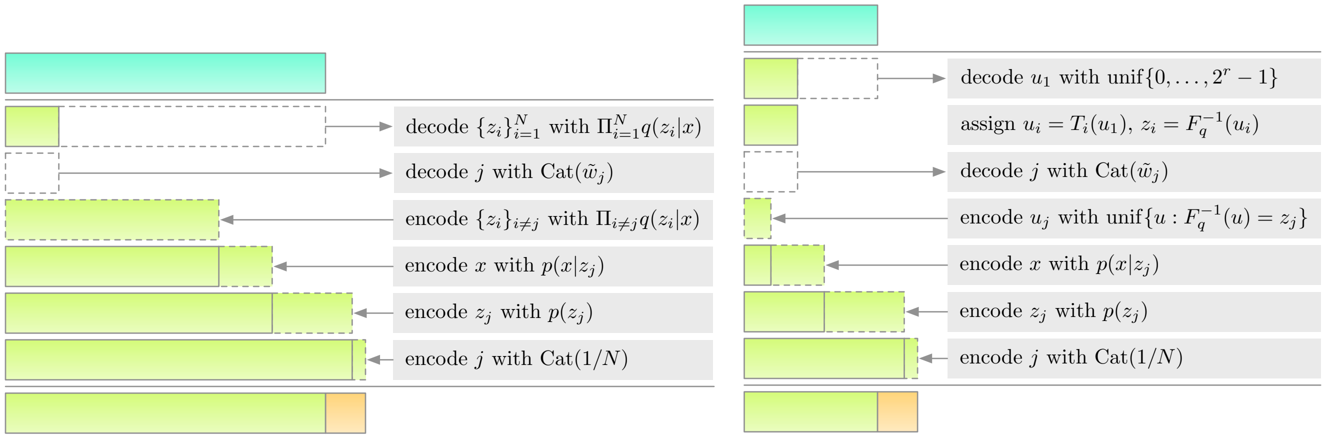
IS provides a consistent estimator of $p(x)$. If the importance weights are bounded, the left-hand side of equation (3) converges monotonically to $-\log p(x)$ (Burda et al., 2015).

Surprisingly, equation (3) is actually the evidence lower bound between a different model and a different approximate posterior on an extended space (Andrieu et al., 2010; Cremer et al., 2017; Domke & Sheldon, 2018). In particular, consider an expanded latent space $\mathbb{S}'^N \times \{1 \mathinner{..} N\}$ that includes the configurations of the N particles $\{z_i\}_{i=1}^N$ and an index $j \in \{1 \mathinner{..} N\}$. The left-hand side of equation (3) can be re-written as the (negative) ELBO between a pair of distributions P and Q defined over this extended latent space, which are given in Alg. 1. Briefly, given x, Q samples N particles $z_i \sim q(z_i \mid x)$ i.i.d. and selects one of them by sampling an index j with probability $\tilde{w}_j \propto p(x, z_j)/q(z_j \mid x)$.

Algorithm 1: Extended Latent Space Representation of Importance Sampling

```
Process $Q(\mathcal{Z} \mid x)$
    sample $\{z_i\}_{i=1}^N \sim \prod_{i=1}^N q(z_i \mid x)$
    compute $\tilde{w}_i \propto \frac{p(x,z_i)}{q(z_i \mid x)}$
    sample $j \sim \text{Cat}(\tilde{w}_j)$
    return $\{z_i\}_{i=1}^N, j$

Process $P(x, \mathcal{Z})$
    sample $j \sim \text{Cat}(1/N)$
    sample $z_j \sim p(z_j)$
    sample $x \sim p(x \mid z_j)$
    sample $\{z_i\}_{i \neq j} \sim \prod_{i \neq j} q(z_i \mid x)$
    return $x, \{z_i\}_{i=1}^N, j$
```


decode $\{z_i\}_{i=1}^N$ with $\Pi_{i=1}^N q(z_i|x)$
decode j with $\text{Cat}(\tilde{w}_j)$
encode $\{z_i\}_{i\neq j}$ with $\Pi_{i\neq j} q(z_i|x)$
encode x with $p(x|z_j)$
encode z_j with $p(z_j)$
encode j with $\text{Cat}(1/N)$
decode u_1 with $\text{unif}\{0,\ldots,2^r-1\}$
assign $u_i = T_i(u_1)$, $z_i = F_q^{-1}(u_i)$
decode j with $\text{Cat}(\tilde{w}_j)$
encode u_j with $\text{unif}\{u : F_q^{-1}(u) = z_j\}$
encode x with $p(x|z_j)$
encode z_j with $p(z_j)$
encode j with $\text{Cat}(1/N)$

(a) Bits-Back Importance Sampling (BB-IS)

(b) Bits-Back Coupled Importance Sampling (BB-CIS)

Figure 4. The initial bit cost of encoding a single symbol x with IS-based coders is reduced from $\mathcal{O}(N)$ to $\mathcal{O}(\log N)$ by coupling the latents with shared randomness. Both of these coders achieve a net bitrate that approaches $-\log p(x)$ as $N \to \infty$.

The distribution P pre-selects the special jth particle uniformly at random, samples its value $z_j \sim p(z_j)$ from the prior of the underlying model, and samples $x \sim p(x \mid z_j)$ given z_j. The remaining $z_i \sim q(z_i \mid x)$ for $i \neq j$ are sampled from the underlying approximate posterior. Because P and Q use q for all but the special jth particle, most of the terms in the difference of the log-mass functions cancel, and all that remains is equation (3). See Appendix A.4.

Once we identify the left-hand side of equation (3) as a negative ELBO over the extended space, we can derive a bits-back scheme that achieves an expected net bitrate equal to equation (3). We call this the Bits-Back Importance Sampling (BB-IS) coder, and it is visualized in Fig. 4a. To encode a symbol x, we first decode N particles z_i and the index j with the Q process by translating each 'sample' to 'decode'. Then we encode x, the particles z_i, and the index j jointly with the P process by translating each 'sample' to 'encode' in *reverse order*. By reversing P at encode time, we ensure that receiver decodes with P in the right order.

Thus, ignoring the clean bits question, BB-IS's asymptotic net bitrate is close to the left-hand side of equation (3), which converges to the marginal log-likelihood (Burda et al., 2015). Ultimately, as $N \to \infty$, it reaches the cross-entropy.

3.2. Bits-Back Coupled Importance Sampling

Unfortunately, the BB-IS coder requires roughly $-\log \tilde{w}_j - \sum_{i=1}^N \log q(z_i \mid x) \in \mathcal{O}(N)$ initial bits. The reason is that each decoded random variable needs to remove some bits from m. Can this be avoided? Here, we design Bits-Back Coupled Importance Sampling (BB-CIS), which achieves a net bitrate comparable to BB-IS while reducing the initial bit cost to $\mathcal{O}(\log N)$. BB-CIS achieves this by decoding a *single* common random number, which is shared by the z_i. The challenge is showing that a net bitrate comparable to BB-IS is still achievable under such a reparameterization.

BB-CIS is based on a reparameterization of the particles z_i as deterministic functions of coupled uniform random variables. The method is a discrete analog of the inverse CDF technique for simulating non-uniform random variates (Devroye, 2006). Specifically, suppose that the latent space $\mathbb{S}'$ is totally ordered, and the probabilities of q are approximated to an integer precision $r > 0$, i.e., $2^r q(z \mid x)$ is an integer for all $z \in \mathbb{S}'$. Define the function $F_q^{-1} : \{0 \,..\, 2^r - 1\} \to \mathbb{S}'$,

$$F_q^{-1}(u) = \arg\min \left\{ z : \sum\nolimits_{z' \leq z} 2^r q(z' \mid x) > u \right\}. \quad (4)$$

F_q^{-1} maps the uniform samples into samples from q. This is visualized in Fig. 5. A coupled set of particles z_i with

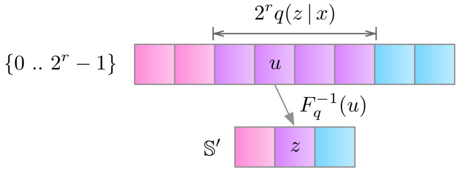


Figure 5. Mapping uniforms in $\{0 \mathrel{..} 2^r - 1\}$ to samples in $\mathbb{S}'$. Colors represent the subsets mapped to z of size $2^r q(z\,|\,x)$.

marginals $q(z_i\,|\,x)$ can be simulated with a common random number by sampling a single uniform u and setting $z_i = F_q^{-1}(T_i(u))$ for some functions $T_i : \{0 \mathrel{..} 2^r - 1\} \to \{0 \mathrel{..} 2^r - 1\}$, as in Fig. 6. Intuitively, T_i maps u to the uniform u_i underlying z_i. To ensure that z_i have the same marginal, we require that T_i are bijective functions. For example, T_i can be defined as a fixed 'shift' by integer k_i , i.e., $T_i(u) = (u + k_i) \bmod 2^r$ with inverse $T_i^{-1}(u) = (u - k_i) \bmod 2^r$. We set $T_1(u) = u$ by convention.

BB-CIS uses these couplings in a latent decoding process that saves initial bits. It decodes u_1 from m with $\text{unif}\{0 \mathrel{..} 2^r - 1\}$, sets $z_i = F_q^{-1}(T_i(u_1))$, and decodes the index j of the special particle z_j with $\text{Cat}(\tilde{w}_j)$. This reduces the initial bit cost to $r - \log \tilde{w}_j \in \mathcal{O}(1) + \mathcal{O}(\log N) = \mathcal{O}(\log N)$. Note that the $\mathcal{O}(\log N)$ term is for decoding the index j which does not scale with latent dimension, thus the $\mathcal{O}(1)$ term dominates for high dimensional latents. Also, in the ANS implementation, all compressed message lengths are rounded to a multiple of a specified precision (e.g., 16) which may mask small changes caused by the $\mathcal{O}(\log N)$ term. Thus, in practice, BB-CIS demonstrates a nearly constant initial bits cost (Fig. 8c).

The coupled latent decoding process needs to be matched with an appropriate encoding process for x. Suppose that we finished encoding x, as with BB-IS, by encoding $\{z_i\}_{i \neq j}$ with $q(z_i\,|\,x)$ i.i.d. and encoding (x, z_j, j) with $p(x, z_j)/N$. The net bitrate would be

$$\mathbb{E}_{u_1}\left[-r - \sum_{i=1}^{N} \log q(z_i\,|\,x) - \log\left(\sum_{i=1}^{N} \frac{1}{N}\frac{p(x, z_i)}{q(z_i\,|\,x)}\right)\right].$$

This is clearly worse than BB-IS. The culprits are the $N - 1$ latents that BB-IS pushes onto m, which is wasteful, because they are deterministically coupled. Fundamentally, the encoding process of BB-IS is not balanced with the initial bit savings of our coupled latent decoding process.

The solution is to design an encoding process over $\{z_i\}_{i=1}^N$, $\{u_i\}_{i=1}^N$, and x, which exactly matches the initial bit savings of the coupled decoding process. The idea is to encode just (u_j, x, z_j, j), which is enough information for the receiver to reconstruct all other variables. The key is to design the encoding for u_j. Encoding u_j with a uniform on $\{0 \mathrel{..} 2^r - 1\}$ is unnecessarily wasteful, because z_j restricts the range of u_j. It turns out, that the best we can do is to encode u_j with $\text{unif}\{u : F_q^{-1}(u) = z_j\}$. Finally (x, z_j, j) are encoded with $p(x, z_j)/N$. BB-CIS's encode is given in Fig. 4b.

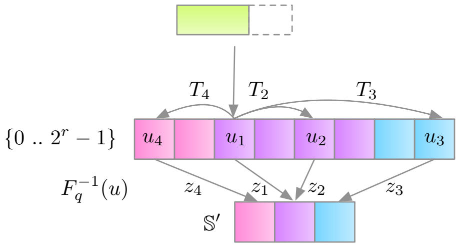


Figure 6. BB-CIS only needs to decode a single random uniform u and then applies bijections T_i to produce the uniforms u_i underlying z_i. We set $T_1(u) = u$ by convention.

BB-CIS has the following expected net bitrate:

$$-\,\mathbb{E}_{u_1}\left[\log\left(\sum_{i=1}^{N} \frac{1}{N}\frac{p(x, z_i)}{q(z_i\,|\,x)}\right)\right]. \tag{5}$$

Thus, BB-CIS achieves a net bitrate comparable to BB-IS, but uses only $\mathcal{O}(\log N)$ initial bits. We show this in Appendix A.5. We also show that the BB-CIS net bitrate can be interpreted as a negative ELBO over an extended latent space, which accounts for the configurations of z_i, u_i, and j. This extended space construction is novel, and we believe it may be useful for deriving other coupling schemes that reduce initial bit consumption. Common random numbers are classical tools in Monte Carlo methods; in bits-back they may serve as a tool for controlling initial bit cost.

The net bitrate of BB-CIS can fail to converge to $-\log p(x)$, if the T_i are poorly chosen. In our experiments, we used fixed (but randomly chosen) shifts shared between the sender and receiver. This scheme converged as quickly as BB-IS in terms of net bitrate, but at a greatly reduced total bitrate. However, we point out that further work is required to explore more efficient bijections in terms of computation cost and compression performance.

3.3. General Framework

Monte Carlo bits-back coders generalize these two examples. They are bits-back coders built from extended latent space representations of Monte Carlo estimators of the marginal likelihood. Let $\hat{p}_N(x)$ be a positive unbiased Monte Carlo estimator of the marginal likelihood that can be simulated with $\mathcal{O}(N)$ random variables, i.e., $\mathbb{E}[\hat{p}_N(x)] = p(x)$. Importance sampling is the quintessential example, $\hat{p}_N(x) = N^{-1}\sum_{i=1}^N p(x, z_i)/q(z_i\,|\,x)$, but more efficient estimators of $p(x)$ can be built using techniques like annealed importance sampling (Neal, 2001) or sequential Monte Carlo (Doucet et al., 2001).

A variational bound on the log-marginal likelihood can be derived from $\hat{p}_N(x)$ by Jensen's inequality,

$$-\mathbb{E}[\log \hat{p}_N(x)] \geq -\log p(x). \tag{6}$$

If $\hat{p}_N(x)$ is strongly consistent in N (as is the case with many of these estimators) and $-\log \hat{p}_N(x)$ satisfies a uniform integrability condition, then $-\mathbb{E}[\log \hat{p}_N(x)] \to -\log p(x)$ (Maddison et al., 2017). This framework captures recent efforts on tighter variational bounds (Burda et al., 2015; Maddison et al., 2017; Naesseth et al., 2018; Le et al., 2018; Domke & Sheldon, 2018; Caterini et al., 2018).

As with BB-IS and BB-CIS, the key step in the McBits framework is to identify an extended latent space representation of $\hat{p}_N(x)$. Let $\mathcal{Z} \sim Q(\mathcal{Z} \mid x)$ be a set of random variables (often including those needed to compute $\hat{p}_N(x)$). If there exists a 'target' probability distribution $P(x, \mathcal{Z})$ over x and $\mathcal{Z}$ with marginal $p(x)$ such that

$$\hat{p}_N(x) = \frac{P(x, \mathcal{Z})}{Q(\mathcal{Z} \mid x)}, \tag{7}$$

then the McBits coder, which decodes $\mathcal{Z}$ with $Q(\mathcal{Z} \mid x)$ and encodes $(x, \mathcal{Z})$ with $P(x, \mathcal{Z})$, will achieve a net bitrate of $-\log \hat{p}_N(x)$. In particular, if the log estimator converges in expectation to the log-marginal likelihood and we ignore the dirty bits issue, then $D_{\mathrm{KL}}(Q(\mathcal{Z} \mid x) \,\|\, P(\mathcal{Z} \mid x)) \to 0$ and the McBits coder will achieve a net bitrate of $H(p_d, p)$.

The challenge is to identify $\mathcal{Z}$, Q, and P. While Monte Carlo estimators of $p(x)$ get quite elaborate, many of them admit such extended latent space representations (e.g., Neal, 2001; Andrieu et al., 2010; Finke, 2015; Domke & Sheldon, 2018). These constructions are techniques for proving the unbiasedness of the estimators; one of our contributions is to demonstrate that they can become efficient bits-back schemes. In Appendix A, we provide pseudocode and details for all McBits coders.

Bits-Back Annealed Importance Sampling (BB-AIS) Annealed importance sampling (AIS) is a generalization of importance sampling, which introduces a path of N intermediate distributions between the base distribution $q(z \mid x)$ and the unnormalized posterior $p(z \mid x)$. AIS samples a sequence of latents by iteratively applying MCMC transition kernels that leave each intermediate distributions invariant. By bridging the gap between $q(z \mid x)$ and $p(z \mid x)$, AIS's estimate of $p(x)$ typically converges faster than importance sampling (Neal, 2001). The corresponding McBits coder, BB-AIS, requires $\mathcal{O}(N)$ initial bits, but this can be addressed with the BitSwap trick (Kingma et al., 2019), which we call BB-AIS-BitSwap. Another issue is that the intermediate distributions are usually not factorized, which makes it challenging to work with high-dimensional z.

Bits-Back Sequential Monte Carlo (BB-SMC) Sequential Monte Carlo (SMC) is a particle filtering method that combines importance sampling with resampling. Its estimate of $p(x)$ typically converges faster than importance sampling for time series models (Cérou et al., 2011; Bérard et al., 2014). SMC maintains a population of particles. At each time step, the particles independently sample an extension from the proposal distribution. Then the whole population is resampled with probabilities in proportion to importance weights. The corresponding McBits coder, BB-SMC, requires $\mathcal{O}(TN)$ initial bits, where T is the length of the time series. We introduce a coupled variant, BB-CSMC, in the appendix that reduces this to $\mathcal{O}(T \log N)$.

Computational Cost All of our coders require $\mathcal{O}(N)$ computational cost, but the IS- and SMC-based coders are amenable to parallelization over particles. In particular, we implemented an end-to-end parallelized version of BB-IS based on the JAX framework (Bradbury et al.) and benchmarked on compressing the binarized MNIST dataset with one-layer VAE model. As shown in Fig. 7, the computation time scales sublinearly with the number of particles, which demonstrates the potential practicality of our method, in some settings, with hundreds of particles. Detailed discussion is in Appendix B.

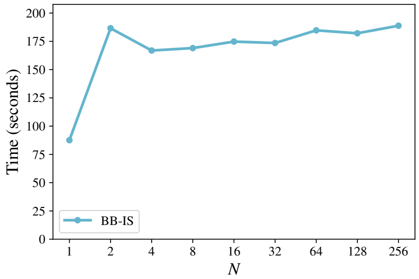
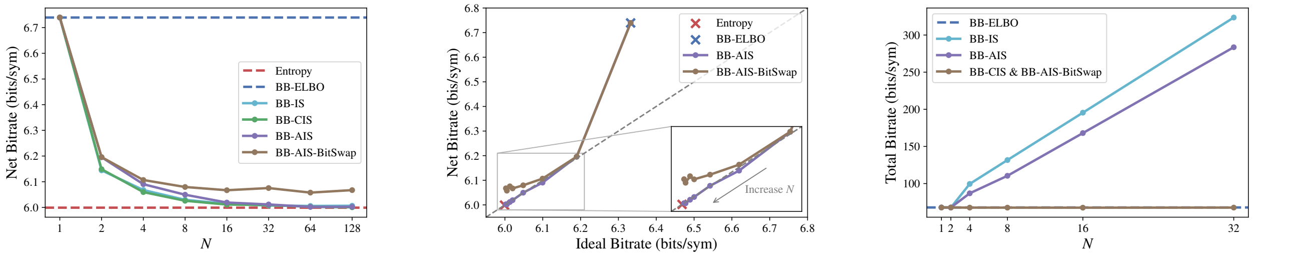


Figure 7. BB-IS total encode + decode time for binarized MNIST scales very well with N using a parallel implementation of ANS. BB-ELBO ($N = 1$) uses a simpler code-base, which runs faster. The experiment was run on a Tesla P100 GPU with 12GB of memory, together with an Intel Xeon Silver 4110 CPU at 2.10GHz.

4. Related Work

Recent work on learned compressors considers two classes of deep generative models: models based on normalizing flows (Rippel & Adams, 2013; Dinh et al., 2014; 2016; Rezende & Mohamed, 2015) and deep latent variable models (Kingma & Welling, 2013; Rezende et al., 2014). Flow models are generally computationally expensive, and optimizing a function with discrete domain and codomain using gradient descent can be cumbersome. Despite these issues, flow models are the state of the art for lossless compression of small images (van den Berg et al., 2020).



(a) As $N \to \infty$, the net bitrate converges to the entropy for most coders on the toy mixture model.

(b) As $N \to \infty$, the net bitrate converges. Convergence to the ideal bitrate (dashed line) indicates clean bits.

(c) The initial bit cost (reflected in the total bitrate after the first symbol) is controlled by some McBits coders, but not others.

Figure 8. The bitrate of McBits coders converges (in the number of particles N) to the entropy when using the data generating distribution. The initial bit cost of naive coders scales like $\mathcal{O}(N)$, but coupled and BitSwap variants significantly reduce it. Bitrates are bits/sym.

'Bits-back' was originally meant to provide an information theoretic basis for approximate Bayesian inference methods (Wallace, 1990; Hinton & Van Camp, 1993; Frey & Hinton, 1997; Frey, 1998). Townsend et al. (2019) showed that the idea can lead directly to a practical compression algorithm for latent variable models. Follow up work reduced the initial bit cost for hierarchical models (Kingma et al., 2019), and extended it to large scale models and larger images (Townsend et al., 2020). For lossy compression, the variational autoencoder framework is a natural fit for training transform coders (Johnston et al., 2019; Ballé et al., 2016; 2018; Minnen et al., 2018; Yang et al., 2020).

Relative entropy coding (REC, Havasi et al., 2019; Flamich et al., 2020) is an alternative coding scheme for latent variable models that seeks to address the initial bits overhead in bits-back schemes. However, practical REC implementations require a particular reparameterization of the latent space, and it is unclear whether the REC latent structure is compatible with the extended latents in McBits.

5. Experiments

We studied the empirical properties and performance of our McBits coders on both synthetic data and practical image and music piece compression tasks. Many of our experiments used continuous latent variable models and we adopted the maximum entropy quantization in Townsend et al. (2019) to discretize the latents. We sometimes evaluated the *ideal bitrate*, which for each coder is the corresponding variational bound estimated with pseudorandom numbers. For continuous latent variable models, the ideal bitrate does not account for quantization. We rename BB-ANS to BB-ELBO. N refers to the number of intermediate distributions for BB-AIS. Details are in Appendix C. Our implementation is available at https://github.com/ryoungj/mcbits.

5.1. Lossless Compression on Synthetic Data

We assessed the convergence properties, impact of dirty bits, and initial bit cost of our McBits coders on synthetic data. First, a dataset of 5000 symbols was generated i.i.d. from a mixture model with alphabet sizes 64 and 256 for the observations and latents, respectively. BB-ELBO, BB-IS, BB-CIS, and BB-AIS were evaluated using the true data generating distribution with a uniform approximate posterior, ensuring a large mismatch with the true posterior. For BB-AIS, a Metropolis–Hastings kernel with a uniform proposal was used. The bijective operators of BB-CIS applied randomly selected, but fixed, shifts to the sampled uniform.

The net bitrates of BB-IS, BB-CIS, and BB-AIS converged to the entropy (optimal rate) as the number of particles increased. This is shown in 8a. The indistinguishable gap between BB-CIS and BB-IS illustrates that particle coupling did not lead to a deterioration of net bitrate. BB-AIS-BitSwap did not converge, likely due to the dirty bits issue.

We measured the impact of dirty bits by plotting *ideal* versus *net* bitrates in Fig. 8b. The deviation of any point to the dashed line indicates the severity of dirty bits. Interestingly, most of our McBits coders appeared to 'clean' the bitstream as N increased, i.e. the net bitrate converged to the entropy, as shown in Fig. 8b for BB-AIS and in Fig. 14a in Appendix C.1 for all coders. Only BB-AIS-BitSwap did *not* clean the bitstream (Fig. 8b), indicating that the order of operations has a significant impact on the cleanliness of McBits coders.

We quantified the initial bits cost by computing the *total* bitrate after the first symbol. As shown in Fig. 8c, it increased linearly with N for BB-IS and BB-AIS, but remained fixed for BB-CIS and BB-AIS-BitSwap.

Our second experiment was with a dataset of 5000 symbol subsequences generated i.i.d. from a small hidden Markov model (HMM). We used 10 timesteps with alphabet sizes 16 and 32 for the observations and latents, respectively. BB-

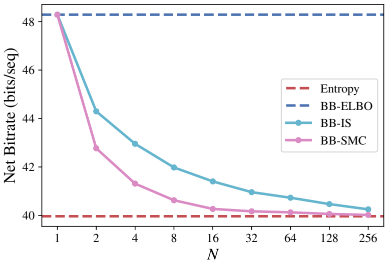


Figure 9. As $N \to \infty$, the net bitrates of BB-IS and BB-SMC converge to the entropy on the toy HMM, but BB-SMC converges much faster.

Table 1. BB-IS performs better on models that were trained with the same number of particles. The net bitrates (bits/dim) of BB-IS on EMNIST-MNIST and CIFAR-10 test sets. $N = 50$ for EMNIST-MNIST and $N = 10$ for CIFAR-10.

	MNIST		CIFAR-10	
	ELBO	IWAE	ELBO	IWAE
BB-ELBO	0.236	0.236	4.898	4.898
BB-IS (5)	0.232	0.231	4.866	4.827
BB-IS (N)	0.230	**0.228**	4.857	**4.810**
Savings	2.5%	3.4%	0.8%	1.8%

ELBO, BB-IS, and BB-SMC were evaluated using the true data generating distribution and a uniform approximate posterior. The net bitrates of BB-IS and BB-SMC converged to the entropy, but BB-SMC converged much faster, illustrating the effectiveness of resampling particles for compressing sequential data (Fig. 9).

5.2. Lossless Compression on Images

We benchmarked the performance of BB-IS and BB-CIS on the standard train-test splits of two datasets: an alphanumeric extension of MNIST called EMNIST (Cohen et al., 2017), and CIFAR-10 (Krizhevsky, 2009). EMNIST was dynamically binarized following Salakhutdinov & Murray (2008), and a VAE with 1 stochastic layer was used as in Burda et al. (2015). For CIFAR-10, we used VQ-VAE (Oord et al., 2017) with discrete latent variables and trained with continuous relaxations (Sønderby et al., 2017).

The variational bounds used to train the VAEs had an impact on compression performance. When using BB-IS with a model trained on the IWAE objective, equalizing the number of particles during compression and training resulted in better rates than BB-IS with an ELBO-trained VAE (Table 1). Therefore, we always use our McBits coders with models trained on the corresponding variational bound.

We assessed BB-IS in an out-of-distribution (OOD) compression setting. We trained models on standard EMNIST-Letters and EMNIST-MNIST splits and evaluated compression performance on the test sets. BB-IS achieved greater rate savings than BB-ELBO when transferred to OOD data (Table 2). This illustrates that BB-IS may be particularly useful in more practical compression settings where the data distribution is different from that of the training data.

Table 2. BB-IS leads to more improved compression rates in out-of-distribution compression settings. The net bitrates (bits/dim) of BB-IS on EMNIST test sets using a VAE.

Trained on	MNIST		Letters	
Compressing	MNIST	Letters	MNIST	Letters
BB-ELBO	0.236	0.310	0.257	0.250
BB-IS (5)	0.231	0.289	0.249	0.243
BB-IS (50)	0.228	0.280	0.244	0.239
Savings	3.4%	**9.7%**	**5.1%**	4.4%

Table 3. BB-CIS achieves the best total bitrates compared to baselines on EMNIST test sets.

Method	MNIST	Letters
PNG	0.819	0.900
WebP	0.464	0.533
gzip	0.423	0.413
lzma	0.383	0.369
bz2	0.375	0.364
BB-ELBO	0.236	0.250
BB-ELBO-IF (50)	0.233	0.246
BB-IS (50)	0.230	0.241
BB-CIS (50)	**0.228**	**0.239**

Finally, we compared BB-IS and BB-CIS to other benchmark lossless compression schemes by measuring the *total* bitrates on EMNIST test sets (without transferring). We also compared with amortized-iterative inference, (Yang et al., 2020) that optimizes the ELBO objective over local variational parameters for each data example at the compression stage. To roughly match the computation budget, the number of optimization steps was set to 50 and this method is denoted as BB-ELBO-IF (50). Both BB-IS and BB-CIS outperformed all other baselines on both test sets, and BB-CIS was better than BB-IS in terms of total bitrate since it effectively reduces the initial bit cost. Additional results can be found in in Appendix C.2.

5.3. Lossless Compression on Sequential Data

We quantified the performance of BB-SMC on sequential data compression tasks with 4 polyphonic music datasets: Nottingham, JSB, MuseData, and Piano-midi.de (Boulanger-Lewandowski et al., 2012). All datasets were composed of sequences of binary 88-dimensional vectors representing active notes. The sequence lengths were very

Table 4. BB-SMC achieves the best net bitrates (bits/timestep) on all piano roll test sets.

	Musedata	Nott.	JSB	Piano.
BB-ELBO	10.66	5.87	12.53	11.43
BB-IS (4)	10.66	4.86	12.03	11.38
BB-SMC (4)	**9.58**	**4.76**	**10.92**	**11.20**
Savings	10.1%	18.9%	12.8%	2.0%

imbalanced, so we chunked the datasets to sequences with maximum length of 100. We trained variational recurrent neural networks (VRNN, Chung et al., 2015) on these chunked datasets using code from Maddison et al. (2017). For each dataset, 3 VRNN models were trained with the ELBO, IWAE and FIVO objectives with 4 particles and were used with their corresponding coders for compression.

We compared the *net* bitrates of all coders for compressing each test set in Table 4. BB-SMC clearly outperformed BB-ELBO and BB-IS with the same number of particles on all datasets. We include the comparison with some benchmark lossless compression schemes in Table 10 in Appendix C.3.

5.4. Lossy Compression on Images

Current state-of-the-art lossy image compressors use hierarchical latent VAEs with quantized latents that are losslessly compressed with hyperlatents (Ballé et al., 2016; 2018; Minnen et al., 2018). Yang et al. (2020) observed that the marginalization gap of jointly compressing the latent and the hyperlatent can be bridged by bits-back. Thus, our McBits coders can be used to further reduce the gap.

We experimented on a simplified setting where we used the binarized EMNIST datasets and a modification of the VAE model with 2 stochastic layers in Burda et al. (2015). The major modifications were the following. The distributions over the 1st stochastic layer (latent) were modified to support quantization in a manner similar to (Ballé et al., 2018). We trained the model on a relaxed rate-distortion objective with a hyperparameter λ controlling the trade-off, where the distortion term was the Bernoulli negative log likelihood and the rate term was the negative ELBO or IWAE that only marginalized over the 2nd stochastic layer (hyperlatent). Details are in Appendix C.4.

We evaluated the *net* bitrate savings of BB-IS compared to BB-ELBO on the EMNIST-MNIST test set with different λ values, as in Fig. 10. We found that BB-IS achieved more than 15% rate savings in some setups, see also the rate-distortion curves in Appendix C.4. The performance can be further improved by applying amortized-iterative inference, which is included in Appendix C.4. We also implemented these experiments for the model in (Ballé et al., 2018), but did not observe significant improvements. This may be due to the specific and complex model architecture.

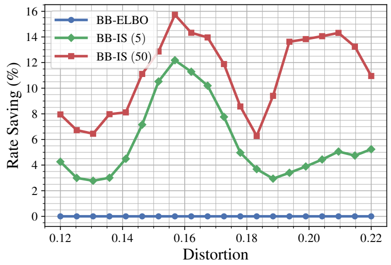


Figure 10. The rate saving curve for lossy compression on the EMNIST-MNIST test set. We measure the net bitrate savings (%) relative to BB-ELBO for fixed distortion values.

6. Conclusion

We showed that extended state space representations of Monte Carlo estimators of the marginal likelihood can be transformed into bits-back schemes that asymptotically remove the D_{KL} gap. In our toy experiments, our coders were 'self-cleaning' in the sense that they reduced the dirty bits gap. In our transfer experiments, our coders had a larger impact on compression rates when compressing out-of-distribution data. Finally, we demonstrated that the initial bit cost incurred by naive variants can be controlled by coupling techniques. We believe these coupling techniques may be of value in other settings to reduce initial bit costs.

Acknowledgements

We thank Yann Dubois and Guodong Zhang for feedback on an early draft. We thank Roger Grosse and David Duvenaud for thoughtful comments. Arnaud Doucet acknowledges support from EPSRC grants EP/R018561/1 and EP/R034710/1. This research is supported by the Vector Scholarship in Artificial Intelligence. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

References

Andrieu, C., Doucet, A., and Holenstein, R. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

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

Bérard, J., Del Moral, P., and Doucet, A. A lognormal central limit theorem for particle approximations of normalizing constants. *Electronic Journal of Probability*, 19, 2014.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-Milne, S. JAX: Composable transformations of Python+NumPy programs.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Caterini, A. L., Doucet, A., and Sejdinovic, D. Hamiltonian variational auto-encoder. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8178–8188, 2018.

Cérou, F., Del Moral, P., and Guyader, A. A nonasymptotic theorem for unnormalized Feynman-Kac particle models. In *Annales de l'IHP Probabilités et statistiques*, volume 47, pp. 629–649, 2011.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. *Advances in Neural Information Processing Systems*, 28:2980–2988, 2015.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley Series in Telecommunications. 1991.

Cremer, C., Morris, Q., and Duvenaud, D. Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*, 2017.

Del Moral, P. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

Devroye, L. Nonuniform random variate generation. *Handbooks in operations research and management science*, 13:83–121, 2006.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Domke, J. and Sheldon, D. R. Importance weighting and variational inference. In *Advances in Neural Information Processing Systems*, pp. 4470–4479, 2018.

Doucet, A., De Freitas, N., and Gordon, N. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice*, pp. 3–14. Springer, 2001.

Duda, J. Asymmetric numeral systems. *arXiv preprint arXiv:0902.0271*, 2009.

Finke, A. *On extended state-space constructions for Monte Carlo methods*. PhD thesis, University of Warwick, 2015.

Flamich, G., Havasi, M., and Hernández-Lobato, J. M. Compressing images by encoding their latent representations with relative entropy coding. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16131–16141. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/ba053350fe56ed93e64b3e769062b680-Paper.pdf.

Frey, B. J. *Graphical Models for Machine Learning and Digital Communications*. MIT Press, 1998.

Frey, B. J. and Hinton, G. E. Efficient stochastic source coding and an application to a Bayesian network source model. *The Computer Journal*, 40(2_and_3):157–165, 1997.

Havasi, M., Peharz, R., and Hernández-Lobato, J. M. Minimal random code learning: Getting bits back from compressed model parameters. In *International Conference on Learning Representations*, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Hinton, G. E. and Van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational Learning Theory*, pp. 5–13, 1993.

Hoogeboom, E., Peters, J., van den Berg, R., and Welling, M. Integer discrete flows and lossless compression. In *Advances in Neural Information Processing Systems*, pp. 12134–12144, 2019.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with Gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Johnston, N., Eban, E., Gordon, A., and Ballé, J. Computationally efficient neural image compression. *arXiv preprint arXiv:1912.08771*, 2019.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

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

Kingma, F. H., Abbeel, P., and Ho, J. Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables. *arXiv preprint arXiv:1905.06845*, 2019.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. Auto-encoding sequential Monte Carlo. In *International Conference on Learning Representations*, 2018.

MacKay, D. J. C. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pp. 6573–6583, 2017.

Minnen, D., Ballé, J., and Toderici, G. D. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pp. 10771–10780, 2018.

Naesseth, C., Linderman, S., Ranganath, R., and Blei, D. Variational sequential Monte Carlo. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 968–977. PMLR, 09–11 Apr 2018. URL http://proceedings.mlr.press/v84/naesseth18a.html.

Neal, R. M. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Oord, A. v. d., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.

Radford, A., Jong Wook, K., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Kreuger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2020.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pp. 14866–14876, 2019.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Rippel, O. and Adams, R. P. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.

Salakhutdinov, R. and Murray, I. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th International Conference on Machine learning*, pp. 872–879, 2008.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *ICLR*, 2017.

Sønderby, C. K., Poole, B., and Mnih, A. Continuous relaxation training of discrete latent variable image models. In *Beysian DeepLearning workshop, NIPS*, volume 201, 2017.

Townsend, J. A tutorial on the range variant of asymmetric numeral systems, 2020.

Townsend, J., Bird, T., and Barber, D. Practical lossless compression with latent variables using bits back coding. *ICLR*, 2019.

Townsend, J., Bird, T., Kunze, J., and Barber, D. Hilloc: Lossless image compression with hierarchical latent variable models. *ICLR*, 2020.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020.

van den Berg, R., Gritsenko, A. A., Dehghani, M., Kaae Sønderby, C., and Salimans, T. Idf++: Analyzing and improving integer discrete flows for lossless compression. *arXiv e-prints*, pp. arXiv–2006, 2020.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. *arXiv e-prints*, art. arXiv:1609.03499, September 2016.

Walker, A. J. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10(8):127–128, 1974.

Wallace, S., C. Classification by minimum-message-length inference. In *Advances in Computing and Information*, pp. 72–81, 1990.

Yang, Y., Bamler, R., and Mandt, S. Improving inference for neural image compression. *arXiv preprint arXiv:2006.04240*, 2020.