Bit-Swap
Recursive Bits-Back Coding for Lossless Compression with Hierarchical Latent Variables

Friso Kingma, Pieter Abbeel, Jonathan Ho
Likelihood-based
Deep Generative Models
(VAE’s, Flows, AR-models, etc.)
Likelihood-based
Deep Generative Models
(VAE’s, Flows, AR-models, etc.)

Images
Likelihood-based Deep Generative Models (VAE’s, Flows, AR-models, etc.)
Likelihood-based
Deep Generative Models
(VAE’s, Flows, AR-models, etc.)

Images
Videos
Text
Likelihood-based Deep Generative Models (VAE’s, Flows, AR-models, etc.)
Likelihood-based Deep Generative Models (VAE’s, Flows, AR-models, etc.)

Images ➔ Videos ➔ Text ➔ Audio

Lossless compression
Hierarchical Latent Variable Models
How do we use Hierarchical Latent Variable Models for efficient compression?
First:
We are using a **Entropy coder** called
Asymmetric Numeral Systems (ANS)

\[ \mathbb{E}[-\log p(x)] \]

**NOT** the same as inference model of VAE

*Duda et al. (2015)*
First:
We are using a **Entropy coder** called
Asymmetric Numeral Systems (ANS)

\[ \mathbb{E} [ - \log p(x) ] \]

\[ p(x) \]

*NOT* the same as generative model of VAE

*Duda et al. (2015)*
How do we compress with a “regular” Latent Variable Model?

**Bits-Back Coding**

Sender

```
encode
```

```
bitstream
```

Receiver

```
decode
```

```
```

*Townsend et al. (2019)*
How do we compress with a “regular” Latent Variable Model?

**Bits-Back Coding**

Sender

\[ \text{bitstream} \]

\[ \text{decode } z \text{ with } q_\theta(z|x) \]

Bitrate \(= \log q_\theta(z|x)\)

*Townsend et al. (2019)*
How do we compress with a “regular” Latent Variable Model?

**Bits-Back Coding**

 Sender

\[ \text{Bitrate} = \log q_\theta(z|x) - \log p_\theta(x|z) \]

*Townsend et al. (2019)*
How do we compress with a “regular” Latent Variable Model?

Bits-Back Coding

Sender

\[ \text{Bitrate} = \log q_\theta(z|x) - \log p_\theta(x|z) - \log p(z) \]

Townsend et al. (2019)
How do we compress with a “regular” Latent Variable Model?

**Bits-Back Coding**

Sender

\[
\text{Bitrate} = \mathbb{E}_{q_{\theta}(z|x)} \left[ \log q_{\theta}(z|x) - \log p_{\theta}(x|z) - \log p(z) \right] = -\mathcal{L}(\theta) \quad (\text{ELBO})
\]

Optimizing model for ELBO

= Directly optimizing for bitrates
How do we compress with a “regular” Latent Variable Model?

Bits-Back Coding

Sender

- encode $z$ with $p(z)$
- encode $x$ with $p_\theta(x|z)$
- decode $z$ with $q_\theta(z|x)$

bitstream

Receiver

- bitstream

- Operations in reverse order
- With encode and decode operations switched

Townsend et al. (2019)
How do we compress with a "regular" Latent Variable Model?

**Bits-Back Coding**

**Sender**

- encode \( z \) with \( p(z) \)
- encode \( x \) with \( p_\theta(x|z) \)
- decode \( z \) with \( q_\theta(z|x) \)

**Receiver**

- decode \( z \) with \( p(z) \)

---

*Townsend et al. (2019)*
How do we compress with a “regular” Latent Variable Model?

**Bits-Back Coding**

**Sender**

- Bitstream
- Decode $z$ with $q_θ(z|x)$
- Encode $x$ with $p_θ(x|z)$
- Encode $z$ with $p(z)$

**Receiver**

- Bitstream
- Decode $z$ with $p(z)$
- Decode $x$ with $p_θ(x|z)$

*Townsend et al. (2019)*
How do we compress with a "regular" Latent Variable Model?

**Bits-Back Coding**

**Sender**

- bitstream
- decode $z$ with $q_\theta(z|x)$
- encode $x$ with $p_\theta(x|z)$
- encode $z$ with $p(z)$

**Receiver**

- bitstream
- decode $z$ with $p(z)$
- decode $x$ with $p_\theta(x|z)$
- encode $z$ with $q_\theta(z|x)$

*Townsend et al. (2019)*
How do we compress with a “regular” Latent Variable Model?

**Bits-Back Coding**

**Sender**

- encode $z$ with $p(z)$
- encode $x$ with $p_\theta(x|z)$
- decode $z$ with $q_\theta(z|x)$

**Receiver**

- decode $z$ with $p(z)$
- decode $x$ with $p_\theta(x|z)$
- encode $z$ with $q_\theta(z|x)$

Initial bits are “back”
How do we compress with a **Hierarchical** Latent Variable Model?  
First: **what is the model structure?**

Can be thought of as multiple **nested** latent variable models.
How do we compress with a **Hierarchical** Latent Variable Model?

First: what is the model structure?

Can be thought of as multiple **nested** latent variable models.
How do we compress with a **Hierarchical** Latent Variable Model?  
First: **what is the model structure?**

Can be thought of as multiple **nested** latent variable models.
How do we compress with a **Hierarchical** Latent Variable Model?
First: **what is the model structure?**

Can be thought of as multiple **nested** latent variable models
How do we compress with a **Hierarchical** Latent Variable Model?

First: **what is the model structure?**

Can be thought of as multiple **nested** latent variable models
How do we compress with a **Hierarchical** Latent Variable Model?

First: **what is the model structure?**

Can be thought of as multiple **nested** latent variable models.
What if we naively apply Bits-Back Coding?

Initial bits ($L = 1$): $- \log q_\theta(z|x)$
What if we **naively** apply Bits-Back Coding?

Initial bits \((L = 1)\): 
\[- \log q_{\theta}(z|x)\]

Initial bits \((L > 1)\): 
\[
\sum_{i=0}^{L-1} - \log q_{\theta}(z_{i+1}|z_i)
\]

Initial bits **grows linearly**!
Question: How do we efficiently compress with Hierarchical Latent Variable Models?

Solution:
- Actually treat the model as multiple nested latent variable models
- apply Bits-Back Coding recursively on every layer
Compression with Hierarchical Latent Variable Models:

**Bit-Swap** (ours)

**Sender**

- initial bitstream

**Corresponding step**

**Bits-Back Coding**

- decode $z$ with $q(z|x)$
- encode $x$ with $p(x|z)$
- encode $z$ with $p(z)$
Compression with Hierarchical Latent Variable Models: 

**Bit-Swap** (ours)

Sender

- initial bitstream

- decode $z_1$ with $q_\theta(z_1|x)$

Corresponding step

**Bits-Back Coding**

- decode $z$ with $q(z|x)$

- encode $x$ with $p(x|z)$

- encode $z$ with $p(z)$

$$= \log q_\theta(z_1|x)$$
Compression with Hierarchical Latent Variable Models: **Bit-Swap** (ours)

Sender

- initial bitstream
- decode $z_1$ with $q_\theta(z_1|x)$
- encode $x$ with $p_\theta(x|z_1)$

Corresponding step

**Bits-Back Coding**

- decode $z$ with $q(z|x)$
- encode $x$ with $p(x|z)$
- encode $z$ with $p(z)$

\[
= \log q_\theta(z_1|x) - \log p_\theta(x|z_1)
\]
Compression with Hierarchical Latent Variable Models:

**Bit-Swap (ours)**

Sender

- Initial bitstream
- Decode $z_1$ with $q_\theta(z_1|x)$
- Encode $x$ with $p_\theta(x|z_1)$

Corresponding step

**Bits-Back Coding**

- Decode $z$ with $q(z|x)$
- Encode $x$ with $p(x|z)$
- Encode $z$ with $p(z)$

$$= \log q_\theta(z_1|x) - \log p_\theta(x|z_1) - \log p(z_1)$$
Compression with Hierarchical Latent Variable Models:

**Bit-Swap** (ours)

Sender

- Initial bitstream
- Decode $z_1$ with $q_\theta(z_1|x)$
- Encode $x$ with $p_\theta(x|z_1)$
- Decode $z_2$ with $q_\theta(z_2|z_1)$

Corresponding step

Bits-Back Coding

- Decode $z$ with $q(z|x)$
- Encode $x$ with $p(x|z)$
- Encode $z$ with $p(z)$

\[
= \log q_\theta(z_1|x) - \log p_\theta(x|z_1) + \log q_\theta(z_2|z_1)
\]
Compression with Hierarchical Latent Variable Models: 

**Bit-Swap** (ours)

\[
\begin{align*}
\text{Sender} & \quad \text{Corresponding step} \\
\text{initial bitstream} & \quad \text{Bits-Back Coding} \\
\text{decode } z_1 \text{ with } q_\theta(z_1|x) & \quad \text{decode } z \text{ with } q(z|x) \\
\text{encode } x \text{ with } p_\theta(x|z_1) & \quad \text{encode } x \text{ with } p(x|z) \\
\text{decode } z_2 \text{ with } q_\theta(z_2|z_1) & \quad \text{encode } z \text{ with } p(z) \\
\text{encode } z_1 \text{ with } p_\theta(z_1|z_2) &
\end{align*}
\]

\[
= \log q_\theta(z_1|x) - \log p_\theta(x|z_1) + \log q_\theta(z_2|z_1) - \log p_\theta(z_1|z_2)
\]
Compression with Hierarchical Latent Variable Models: **Bit-Swap** (ours)

**Send**er

- Initial bitstream
- Decode $z_1$ with $q_\theta(z_1|x)$
- Encode $x$ with $p_\theta(x|z_1)$
- Decode $z_2$ with $q_\theta(z_2|z_1)$
- Encode $z_1$ with $p_\theta(z_1|z_2)$
- Decode $z_3$ with $q_\theta(z_3|z_2)$

**Corresponding step**

**Bits-Back Coding**

- Decode $z$ with $q(z|x)$
- Encode $x$ with $p(x|z)$
- Encode $z$ with $p(z)$

$$= \log q_\theta(z_{1:2}|x) - \log p_\theta(x, z_1, z_2|z_3)$$
Compression with Hierarchical Latent Variable Models: **Bit-Swap** (ours)

Sender

- initial bitstream
- decode $z_1$ with $q_\theta(z_1|x)$
- encode $x$ with $p_\theta(x|z_1)$
- decode $z_2$ with $q_\theta(z_2|z_1)$
- encode $z_1$ with $p_\theta(z_1|z_2)$
- decode $z_3$ with $q_\theta(z_3|z_2)$
- encode $z_2$ with $p_\theta(z_2|z_3)$

$$= \log q_\theta(z_{1:3}|x) - \log p_\theta(x, z_1, z_2|z_3)$$

Corresponding step

**Bits-Back Coding**

- decode $z$ with $q(z|x)$
- encode $x$ with $p(x|z)$
- encode $z$ with $p(z)$
Compression with Hierarchical Latent Variable Models: **Bit-Swap** (ours)

\[
\text{Sender} \\
\text{initial bitstream} \\
\downarrow \\
\text{decode } z_1 \text{ with } q_\theta(z_1|x) \\
\downarrow \\
\text{encode } x \text{ with } p_\theta(x|z_1) \\
\downarrow \\
\text{decode } z_2 \text{ with } q_\theta(z_2|z_1) \\
\downarrow \\
\text{encode } z_1 \text{ with } p_\theta(z_1|z_2) \\
\downarrow \\
\text{decode } z_3 \text{ with } q_\theta(z_3|z_2) \\
\downarrow \\
\text{encode } z_2 \text{ with } p_\theta(z_2|z_3) \\
\downarrow \\
\text{encode } z_3 \text{ with } p(z_3)
\]

\[
= \log q_\theta(z_{1:3}|x) - \log p_\theta(x, z_1, z_2|z_3) - \log p(z_3)
\]
Compression with Hierarchical Latent Variable Models: \textbf{Bit-Swap} (ours)

\begin{align*}
\text{Sender} & \quad \text{Receiver} \\
\text{initial bitstream} & \\
& \quad \text{decode } z_1 \text{ with } q_\theta(z_1|x) \\
& \quad \text{encode } x \text{ with } p_\alpha(x|z_1) \\
& \quad \text{decode } z_2 \text{ with } q_\theta(z_2|z_1) \\
& \quad \text{encode } z_1 \text{ with } p_\alpha(z_1|z_2) \\
& \quad \text{decode } z_3 \text{ with } q_\theta(z_3|z_2) \\
& \quad \text{encode } z_2 \text{ with } p_\alpha(z_2|z_3) \\
& \quad \text{encode } z_3 \text{ with } p(z_3) \\
= \log q_\theta(z_{1:3}|x) - \log p_\theta(x, z_{1:3})
\end{align*}
Compression with Hierarchical Latent Variable Models: 

**Bit-Swap (ours)**

\[
\begin{align*}
\text{Sender} & \quad \text{Receiver} \\
\text{initial bitstream} & \\
\text{decode } z_1 \text{ with } q_\theta(z_1|x) & \quad \text{encode } x \text{ with } p_\theta(x|z_1) \\
\text{encode } x \text{ with } p_\theta(x|z_1) & \quad \text{decode } z_2 \text{ with } q_\theta(z_2|z_1) \\
\text{decode } z_2 \text{ with } q_\theta(z_2|z_1) & \quad \text{encode } z_1 \text{ with } p_\theta(z_1|z_2) \\
\text{encode } z_1 \text{ with } p_\theta(z_1|z_2) & \quad \text{decode } z_3 \text{ with } q_\theta(z_3|z_2) \\
\text{decode } z_3 \text{ with } q_\theta(z_3|z_2) & \quad \text{encode } z_2 \text{ with } p_\theta(z_2|z_3) \\
\text{encode } z_2 \text{ with } p_\theta(z_2|z_3) & \quad \text{encode } z_3 \text{ with } p(z_3) \\
\end{align*}
\]

\[
\begin{align*}
&= \mathbb{E}_{q_\theta(z_{1:3}|x)} \left[ \log q_\theta(z_{1:3}|x) - \log p_\theta(x, z_{1:3}) \right] \\
&= -\mathcal{L}(\theta) \text{ (ELBO)}
\end{align*}
\]
Compression with Hierarchical Latent Variable Models: **Bit-Swap** (ours)

**Sender**
- initial bitstream
- decode $z_1$ with $q_θ(z_1|x)$
- encode $x$ with $p_θ(x|z_1)$
- decode $z_2$ with $q_θ(z_2|z_1)$
- encode $z_1$ with $p_θ(z_1|z_2)$
- decode $z_3$ with $q_θ(z_3|z_2)$
- encode $z_2$ with $p_θ(z_2|z_3)$
- encode $z_3$ with $p(z_3)$

**Receiver**
- bitstream
- decode $z_3$ with $q_θ(z_3|z_2)$
- encode $z_2$ with $p_θ(z_2|z_3)$
- decode $z_2$ with $q_θ(z_2|z_1)$
- encode $z_1$ with $p_θ(z_1|z_2)$
- decode $z_1$ with $q_θ(z_1|x)$
- initial bitstream

- Operations in reverse order
- With encode and decode operations switched
Compression with Hierarchical Latent Variable Models: **Bit-Swap** (ours)

- Operations in reverse order
- With encode and decode operations switched
Compression with Hierarchical Latent Variable Models: **Bit-Swap** (ours)

- Still getting initial bits “back”
- Still compressing with bitrate equal to -ELBO
- But now also ....
Bit-Swap initial bits is bounded

\[ N_{\text{init}}^{\text{BitSwap}} \leq \sum_{i=0}^{L-1} \max \left( 0, \log \frac{p_{\theta}(z_{i-1}|z_i)}{q_{\theta}(z_{i+1}|z_i)} \right) \]

Instead of growing linearly
Cumulative average compression rate of compressing 100 images in sequence

- **Bit-Swap**
- **Bits-Back Coding**

2 layers

4 layers

8 layers

MNIST

CIFAR-10

ImageNet (32x32)
Managed to outperform other benchmark compressor bitrates averaged over **100 ImageNet images** (cropped to multiples of 32 pixels on each side)

<table>
<thead>
<tr>
<th>Compression Rate (bits/dim)</th>
<th>Uncompressed</th>
<th>gzip</th>
<th>bzip2</th>
<th>LZMA</th>
<th>PNG</th>
<th>WebP</th>
<th>Bits-Back Coding</th>
<th>Bit-Swap</th>
</tr>
</thead>
<tbody>
<tr>
<td>uncompressed</td>
<td>8.00</td>
<td>5.96</td>
<td>5.07</td>
<td>5.09</td>
<td>4.71</td>
<td>3.66</td>
<td>3.62</td>
<td>3.51</td>
</tr>
</tbody>
</table>
Bit-Swap
Recursive Bits-Back Coding for Lossless Compression with Hierarchical Latent Variables

Friso Kingma, Pieter Abbeel, Jonathan Ho