Improving Lossless Compression Rates via Monte Carlo Bits-Back Coding

VECTOR INSTITUTE
UNIVERSITY OF TORONTO
FACEBOOK AI
UNIVERSITY OF OXFORD
UCL

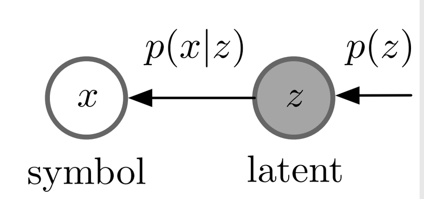
Yangjun Ruan* [1,2] Karen Ullrich* [2,3] Daniel Severo* [1,2] James Townsend [4]
Ashish Khisti [1] Arnaud Doucet [5] Alireza Makhzani [1,2] Chris J. Maddison [1,2]

[1]University of Toronto [2]Vector Institute [3]Facebook AI Research [4]University College London [5]University of Oxford

Overview

Latent variable models

- evaluating $p(x)$ is intractable for **lossless compression**
- jointly encoding (x, z) is wasteful!

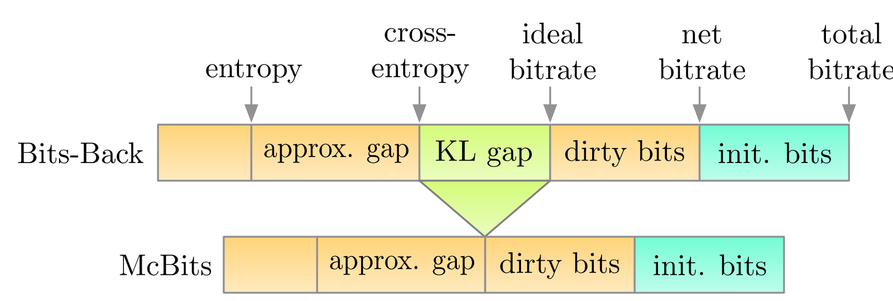


Bits-back coding

☺ applies latent variable models for lossless compression
☺ achieves a better bitrate $= -$ELBO
☹ suffers from a KL gap to the cross-entropy

Monte Carlo bits-back coding

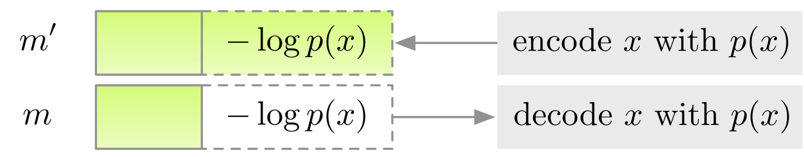
✔ removes the KL gap
✔ little additional cost
✔ better for transfer compression



Background: Bits-Back is Suboptimal

Asymmetric Numeral Systems (ANS)

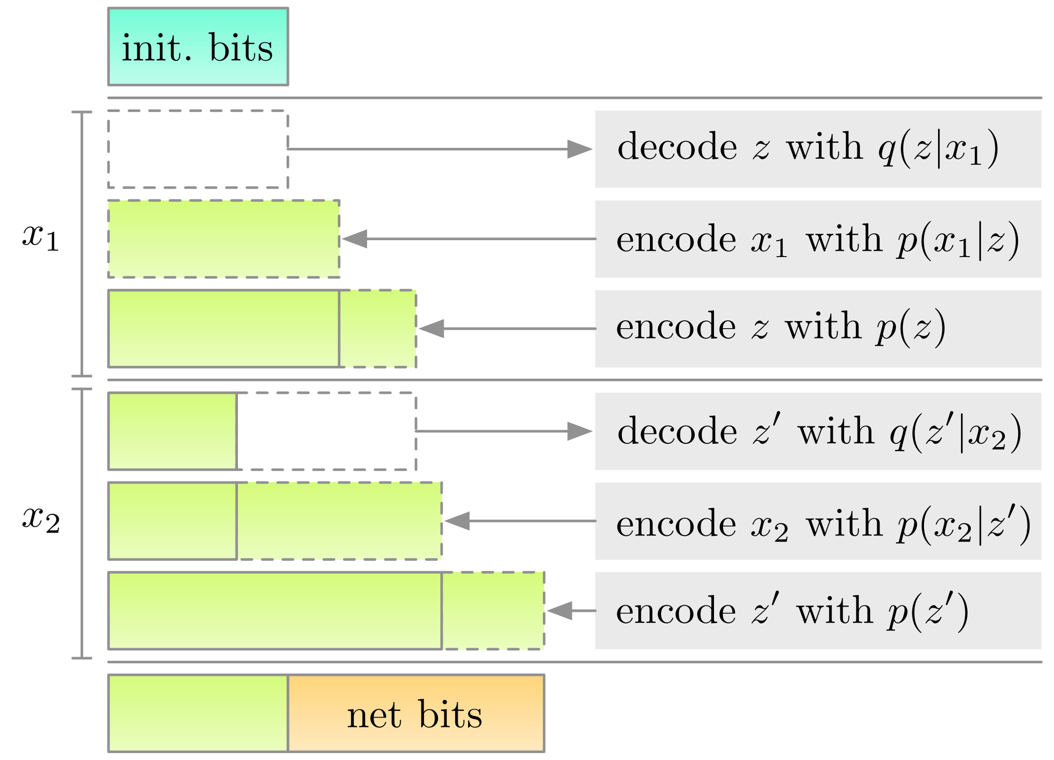
- stack-like (LIFO) entropy coder
- ANS message is a store of randomness



trick: use decode as sample!

Bits-Back with ANS (BB-ANS)

- uses an *approximate posterior* $q(z \mid x)$
- decodes z with $q(z \mid x)$ from intermediate messages



☺ saves $-\log q(z|x)$ bits/sym $\Rightarrow$ net bitrate $= -$ELBO
☹ needs $-\log q(z \mid x)$ initial bits for the *first* symbol
☹ ELBO may be a *loose* bound on $\log p(x) \Rightarrow$ KL gap

From Tighter Variational Bounds to Bits-Back Schemes

Given an unbiased Monte Carlo estimator of the marginal likelihood $\hat{p}_N(x)$

- **Goal:** derive bits-back schemes with a net bitrate $= -\mathbb{E}[\log \hat{p}_N(x)]$
- **Key idea:** exploit the extended latent space representations of $\hat{p}_N(x)$

$$\hat{p}_N(x) = \frac{P(x, \mathcal{Z})}{Q(\mathcal{Z} \mid x)} \begin{array}{l} \rightarrow \text{target distribution} \\ \rightarrow \text{proposal distribution} \end{array}$$

where the extended latents $\mathcal{Z} \sim Q(\mathcal{Z} \mid x)$

✔ Derive McBits coders as with BB-ELBO

Procedure *Encode*(*sym* x, *msg* m)
  decode $\mathcal{Z}$ with $Q(\mathcal{Z} \mid x)$
  encode x and $\mathcal{Z}$ with $P(x, \mathcal{Z})$
  return m'

Instantiations

- ☐ BB-IS (Importance Sampling)
- ☐ BB-SMC (Sequential Monte Carlo)
- ☐ BB-AIS (Annealed Importance Sampling)

Simplest Example: Decode Multiple Particles and Pick One

Importance sampling (IS)

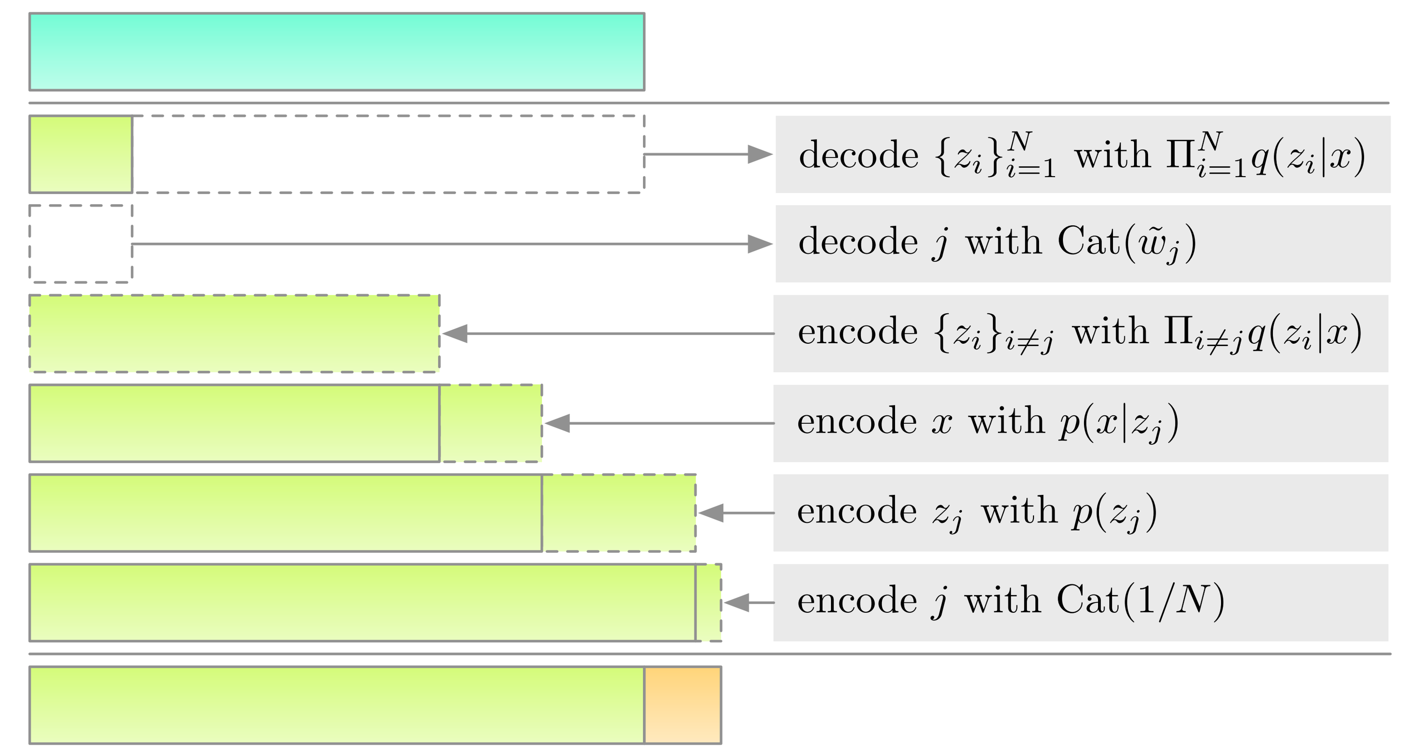
- samples N particles $z_i \sim q(z_i \mid x)$ i.i.d. and averages the importance weights

$$\hat{p}_N(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{p(x, z_i)}{q(z_i \mid x)}$$

- the variational bound (IWAE) converges monotonically to $\log p(x)$ as $N \to \infty$

BB-IS

✔ $\mathcal{Z}$ includes N particles $\{z_i\}_{i=1}^N$ and a particle index $j \in \{1 \,..\, N\}$



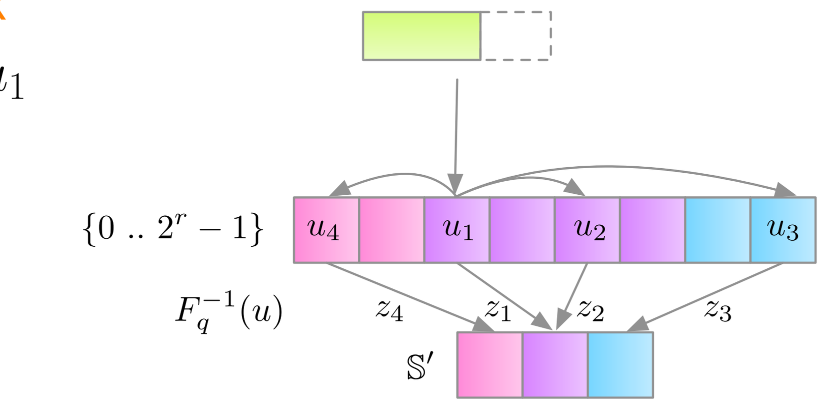
☺ achieves a net bitrate $= -$IWAE
☺ asymptotically reaches the cross-entropy
☹ requires $\mathcal{O}(N)$ initial bits $\Leftarrow \mathcal{O}(N)$ decoded latent variables

Coupling Technique for Reducing Initial Bit Cost

Key idea: coupling the particles $\{z_i\}_{i=1}^N$ by a shared random number

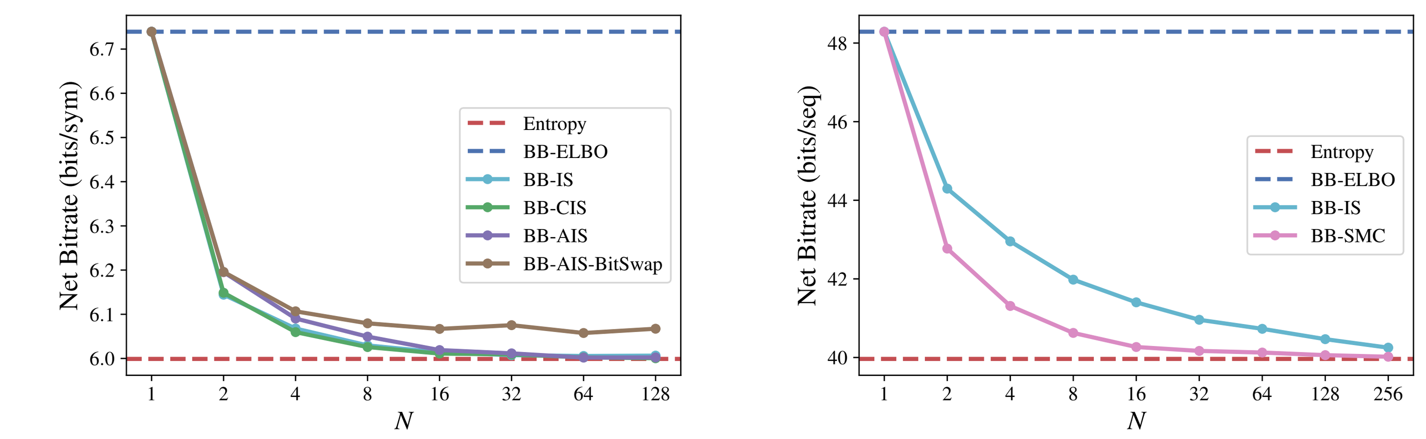
✔ discrete analog of the inverse CDF trick
✔ reparamerize $\{z_i\}_{i=1}^N$ by a uniform r.v. u_1

☺ decoding a single uniform is enough!
☺ reduce the initial bit cost to $\mathcal{O}(\log N)$ (actually $\mathcal{O}(1)$ in practice)

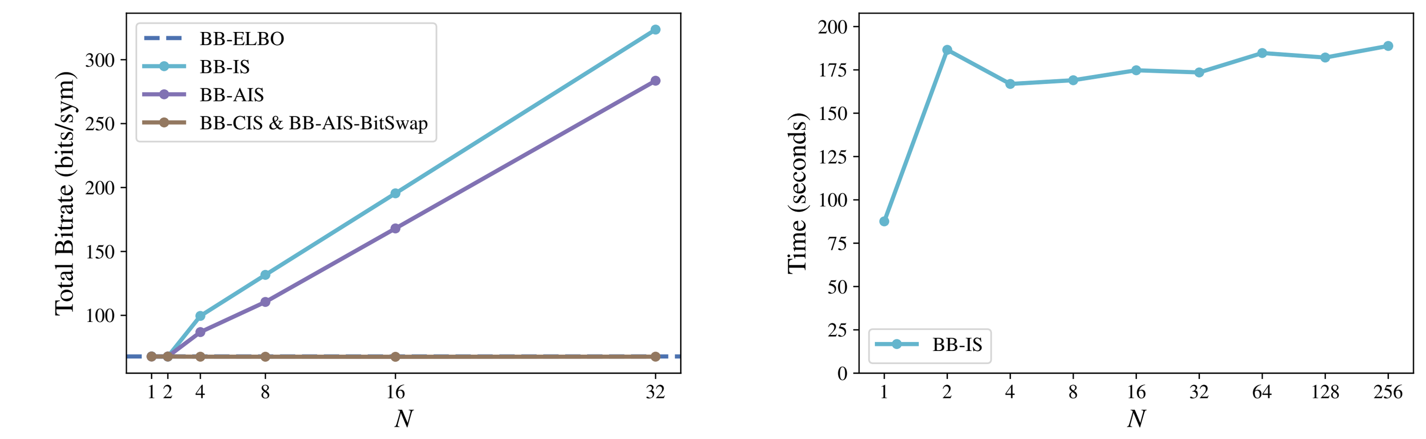


Experiments

Convergence: the net bitrates of McBits coders converge to the entropy on both toy mixture model (left) and toy hidden markov model (right)



Additional cost: nearly $\mathcal{O}(1)$ initial bit cost achieved by coupling (left); sublinear computational cost with parallelization over particles (right)



OOD performance: greater bitrate savings in out-of-distribution compression

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

More extensive evaluation and analysis are in our paper!