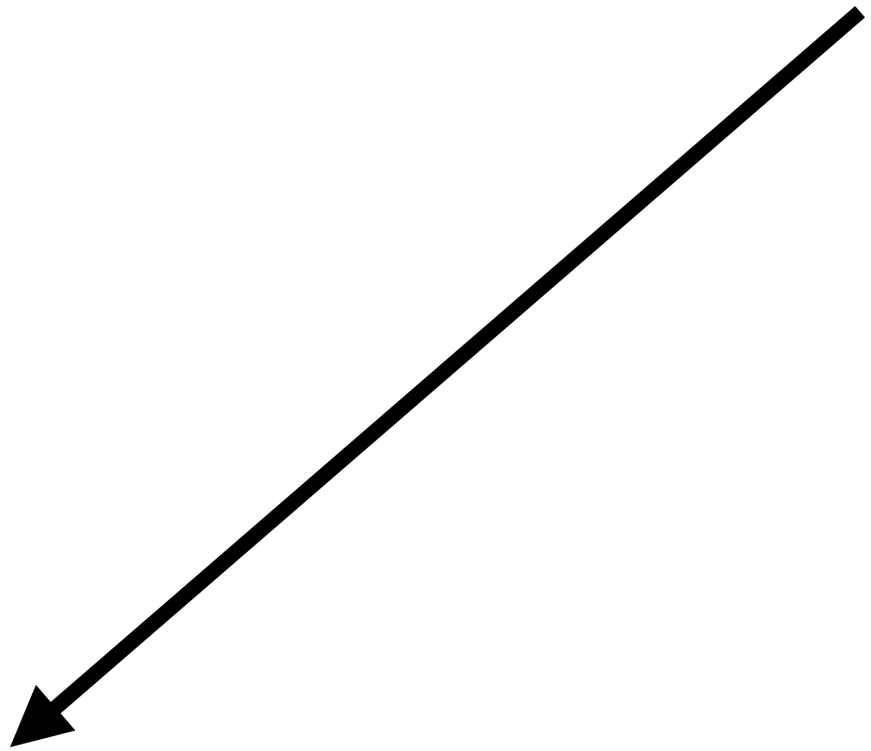
Bit-Swap

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

Friso Kingma, *Pieter Abbeel, Jonathan Ho*

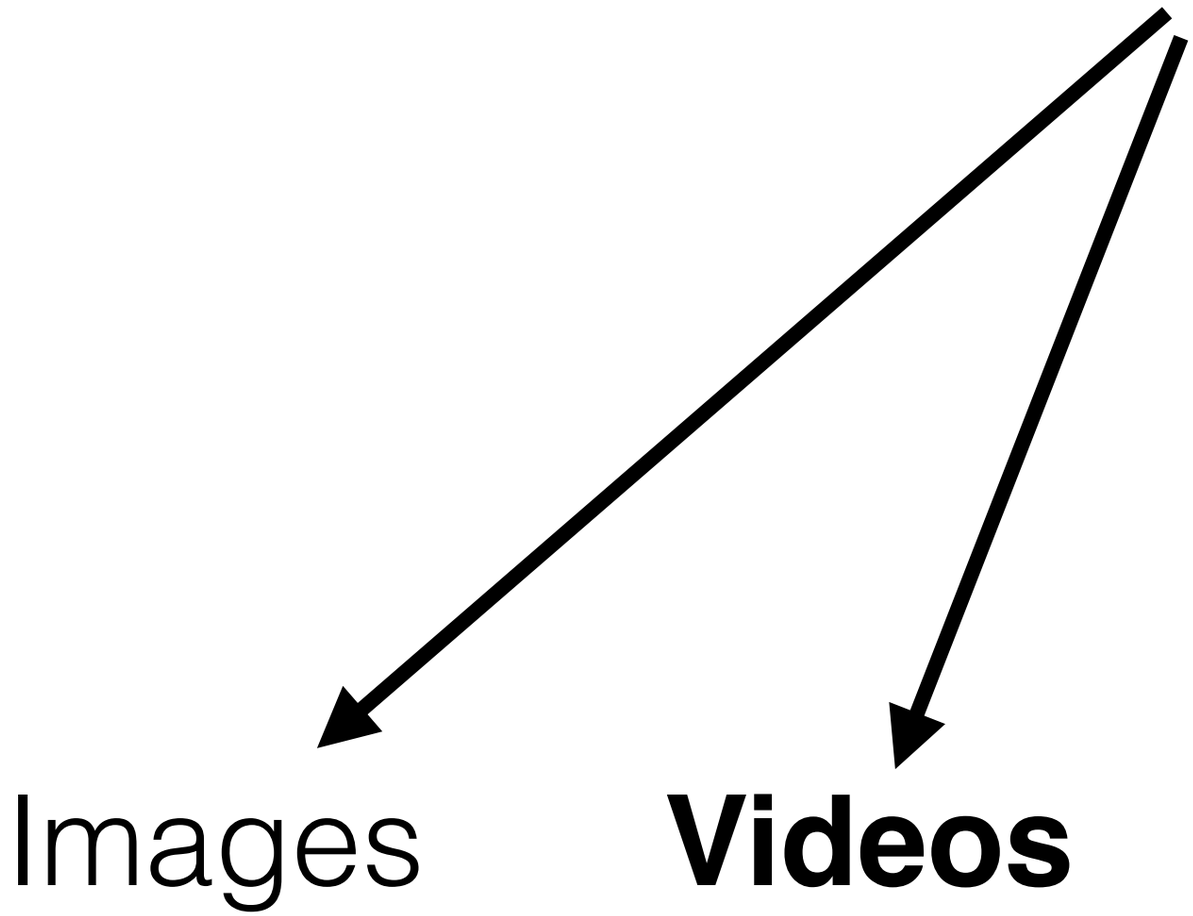
Likelihood-based Deep Generative Models (VAE's, Flows, AR-models, etc.)

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



Images

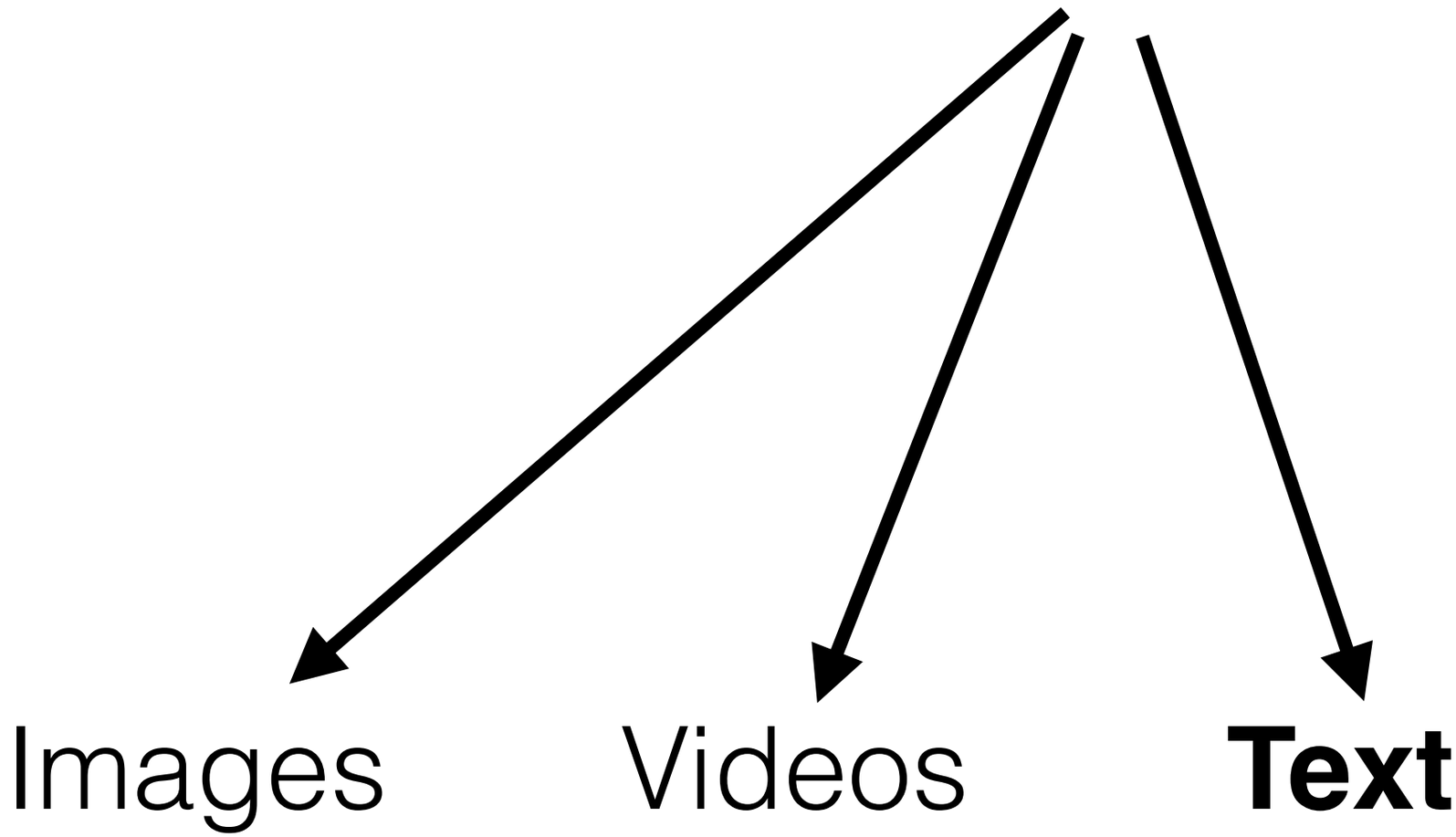
Likelihood-based
Deep Generative Models
(VAE's, Flows, AR-models, etc.)



Images

Videos

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

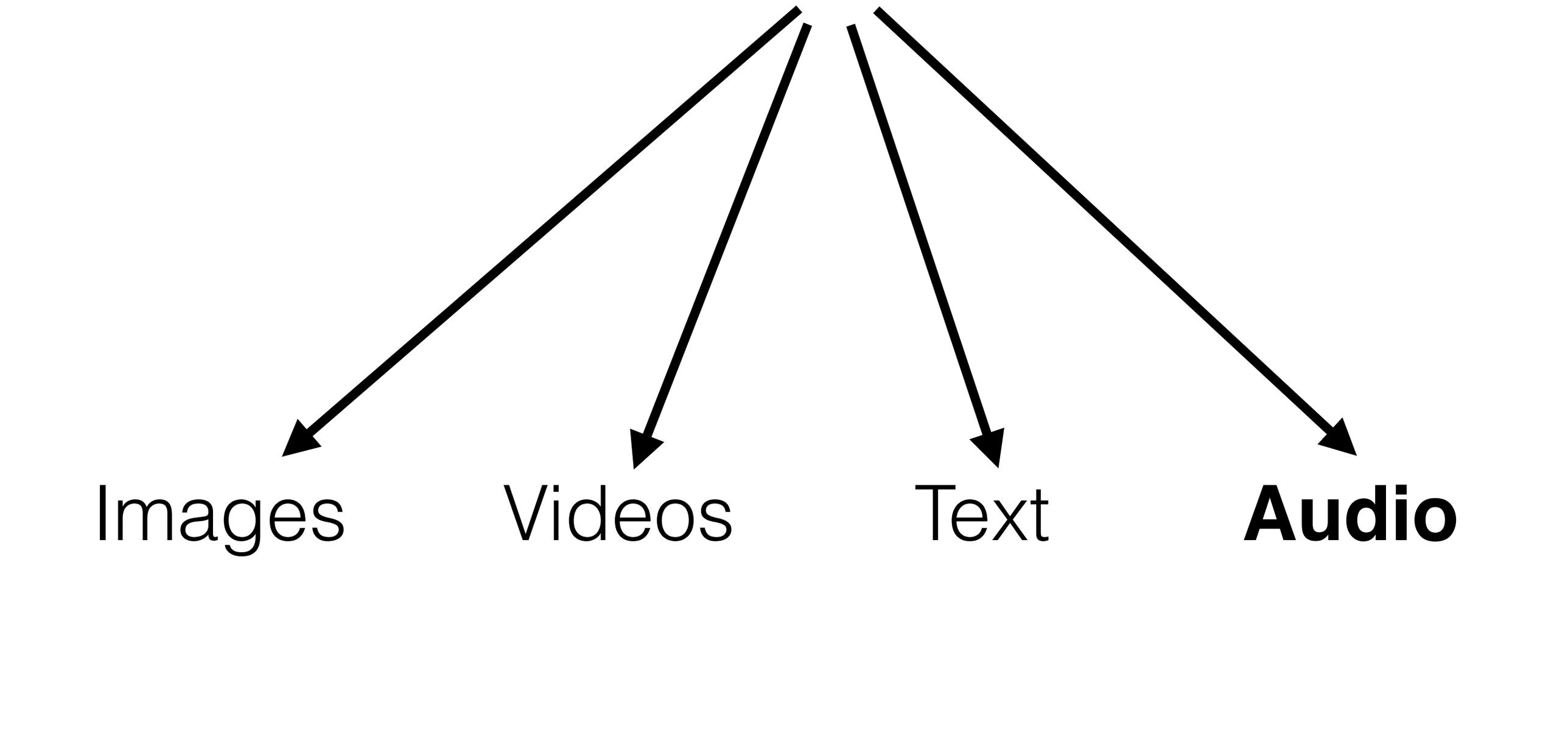


Images

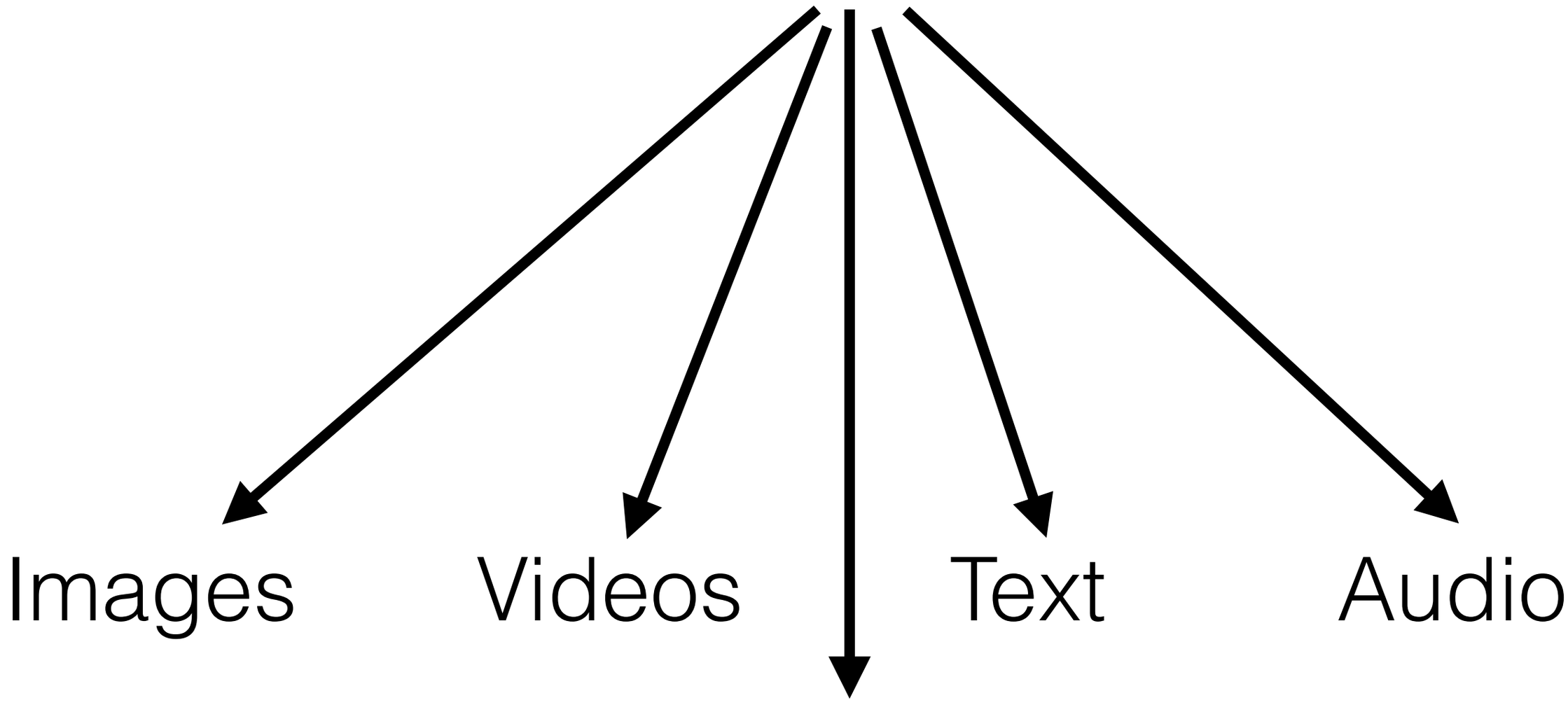
Videos

Text

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



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

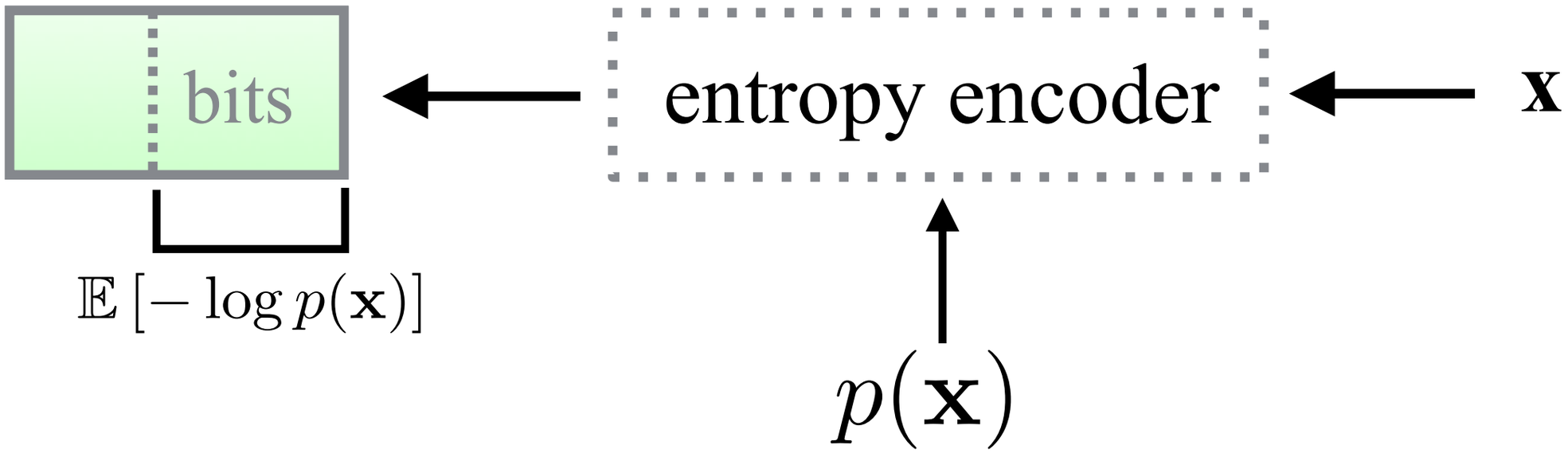


Lossless compression

Hierarchical Latent Variable Models

How do we use Hierarchical Latent Variable Models **for efficient compression?**

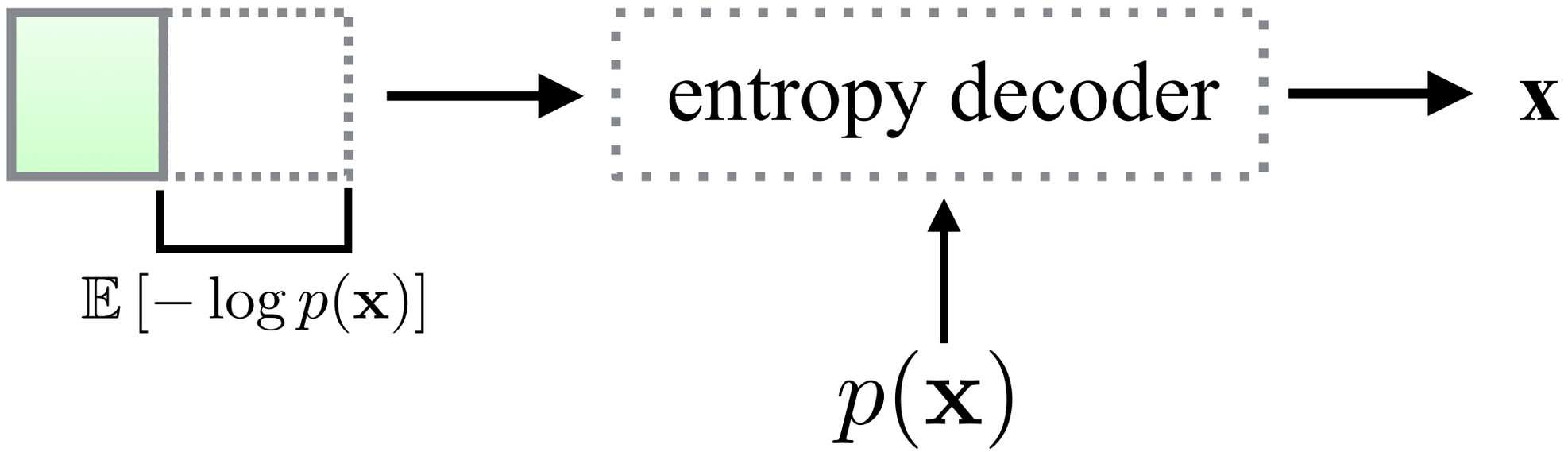
First:
We are using a
Entropy coder called
Asymmetric Numeral Systems (ANS)



NOT the same as inference model of VAE

Duda et al. (2015)

First:
We are using a
Entropy coder called
Asymmetric Numeral Systems (ANS)



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

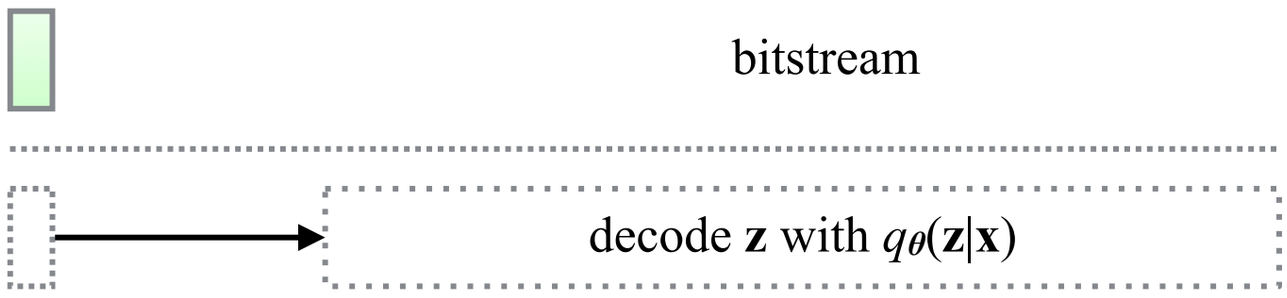
bitstream

Townsend et al. (2019)

How do we compress with a **“regular”** Latent Variable Model?

Bits-Back Coding

Sender

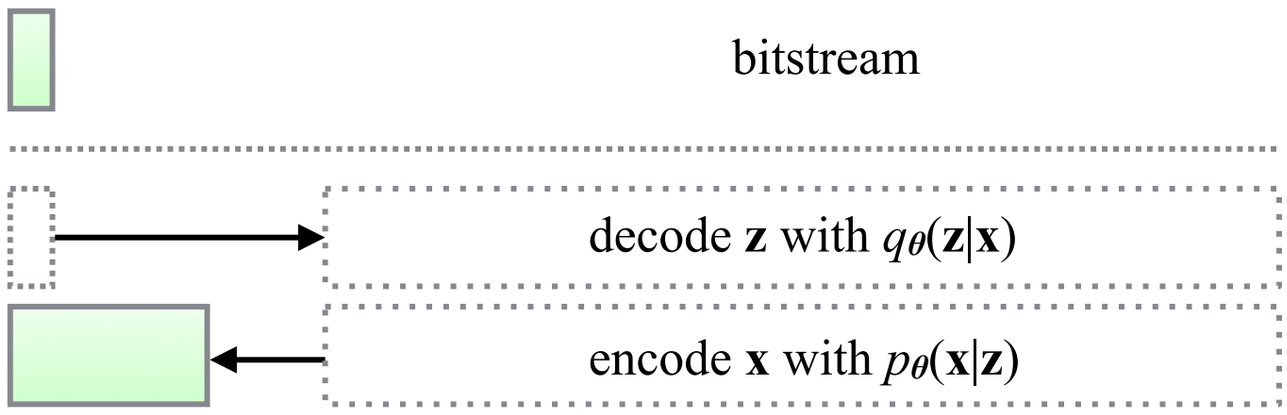


Bitrate $= \log q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$

Townsend et al. (2019)

How do we compress with a **“regular”** Latent Variable Model?
Bits-Back Coding

Sender

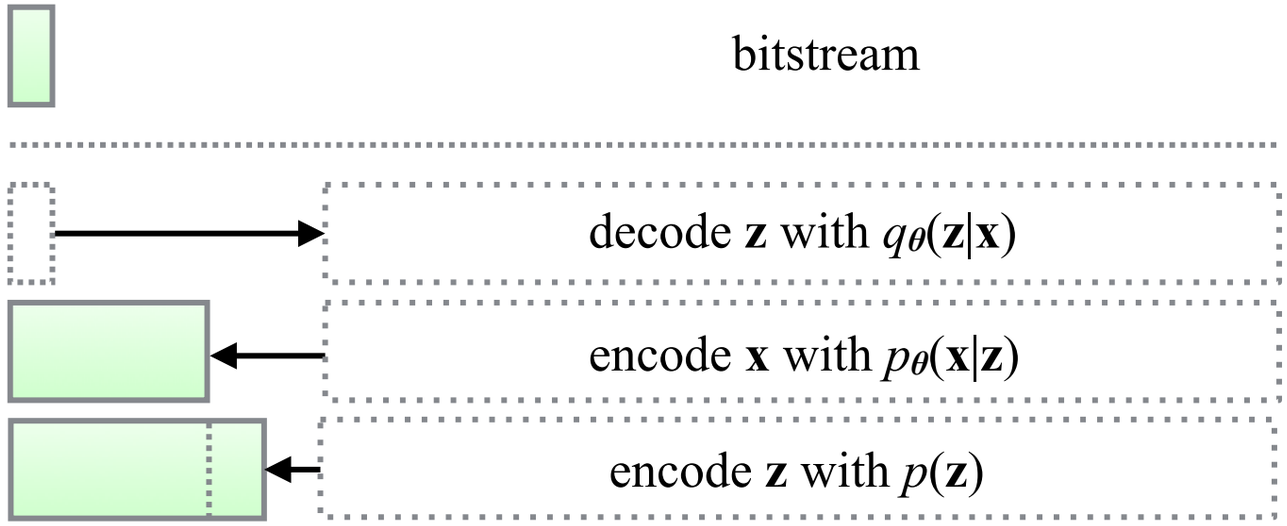


Bitrate $= \log q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$

Townsend et al. (2019)

How do we compress with a **“regular”** Latent Variable Model?
Bits-Back Coding

Sender

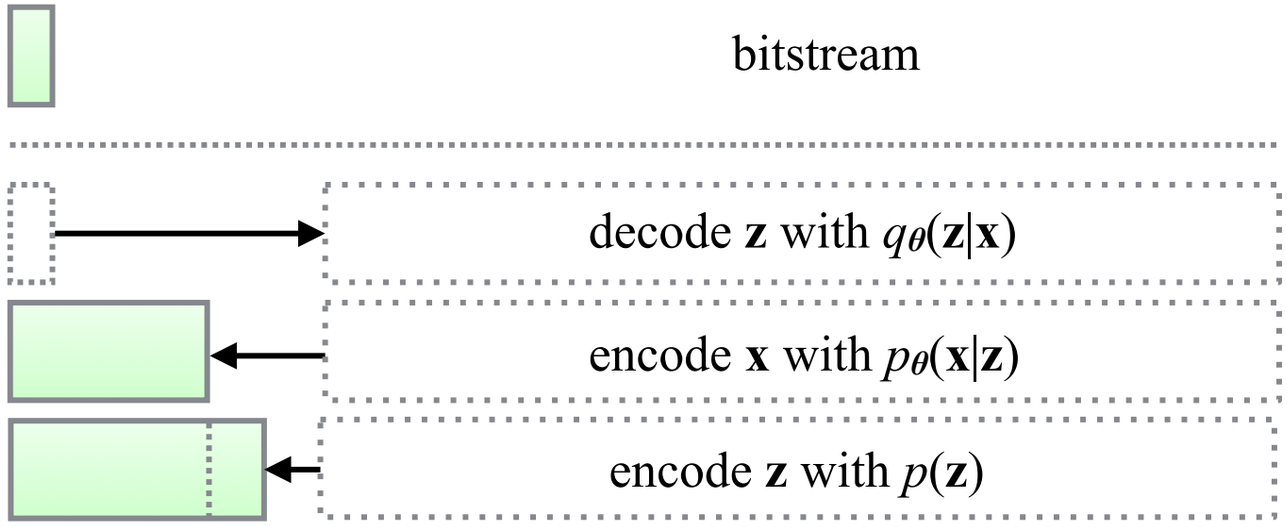


$$\text{Bitrate} = \log q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \log p(\mathbf{z})$$

Townsend et al. (2019)

How do we compress with a **“regular”** Latent Variable Model?
Bits-Back Coding

Sender



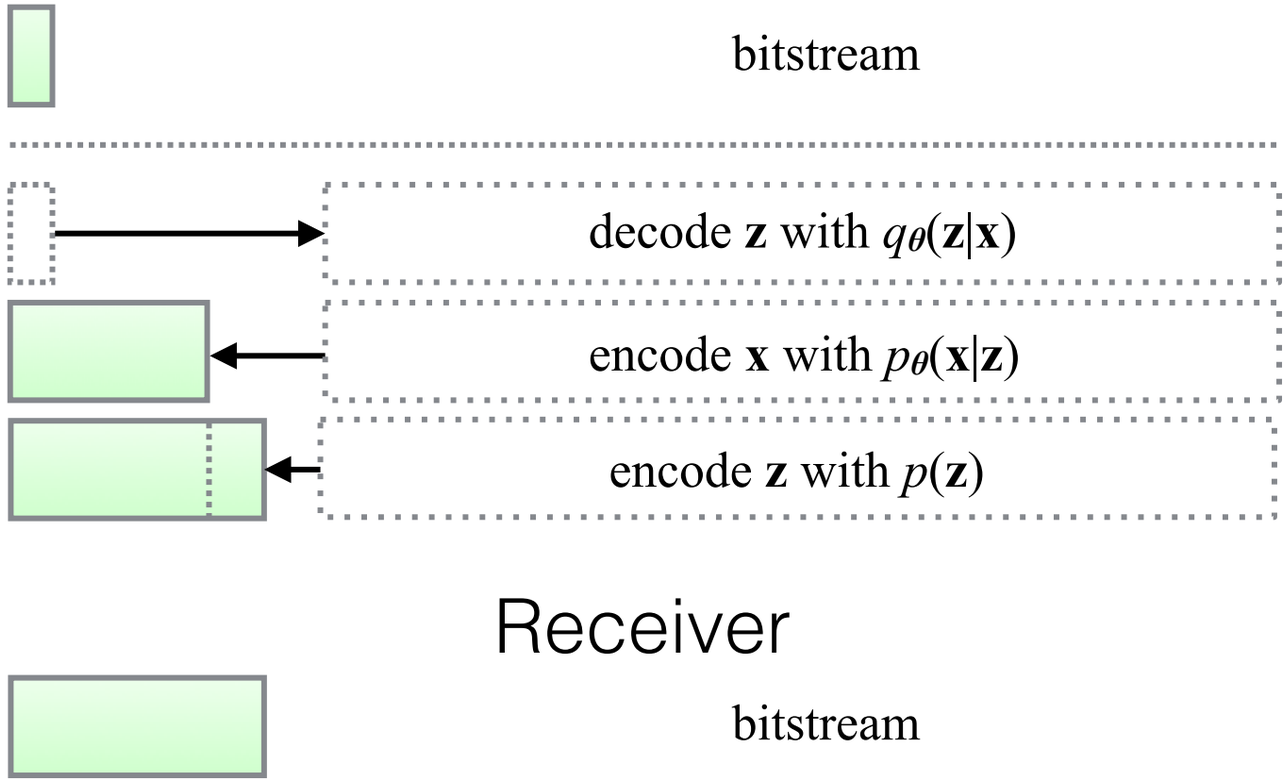
$$\text{Bitrate} = \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}[\log q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \log p(\mathbf{z})]$$
$$= -\mathcal{L}(\boldsymbol{\theta}) \text{ (ELBO)}$$

Optimizing model for ELBO

=

Directly optimizing for bitrates

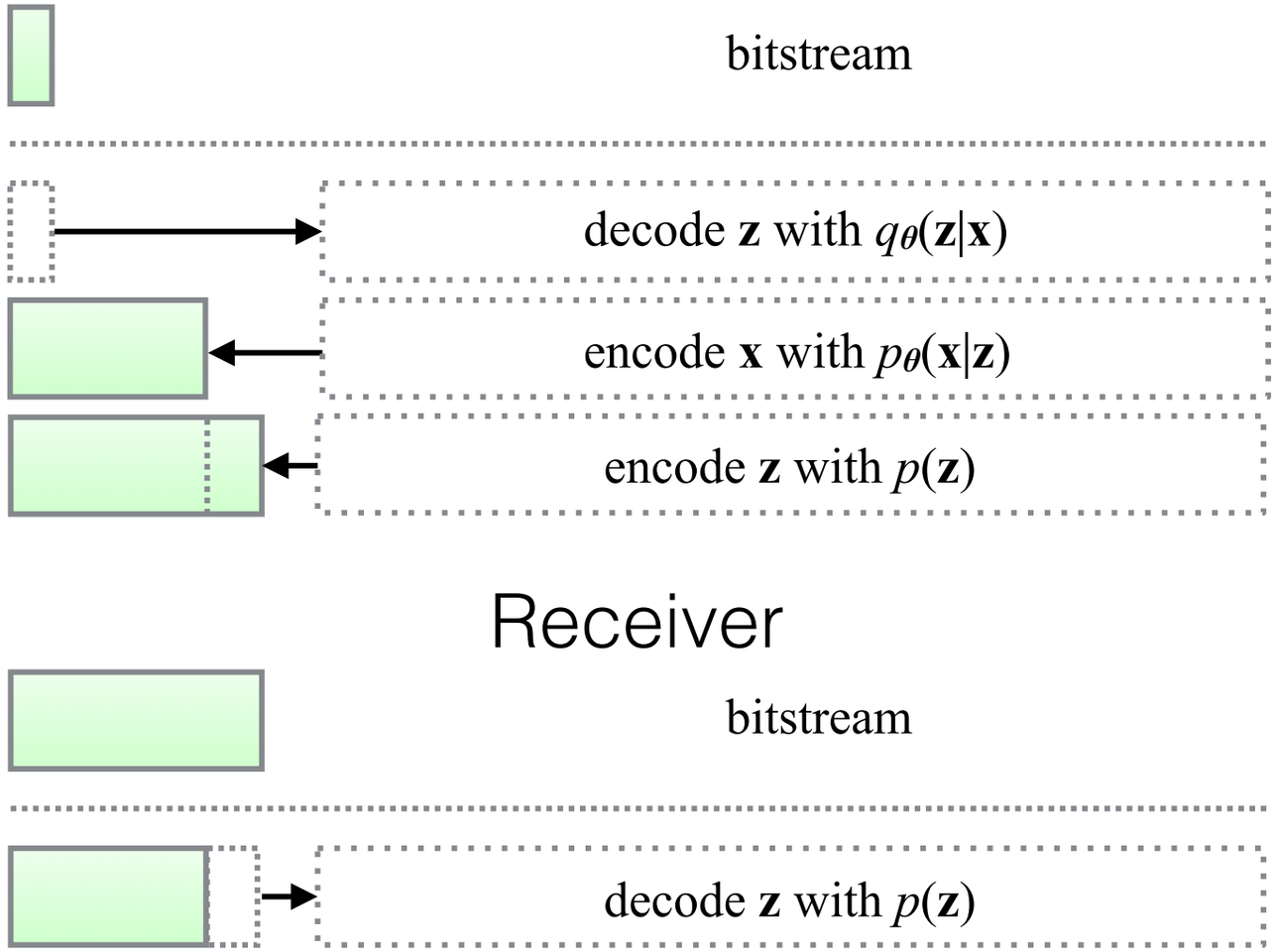
How do we compress with a **“regular”** Latent Variable Model?
Bits-Back Coding



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

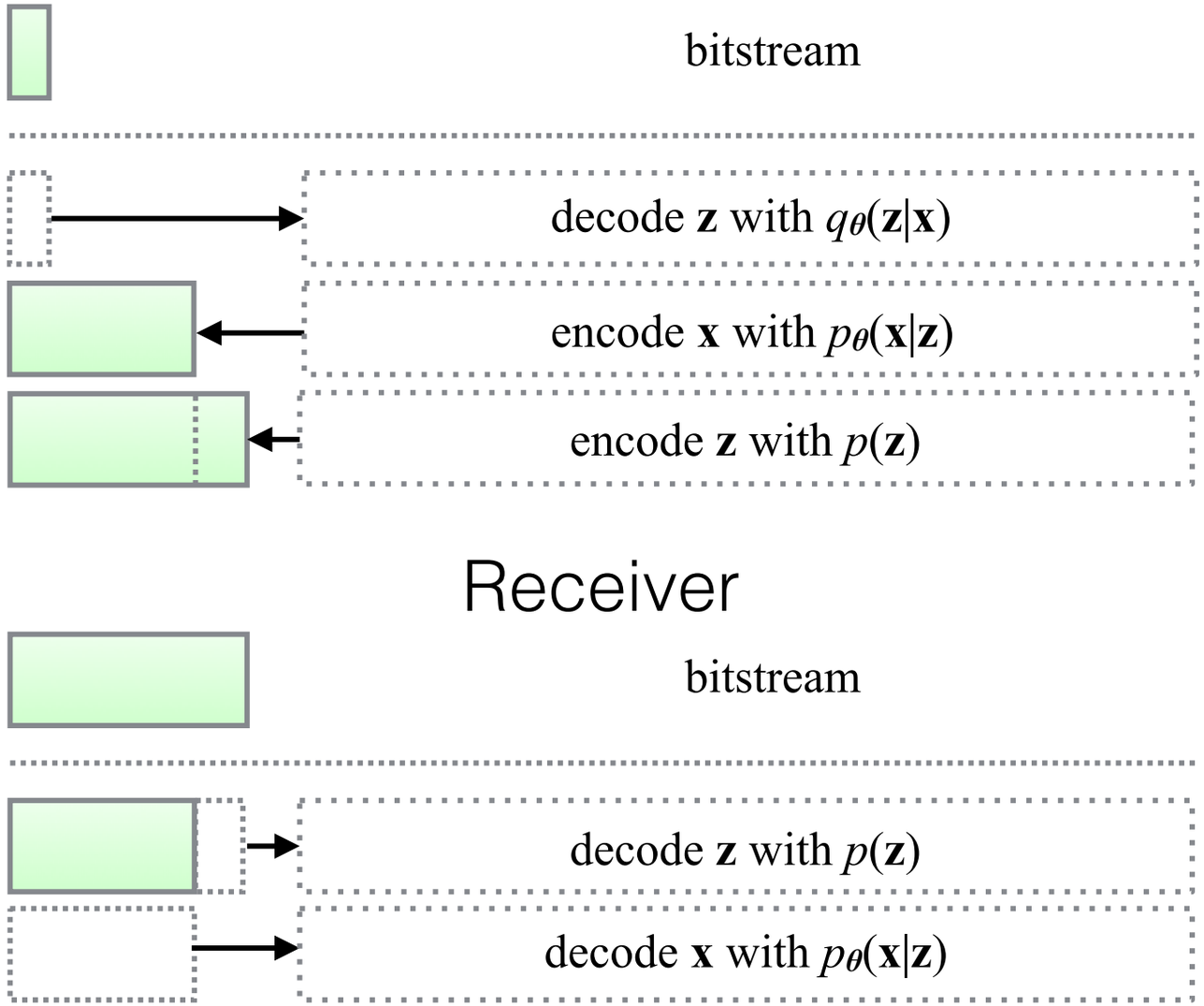
Townsend et al. (2019)

How do we compress with a **“regular”** Latent Variable Model?
Bits-Back Coding



Townsend et al. (2019)

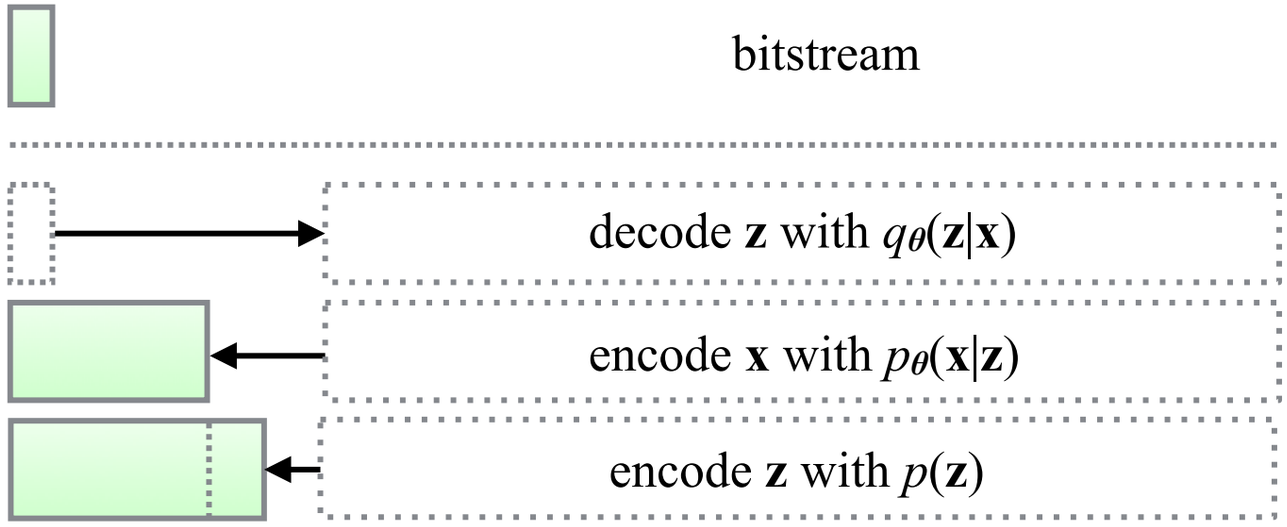
How do we compress with a **“regular”** Latent Variable Model?
Bits-Back Coding



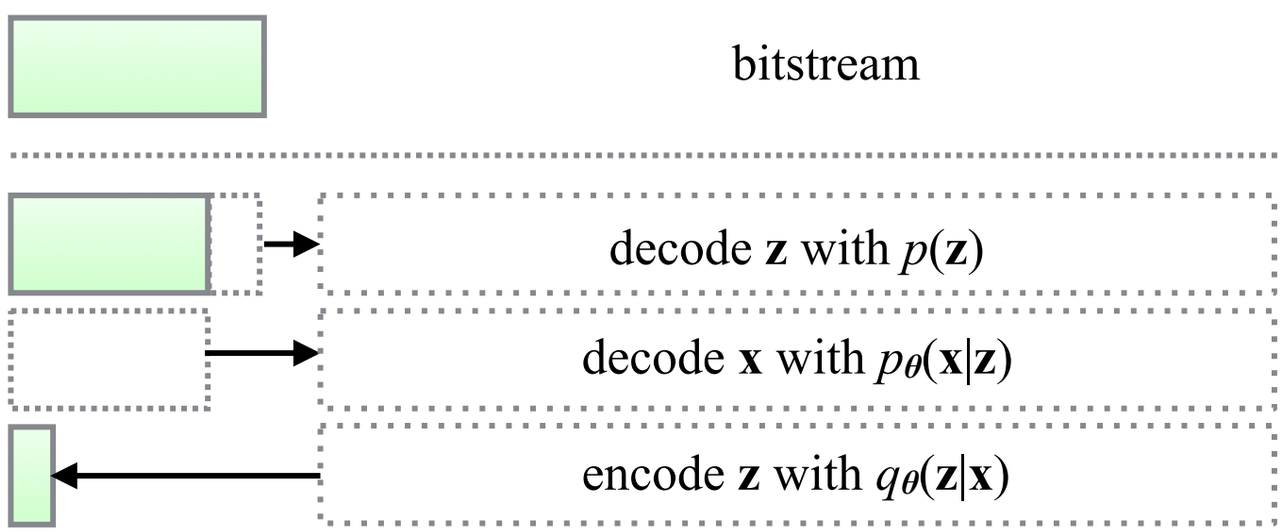
Townsend et al. (2019)

How do we compress with a **“regular”** Latent Variable Model?
Bits-Back Coding

Sender



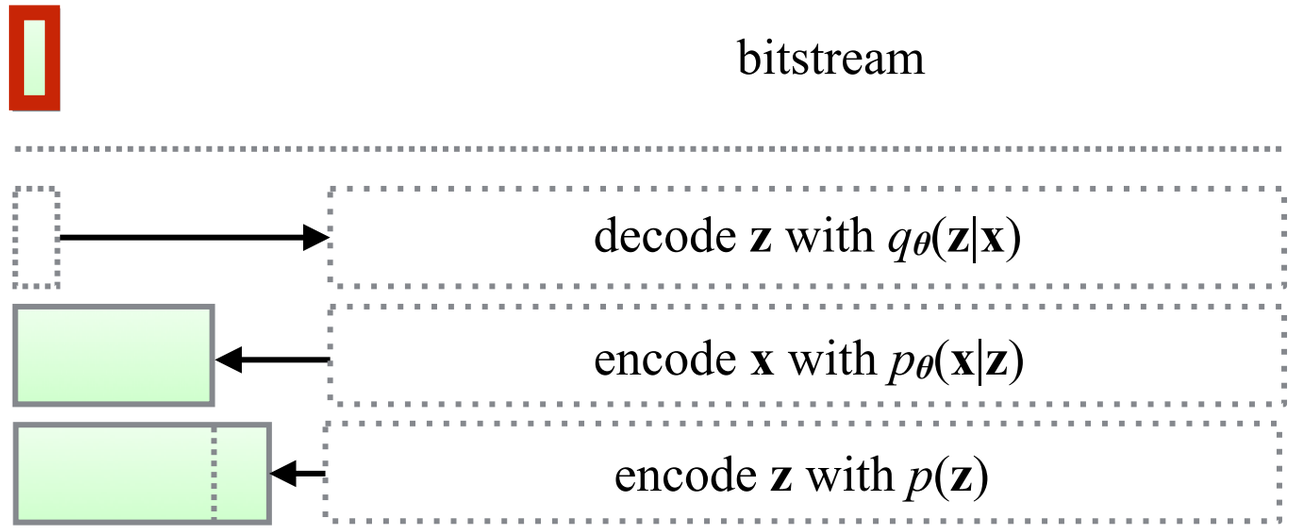
Receiver



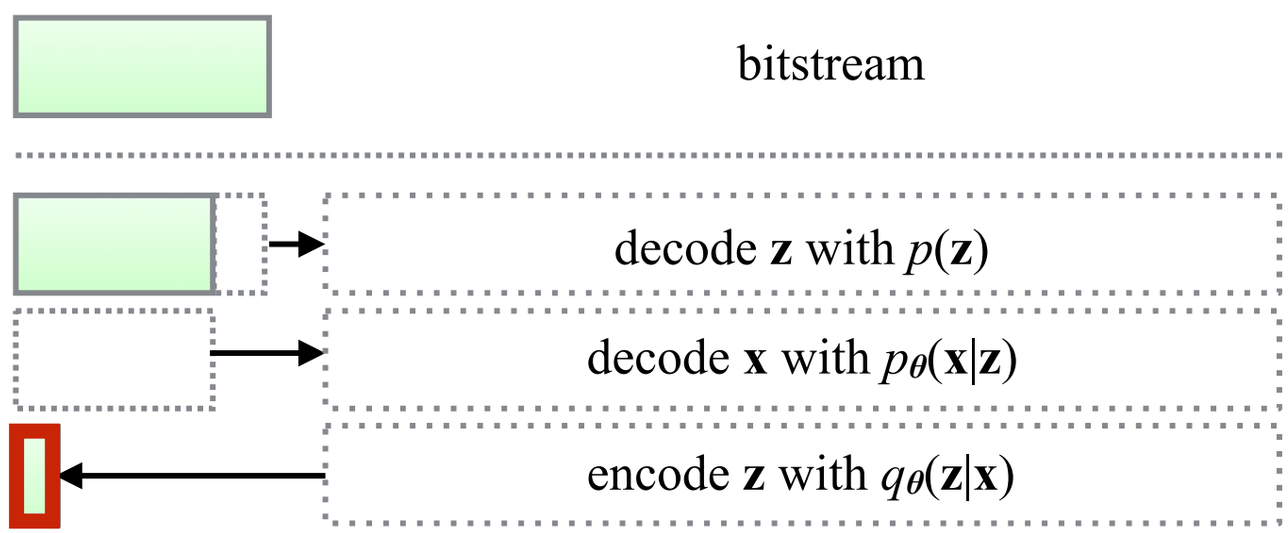
Townsend et al. (2019)

How do we compress with a “**regular**” Latent Variable Model?
Bits-Back Coding

Sender



Receiver

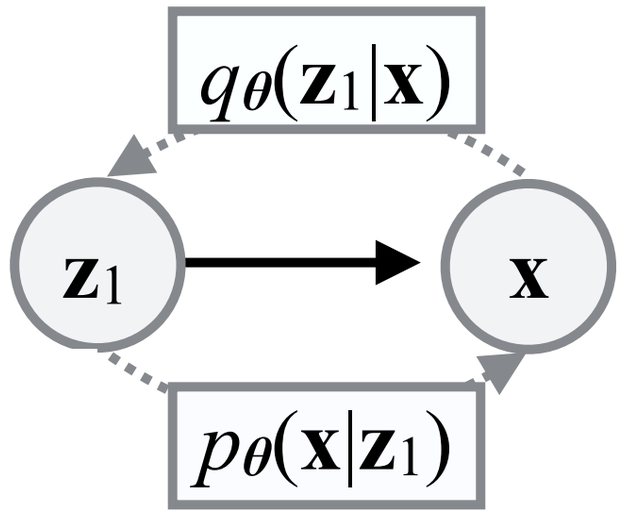


Initial bits are “back”

How do we compress with a
Hierarchical Latent Variable Model?
First: **what is the model structure?**

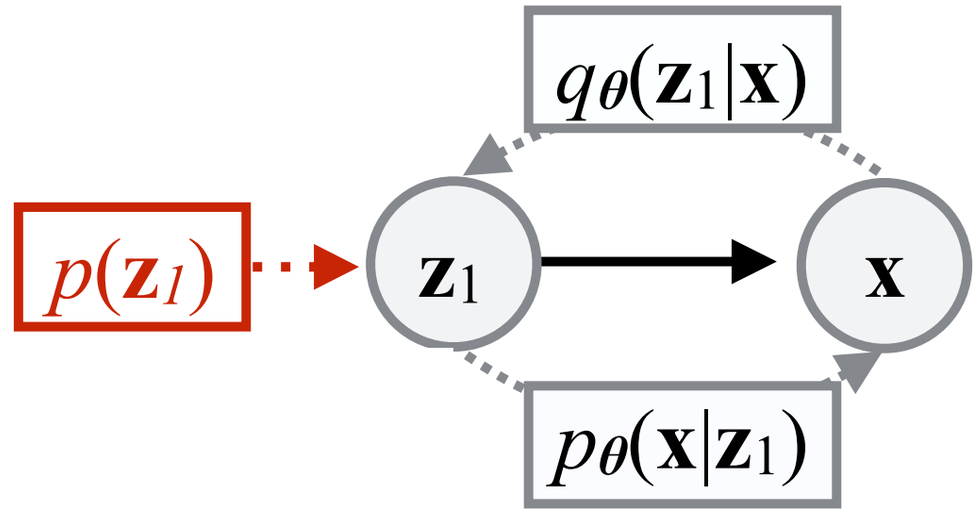
Can be thought of as multiple
nested latent variable models

How do we compress with a
Hierarchical Latent Variable Model?
First: **what is the model structure?**



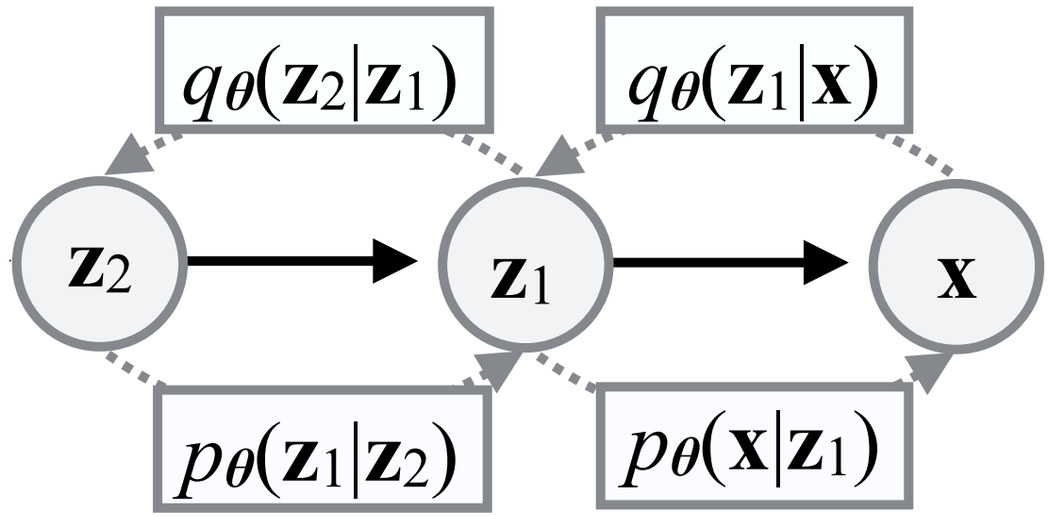
Can be thought of as multiple
nested latent variable models

How do we compress with a
Hierarchical Latent Variable Model?
First: **what is the model structure?**



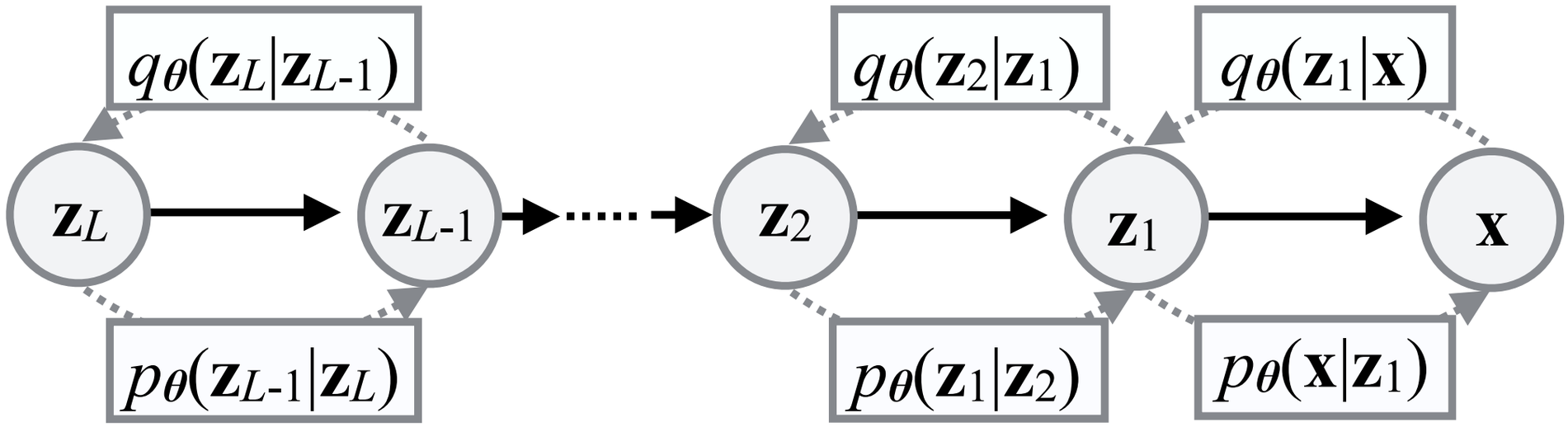
Can be thought of as multiple
nested latent variable models

How do we compress with a
Hierarchical Latent Variable Model?
First: **what is the model structure?**



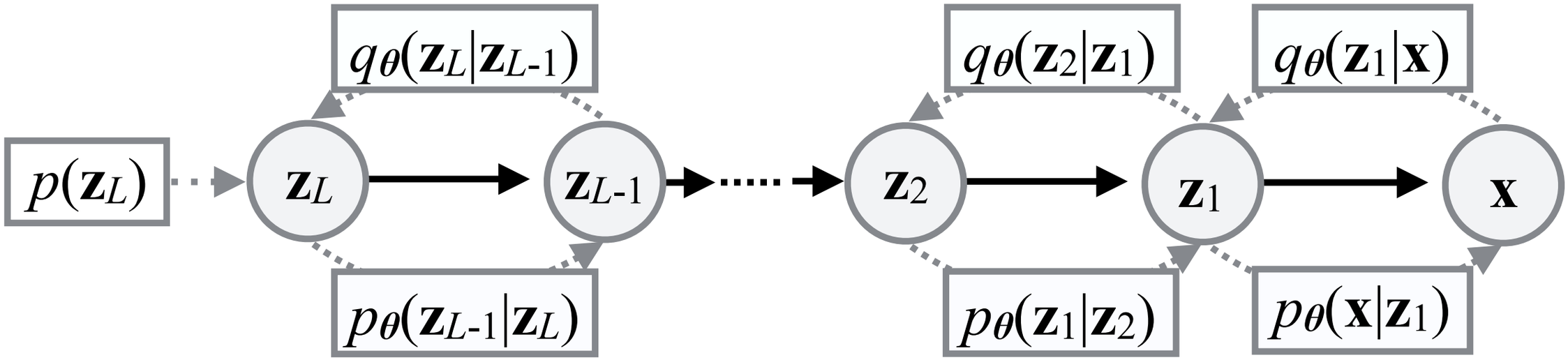
Can be thought of as multiple
nested latent variable models

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 $(L = 1) : -\log q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$

Initial bits

What if we **naively** apply Bits-Back Coding?

Initial bits $(L = 1) : -\log q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$

Initial bits $(L > 1) : \sum_{i=0}^{L-1} -\log q_{\boldsymbol{\theta}}(\mathbf{z}_{i+1}|\mathbf{z}_i)$

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

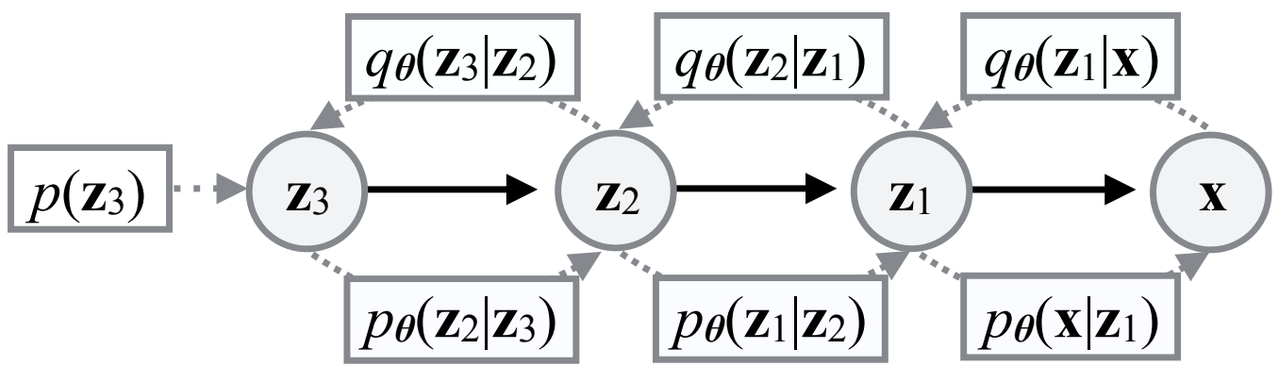
initial bitstream

Corresponding step Bits-Back Coding

decode $\mathbf{z}$ with $q(\mathbf{z}|\mathbf{x})$

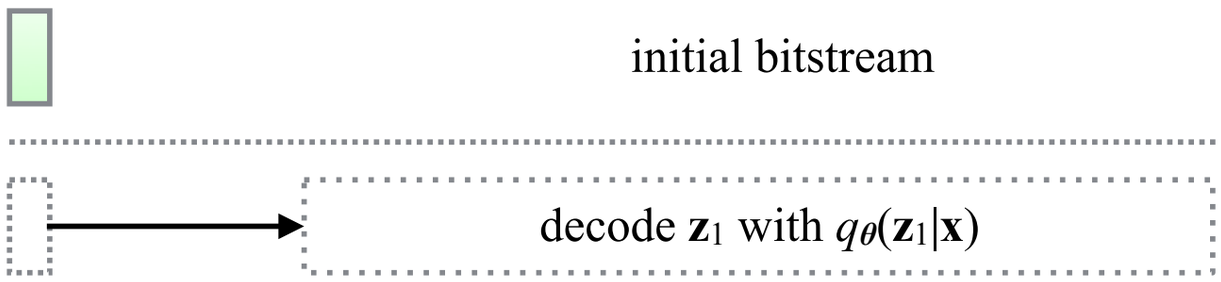
encode $\mathbf{x}$ with $p(\mathbf{x}|\mathbf{z})$

encode $\mathbf{z}$ with $p(\mathbf{z})$

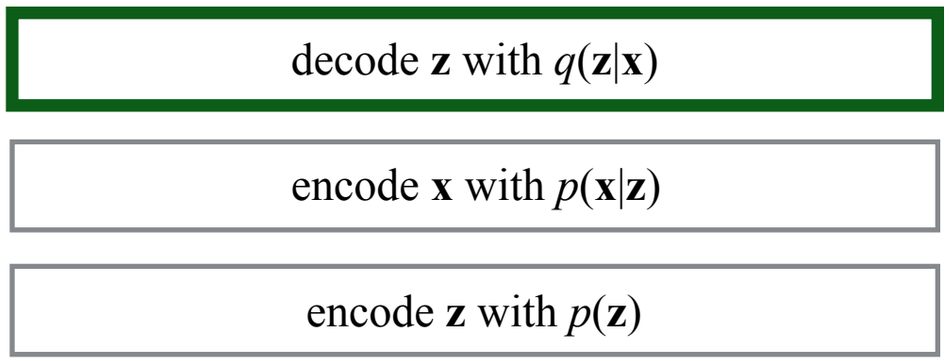


Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

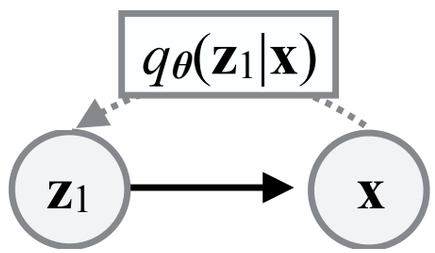
Sender



Corresponding step Bits-Back Coding

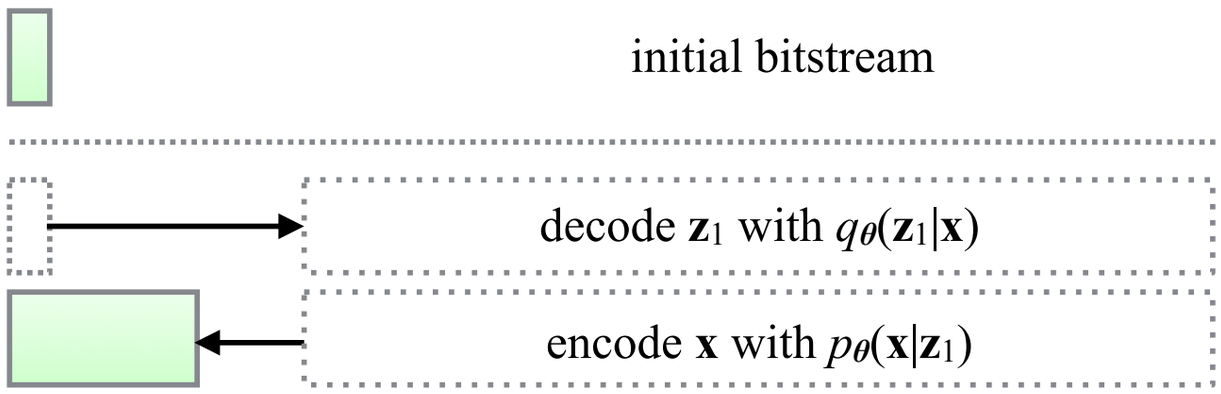


$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x})$$

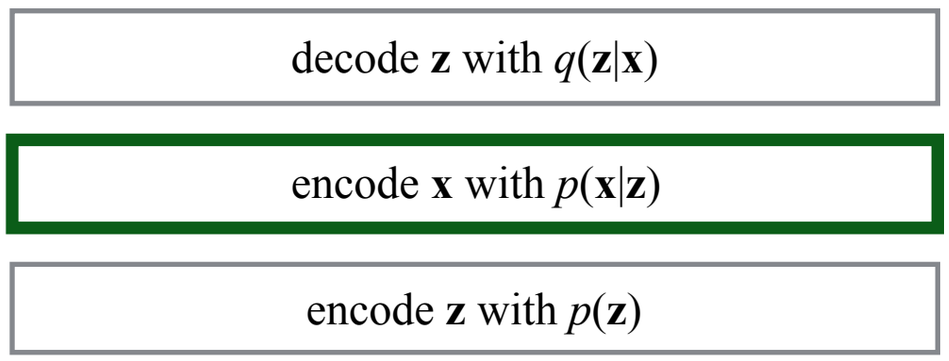


Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

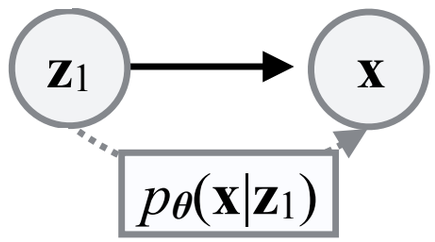
Sender



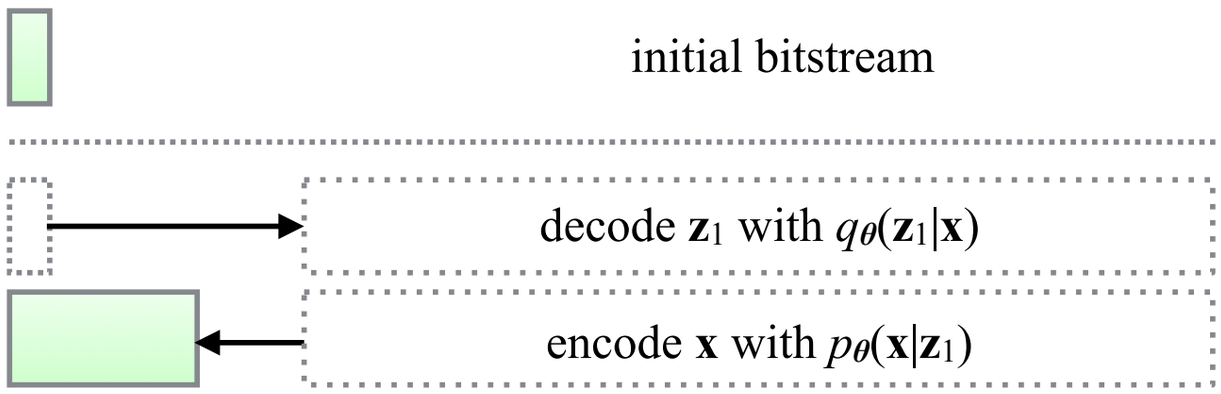
Corresponding step
Bits-Back Coding



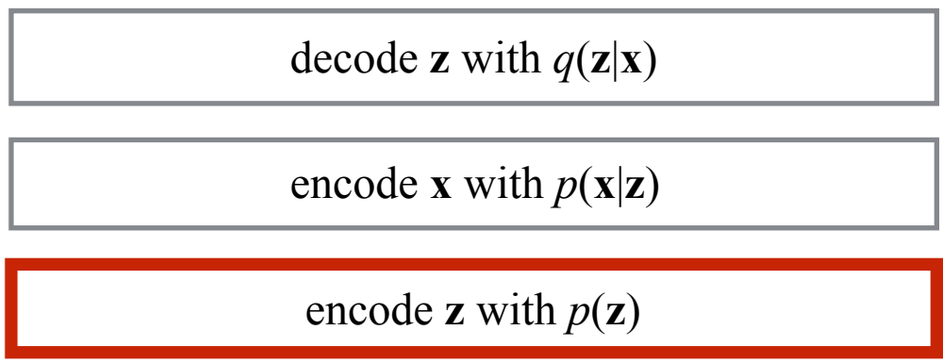
$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1)$$



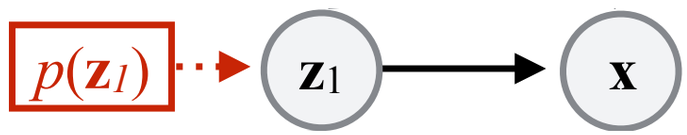
Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)



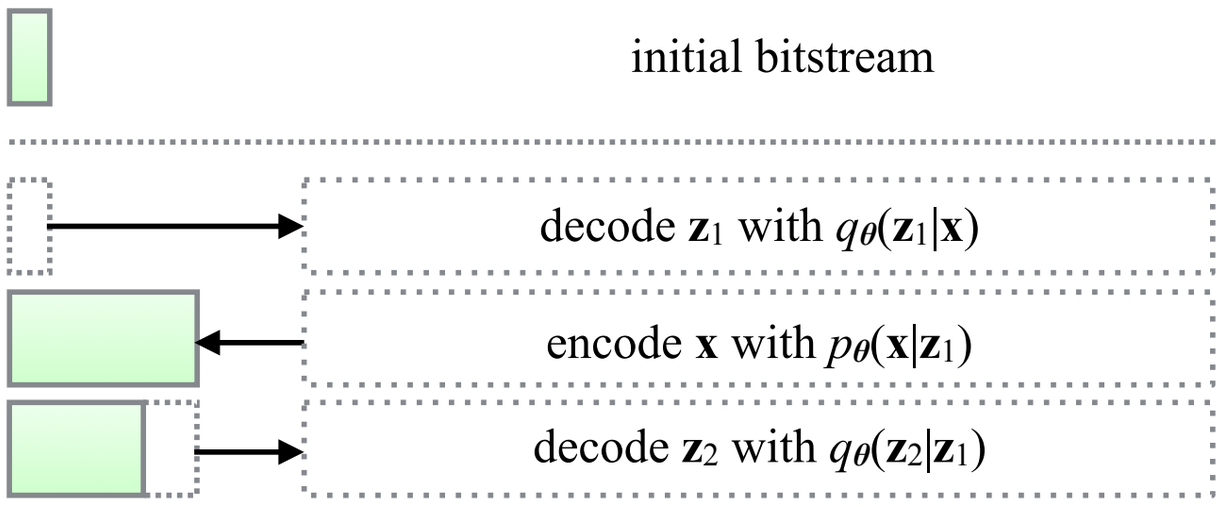
Corresponding step Bits-Back Coding



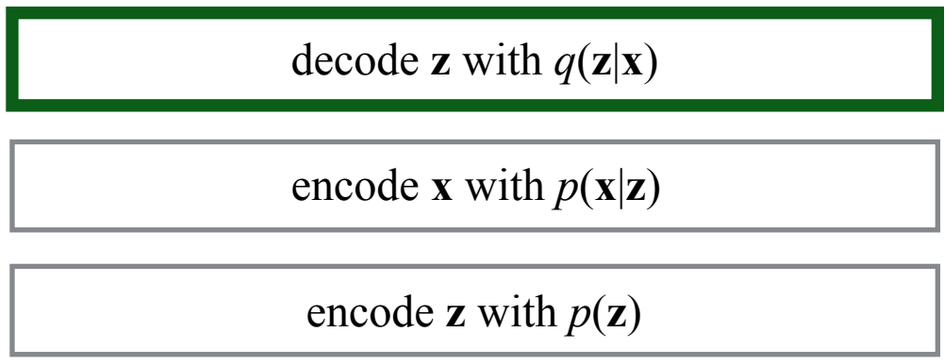
$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) - \log p(\mathbf{z}_1)$$



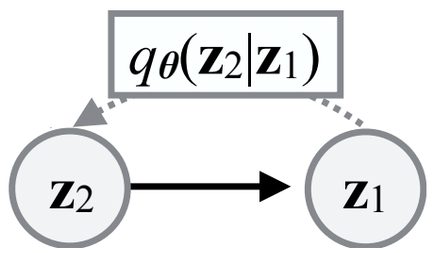
Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)



Corresponding step Bits-Back Coding

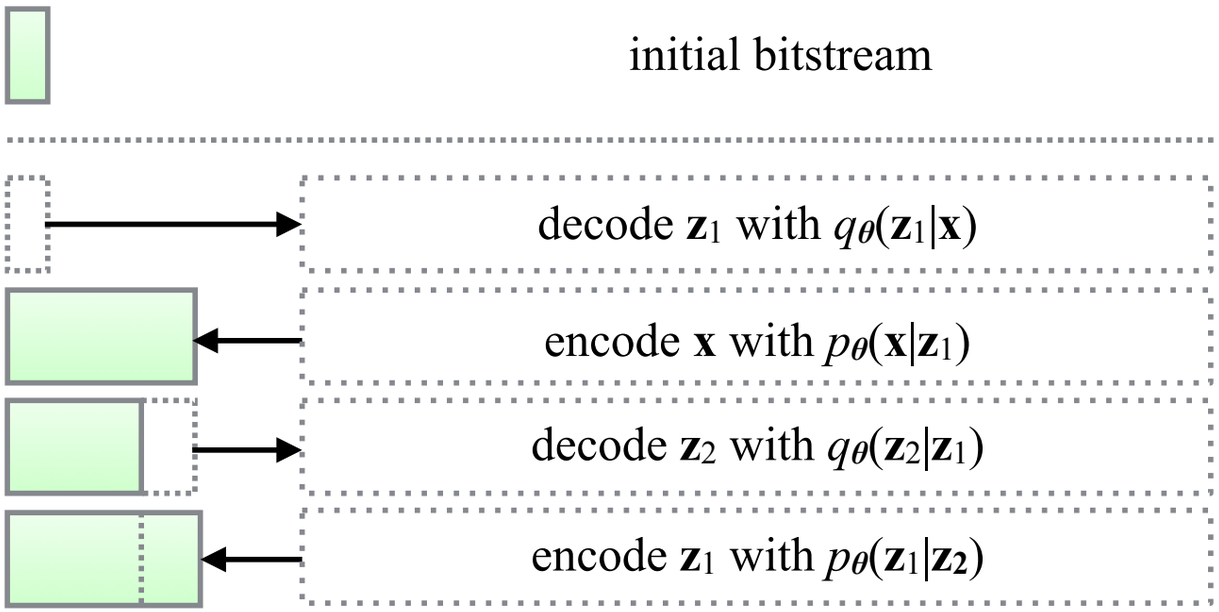


$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) + \log q_{\boldsymbol{\theta}}(\mathbf{z}_2|\mathbf{z}_1)$$

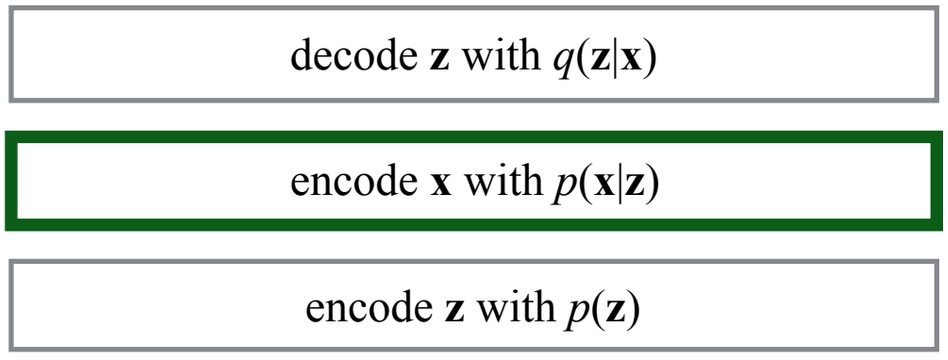


Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

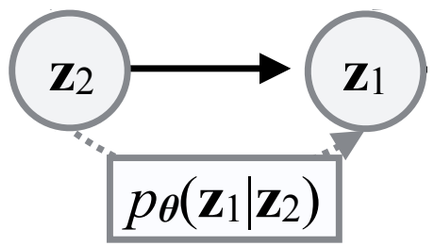
Sender



Corresponding step Bits-Back Coding

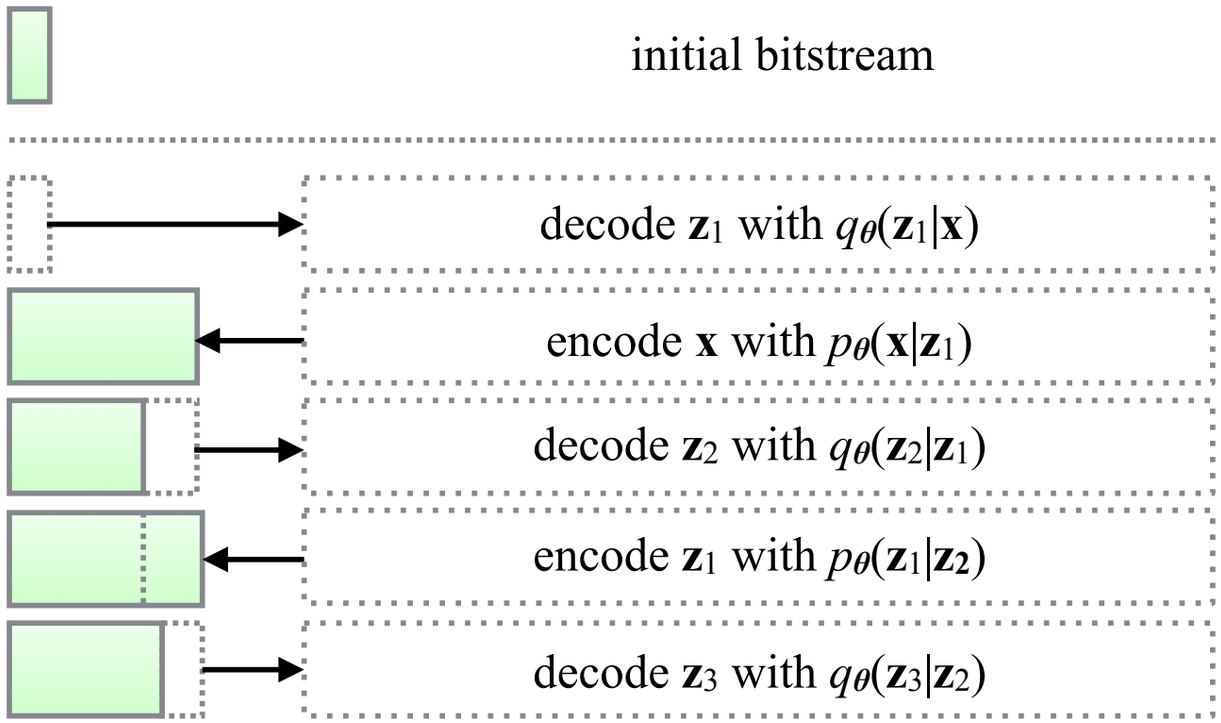


$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) + \log q_{\boldsymbol{\theta}}(\mathbf{z}_2|\mathbf{z}_1) - \log p_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{z}_2)$$

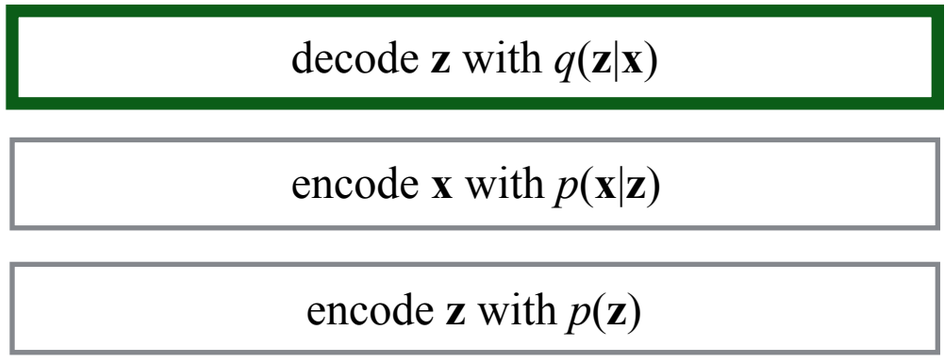


Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

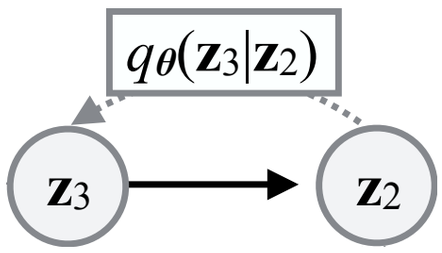
Sender



Corresponding step Bits-Back Coding

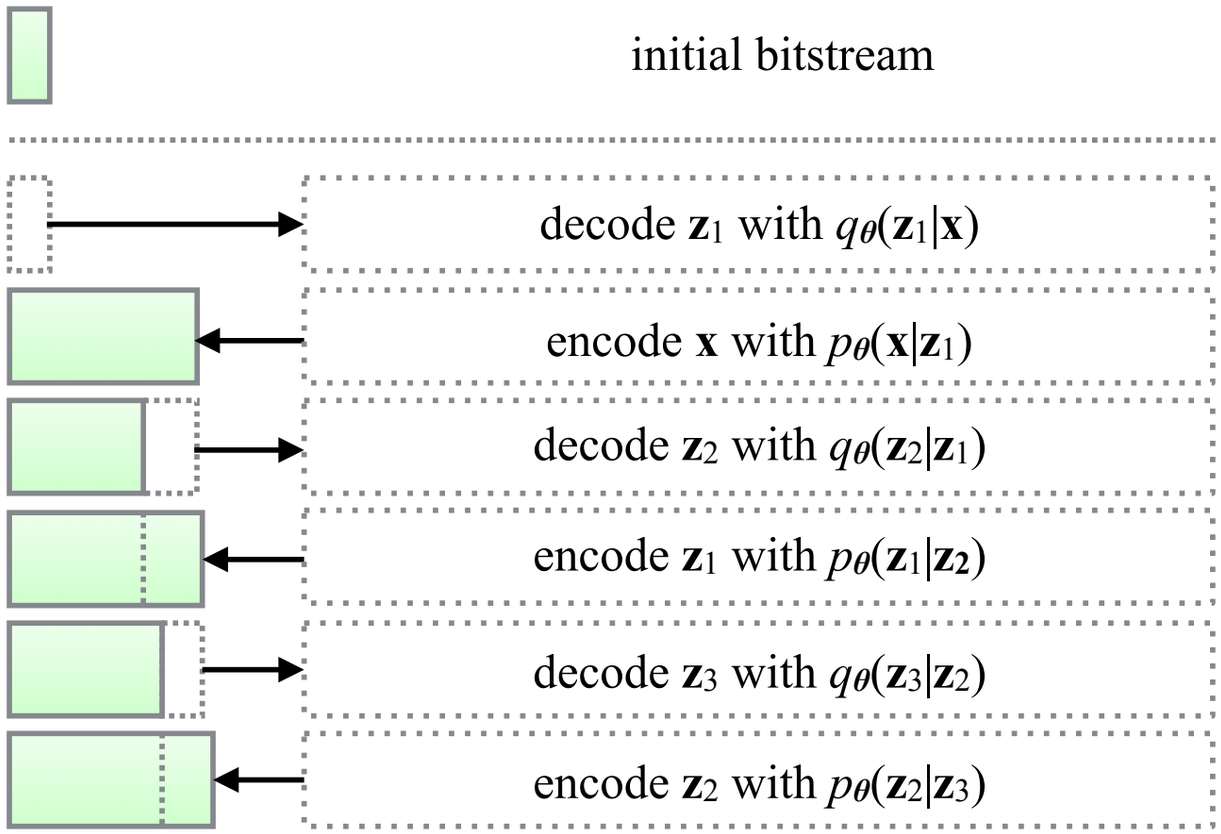


$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_{1:2}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2|\mathbf{z}_3)$$

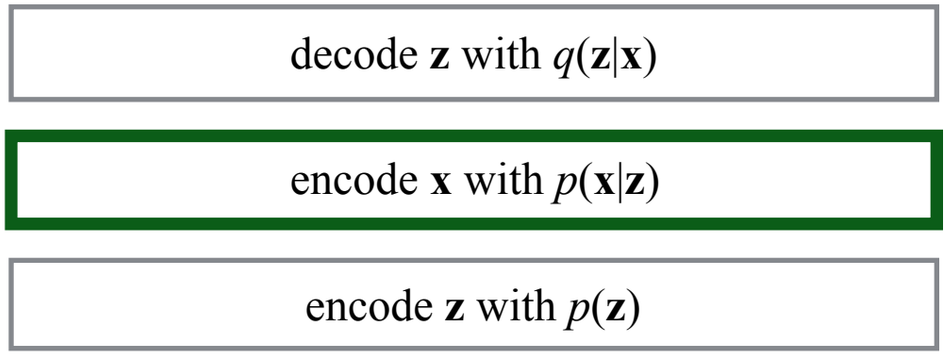


Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

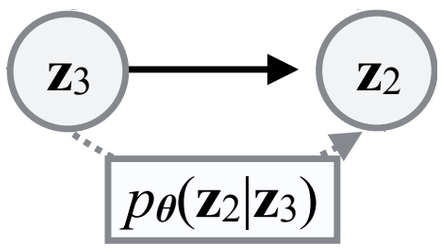
Sender



Corresponding step Bits-Back Coding

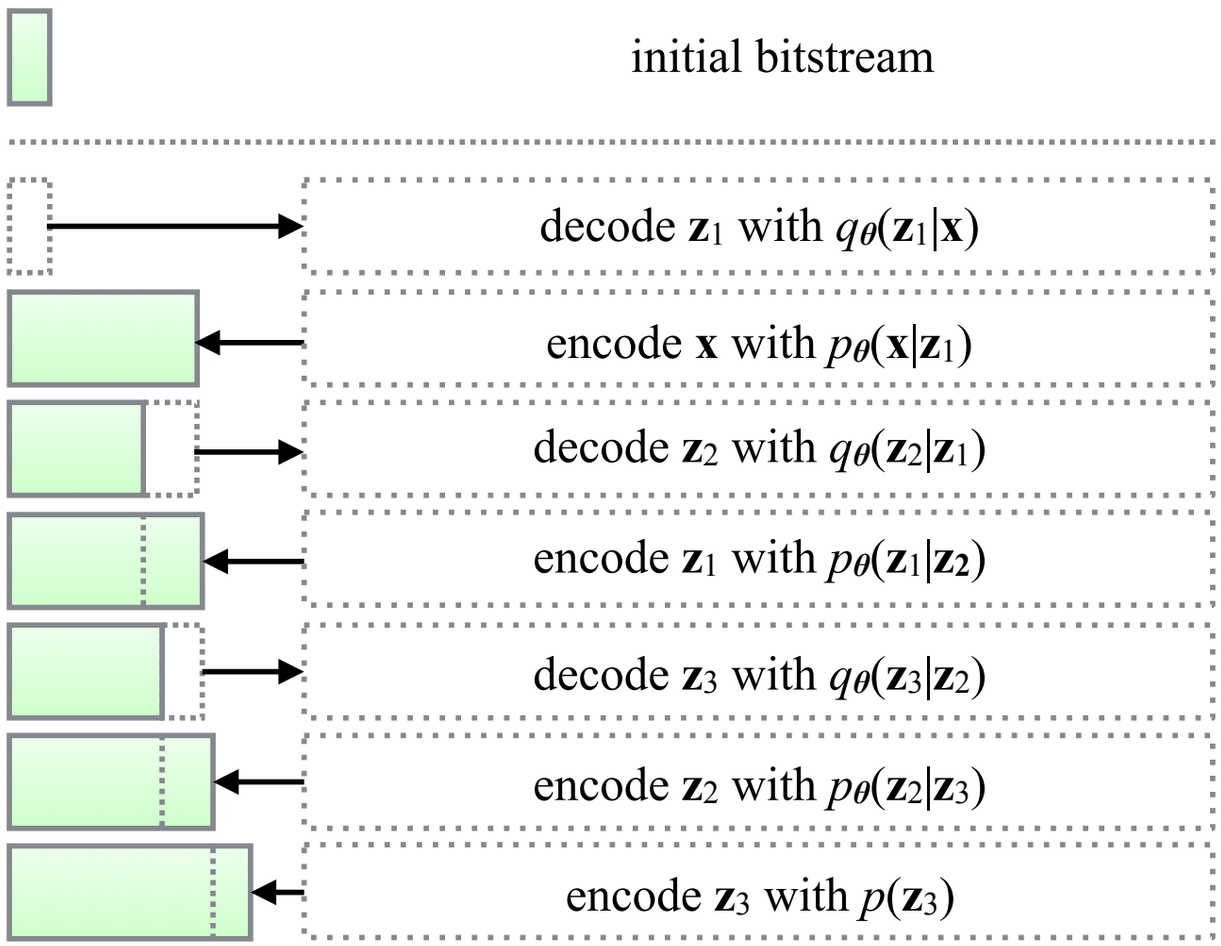


$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_{1:3}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2|\mathbf{z}_3)$$

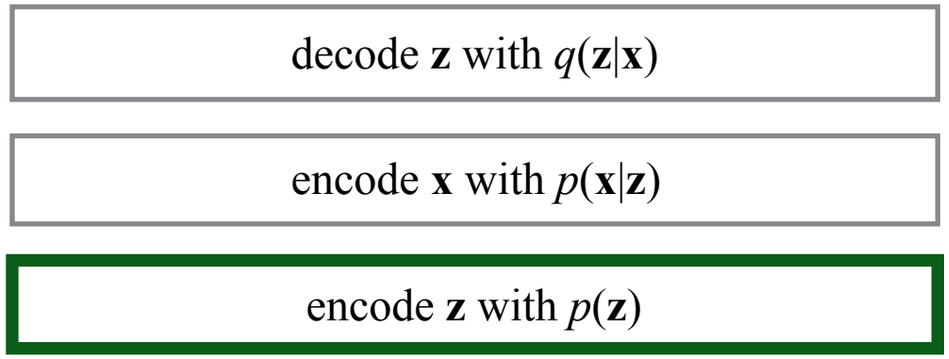


Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

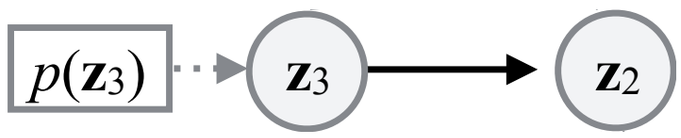
Sender



Corresponding step Bits-Back Coding



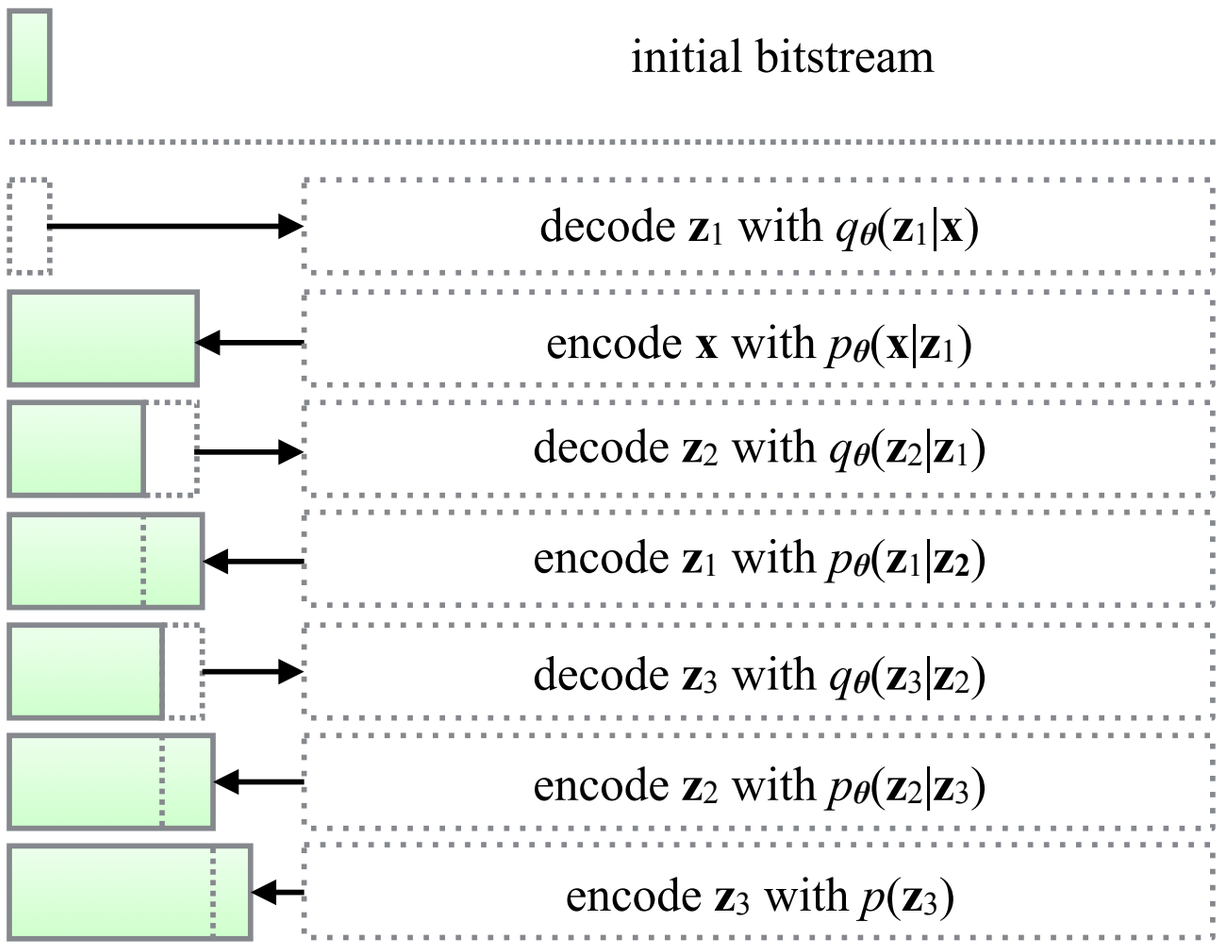
$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_{1:3}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2|\mathbf{z}_3) - \log p(\mathbf{z}_3)$$



Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

Sender

Receiver

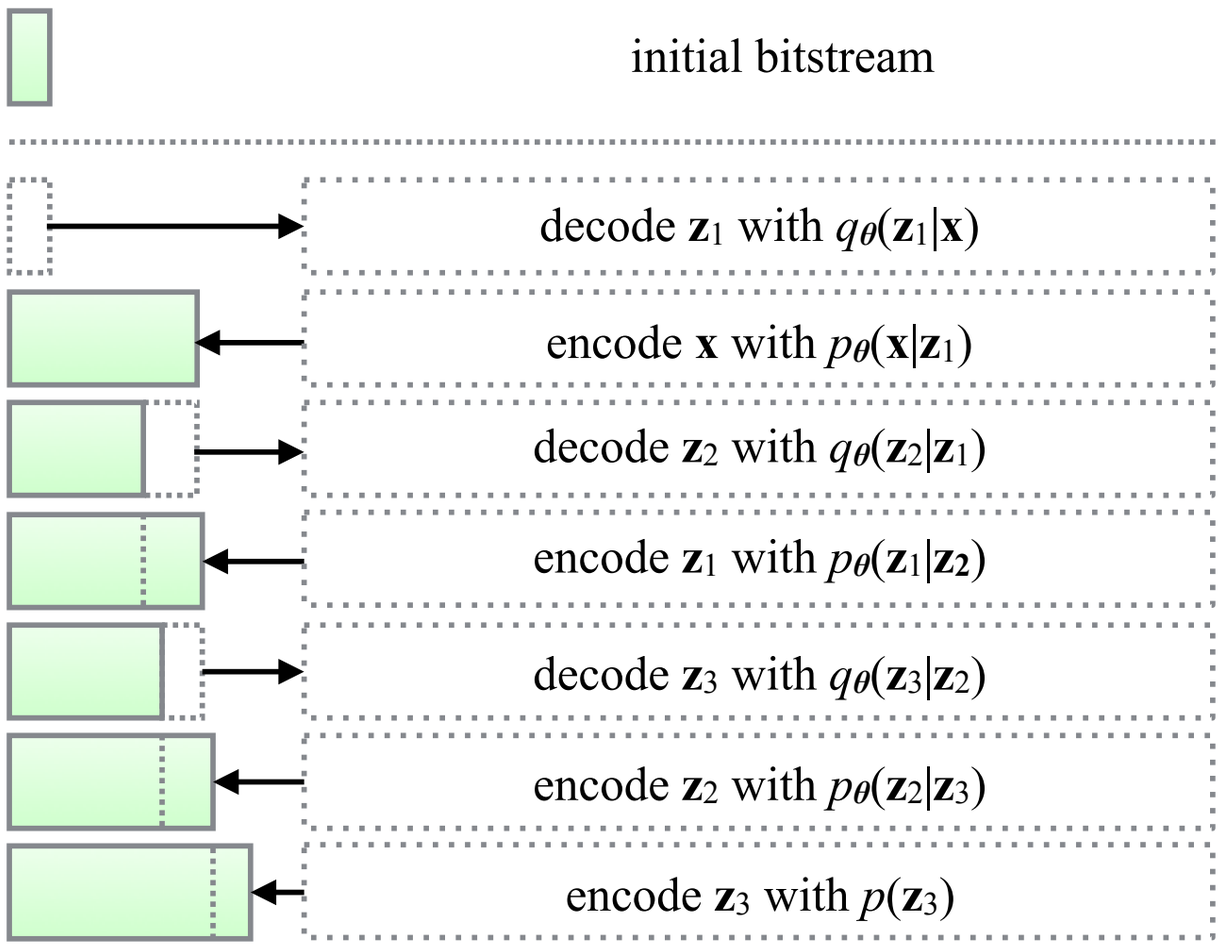


$$= \log q_{\boldsymbol{\theta}}(\mathbf{z}_{1:3}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_{1:3})$$

Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)

Sender

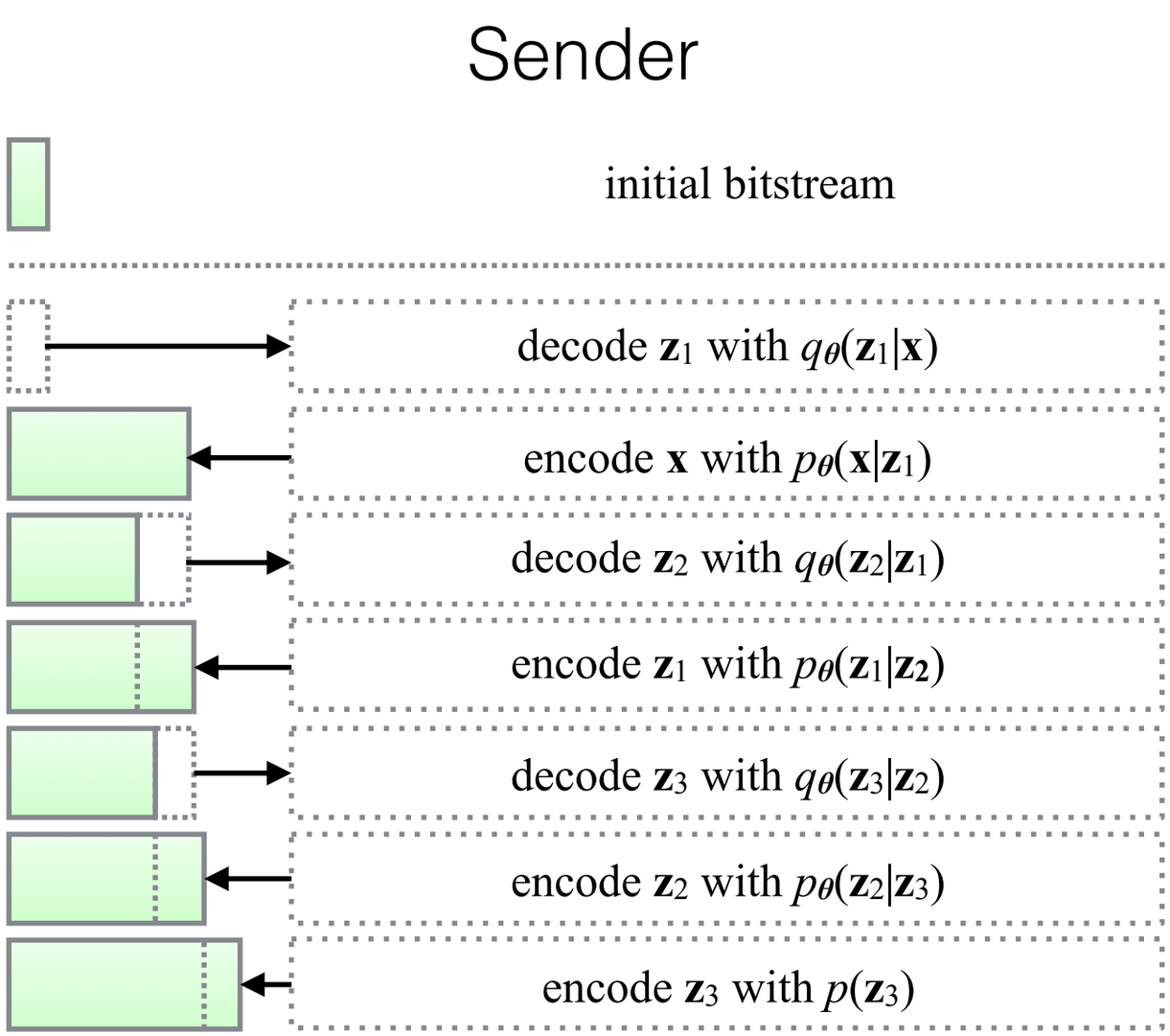
Receiver



$$= \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}_{1:3}|\mathbf{x})}[\log q_{\boldsymbol{\theta}}(\mathbf{z}_{1:3}|\mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_{1:3})]$$

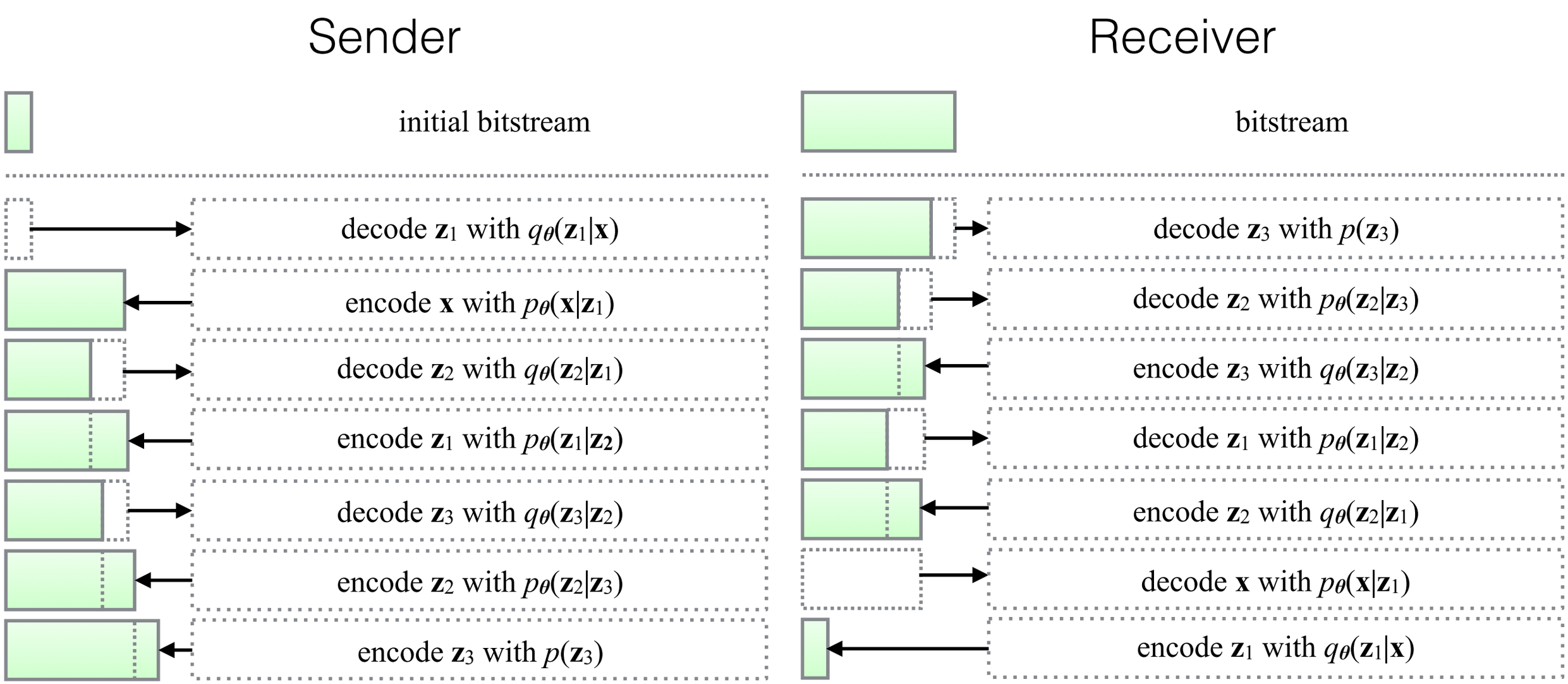
$$= -\mathcal{L}(\boldsymbol{\theta}) \text{ (ELBO)}$$

Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)



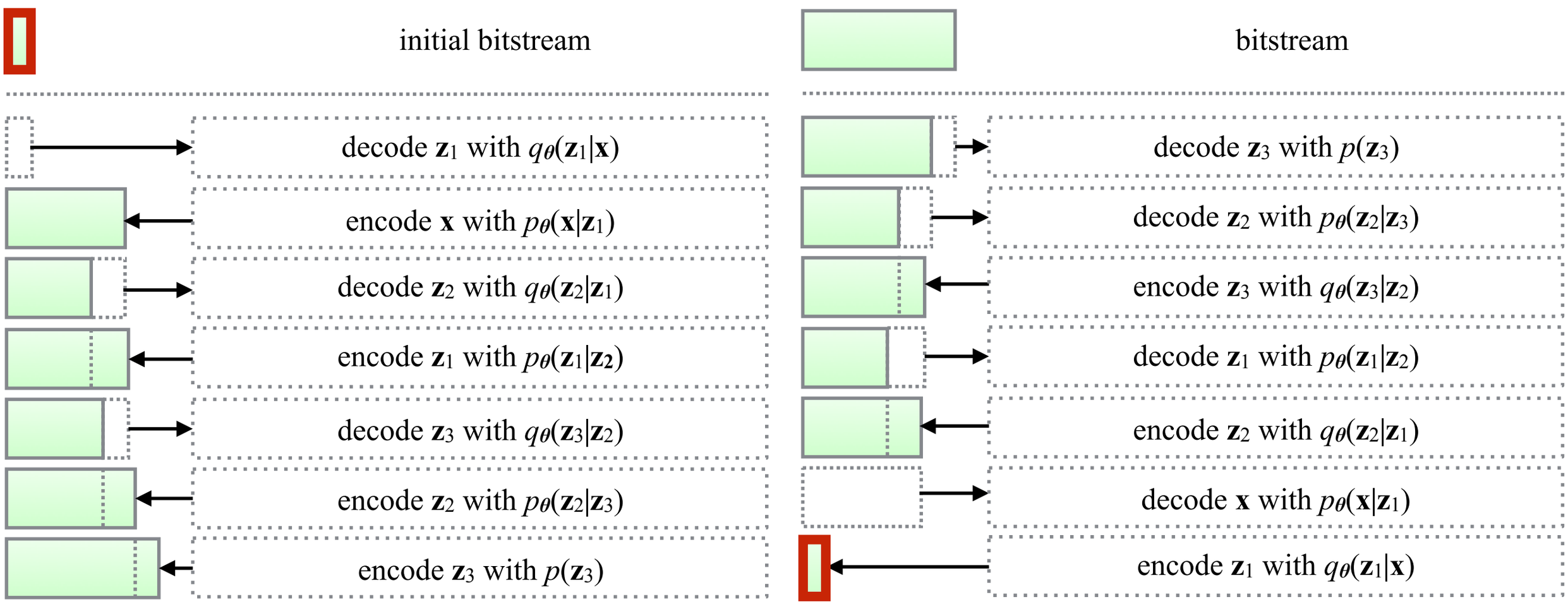
- Operations in reverse order
- With encode and decode operations switched

Compression with Hierarchical Latent Variable Models:
Bit-Swap (ours)



- 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 ….**

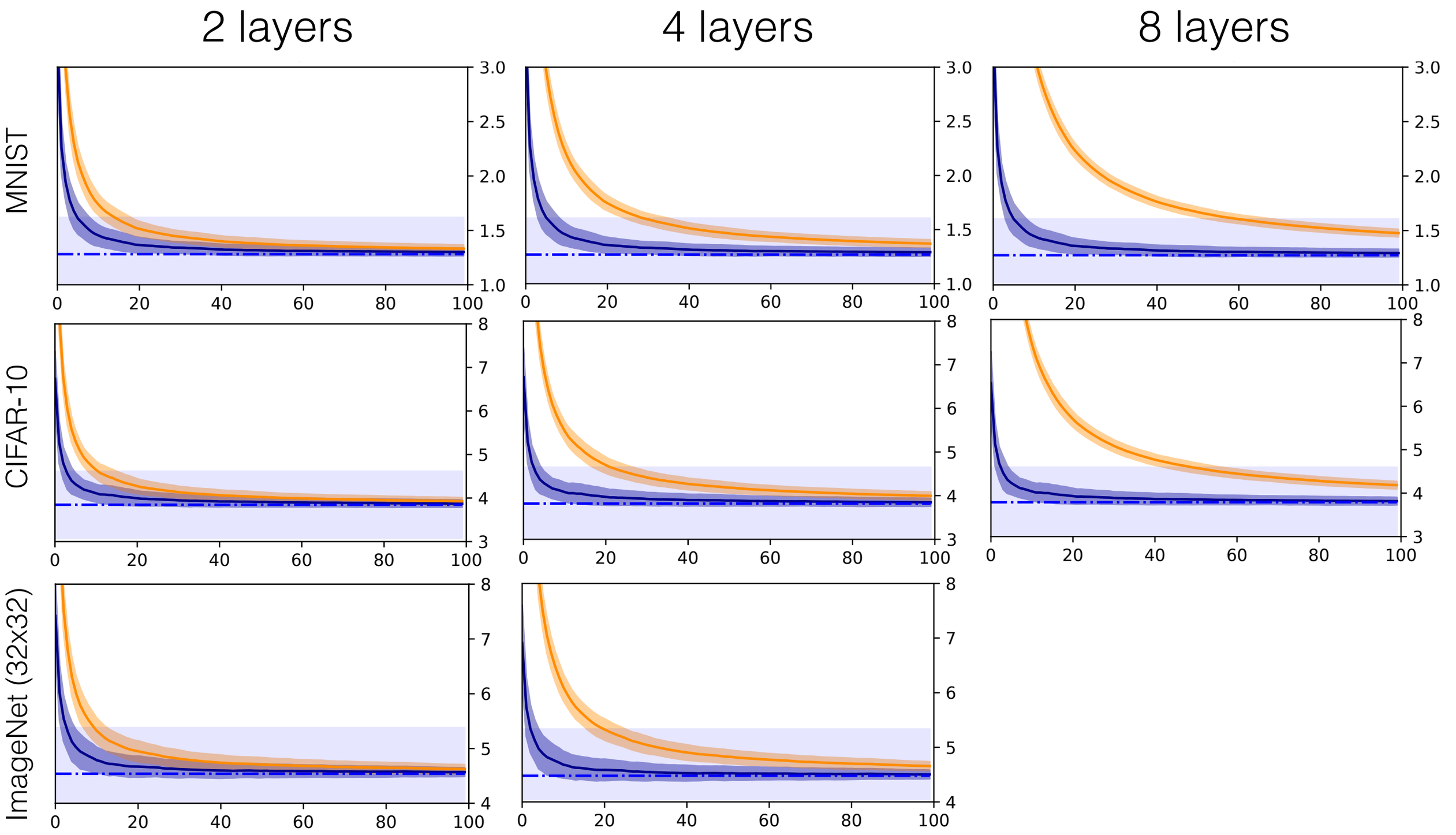
Bit-Swap initial bits is **bounded**

$$N_{\text{init}}^{\text{BitSwap}} \leq \sum_{i=0}^{L-1} \max\left(0, \log \frac{p_{\boldsymbol{\theta}}\left(\mathbf{z}_{i-1} \middle| \mathbf{z}_i\right)}{q_{\boldsymbol{\theta}}\left(\mathbf{z}_{i+1} \middle| \mathbf{z}_i\right)}\right)$$

Instead of growing linearly

Cumulative average compression rate of compressing 100 images in sequence

= Bit-Swap = Bits-Back Coding



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

	Compression Rate (bits/dim)
uncompressed	8.00
gzip	5.96
bzip2	5.07
LZMA	5.09
PNG	4.71
WebP	3.66
Bits-Back Coding	3.62
Bit-Swap	**3.51**

Bit-Swap

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

Friso Kingma, *Pieter Abbeel, Jonathan Ho*