Learned Image and Video Compression with Deep Neural Networks


Dong Xu
University of Sydney, Australia


Guo Lu
Beijing Institute of Technology, China


Ren Yang
ETH Zurich, Switzerland


Radu Timofte
ETH Zurich, Switzerland

Learned Image and Video Compression with Deep Neural Networks

Part 1 Learned Image Compression

Ren Yang
Ph.D. student

Radu Timofte
Lecturer, Group leader

Computer Vision Laboratory
ETH Zurich, Switzerland

ETH zürich

7296 x 5472 = 39,923,712 pixels

Uncompressed image: 39,923,712 x 3 = 120 MB

Uncompressed video (60 fps): 120 MB x 60 = 7.2 GBps (18s needs 128 GB)

Lossless compression (.png): 44 MB

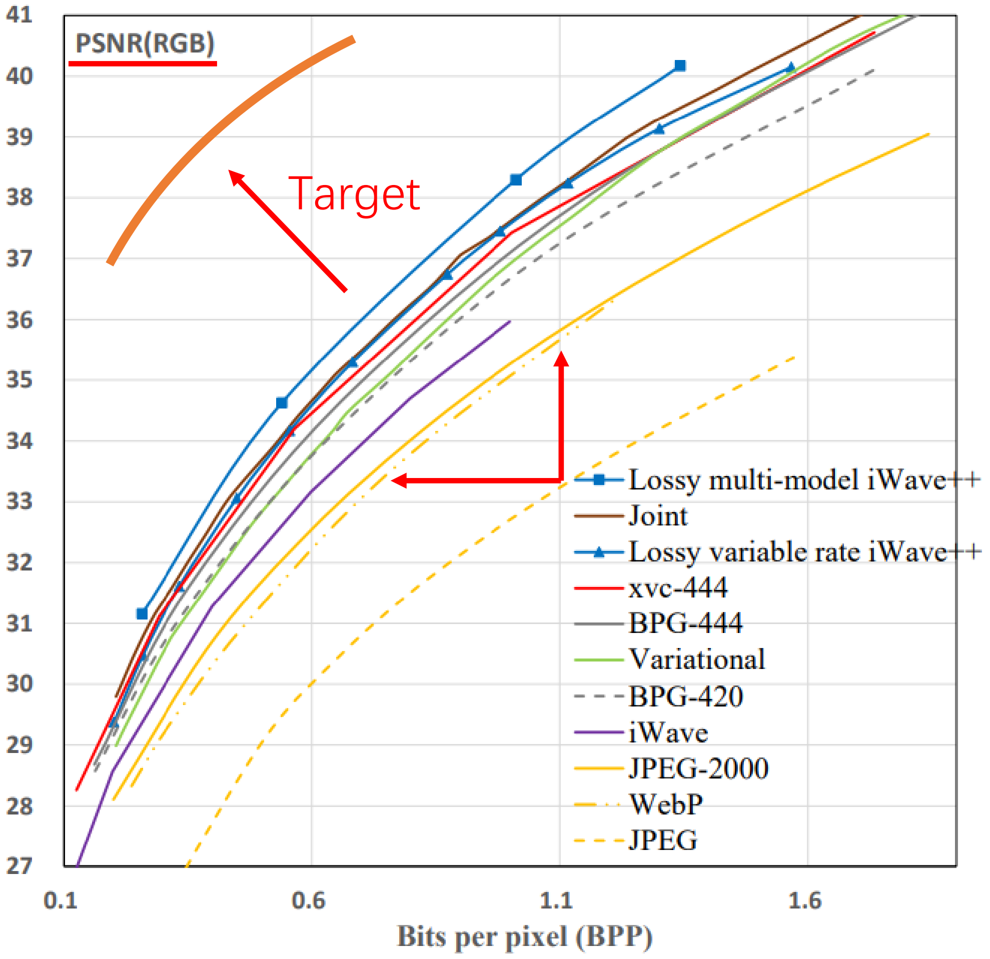
Lossy compression (.jpg): 9 MB

Image/video compression plays an important role in multimedia streaming, online conference, data storage, etc.

Rate-distortion trade-off

Metrics: PSNR, (MS-)SSIM, NIMA, LPIPS, user studies, etc.

Higher quality
(lower distortion)



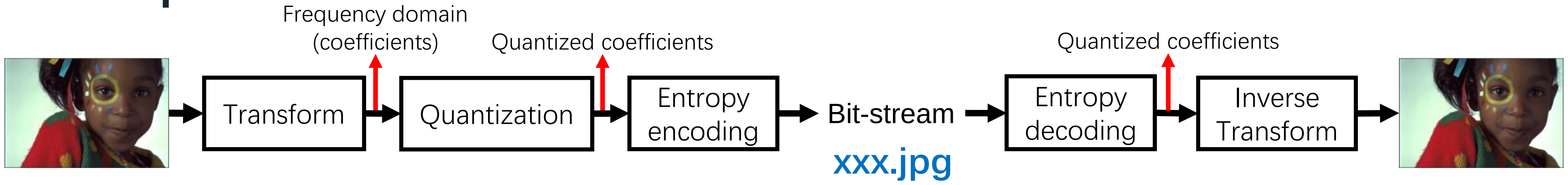
Better quality at the same bit-rate

Lower bit-rate at the same quality

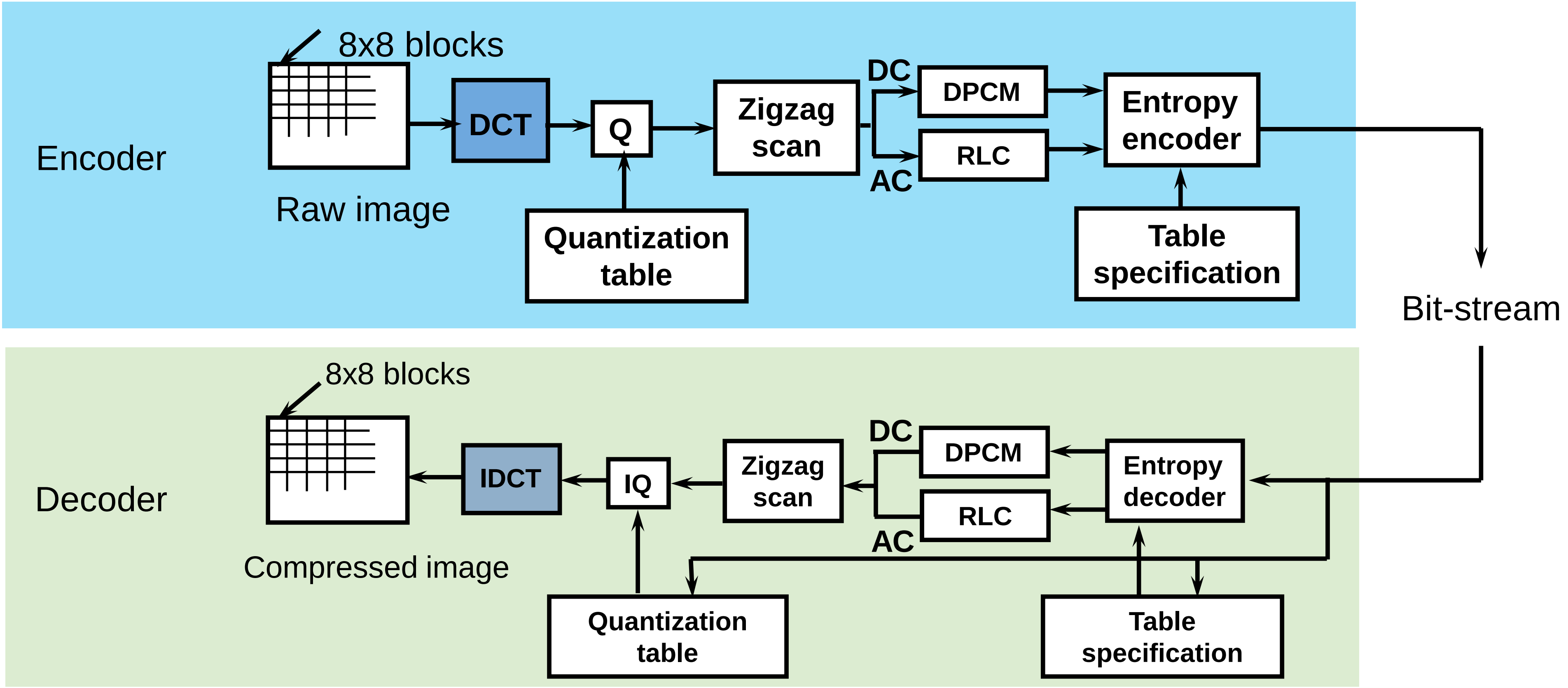
Higher bit-rate (file size)

Traditional Image Compression

- Classical Architecture:



- Standards: JPEG (DCT + Huffman), JPEG2000 (DWT + Arithmetic coding), BPG (HEVC), ···
- Example: JPEG compression framework



Entropy coding

Entropy:

$$\mathrm{H}(X) = \mathrm{E}[\mathrm{I}(X)] = \mathrm{E}[-\log(\mathrm{P}(X))]$$

$$\boxed{\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)}$$

Cross entropy:

$$H(p,q) = -\sum_{x\in\mathcal{X}} \underbrace{p(x)}_{\text{real}}\, \log \underbrace{q(x)}_{\text{estimated}} \quad \textbf{(Eq.1)}$$

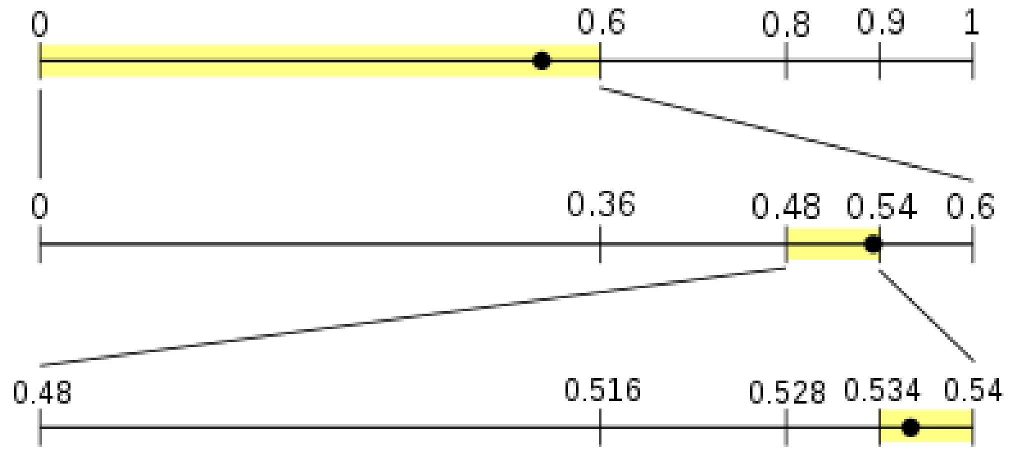
(Adaptive) arithmetic coding is theoretically able to losslessly compress data at

- bit-rate ≅ cross entropy (with little overhead)

Arithmetic coding:

- 60% chance of symbol NEUTRAL
- 20% chance of symbol POSITIVE
- 10% chance of symbol NEGATIVE
- 10% chance of symbol END-OF-DATA.

NEUTRAL NEGATIVE END-OF-DATA message



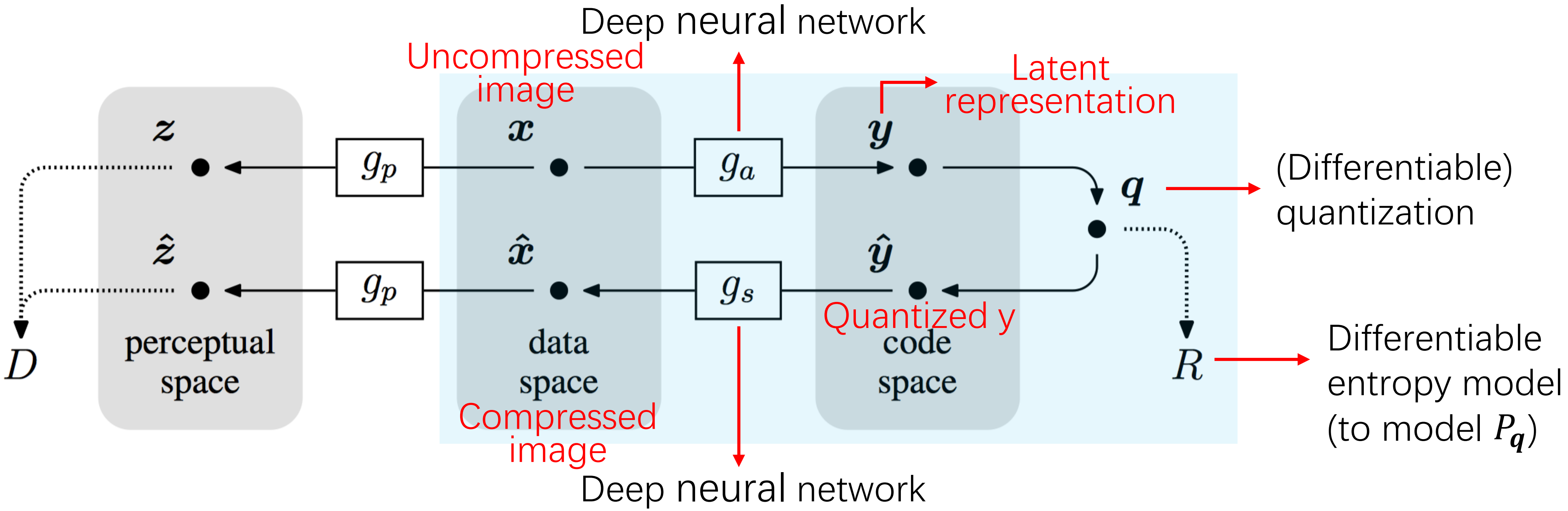
0.10001010 **8 bits**
(0.5390625 decimal)

Adaptive arithmetic coding:

Changing the frequency (or probability) tables while processing the data.

Learned Image Compression

- Basic architecture [1]: **End-to-end trainable**

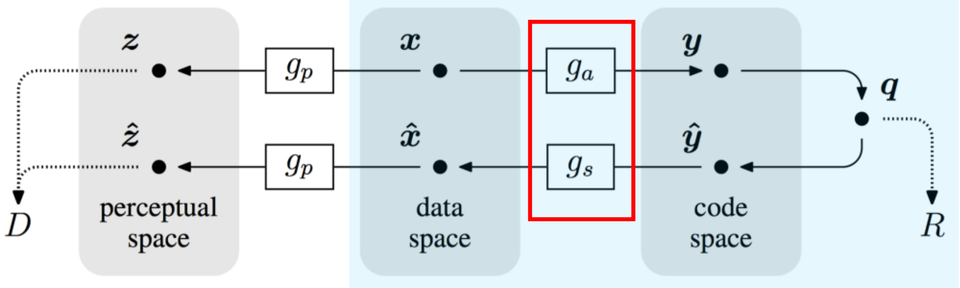


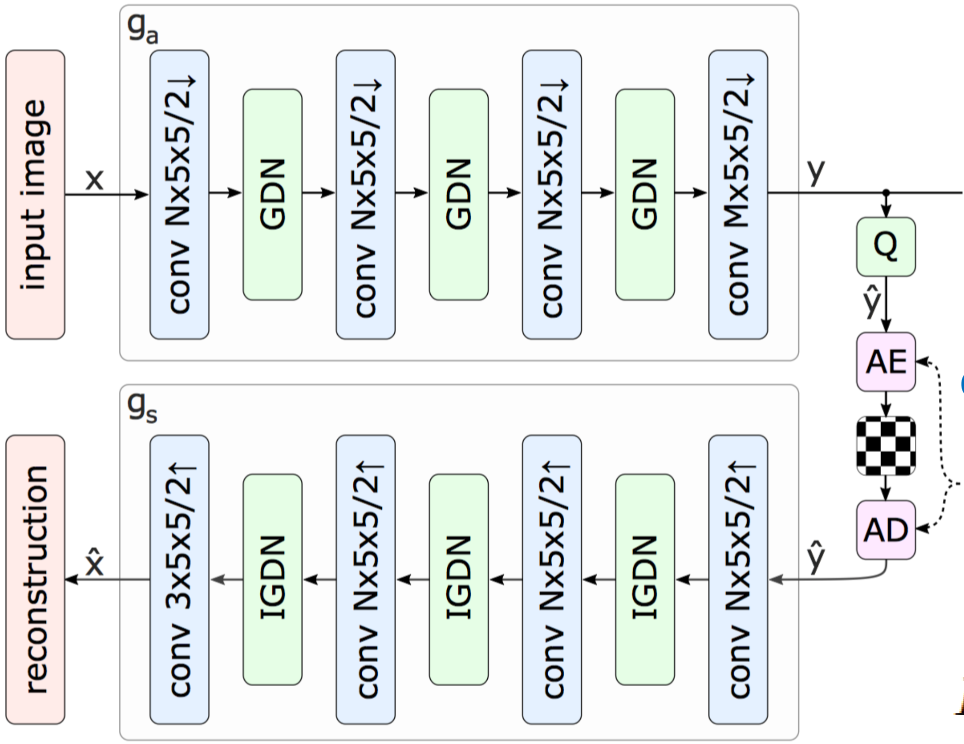
$$L[g_a, g_s, P_q] = \underbrace{-\mathbb{E}[\log_2 P_q]}_{R} + \lambda \mathbb{E}[d(x, \hat{x})]$$

[1] Ballé, Johannes, et al. "End-to-end optimized image compression." in ICLR. 2017.

Learned Image Compression

- CNN transformer + **factorized** entropy model [1]





$$\tilde{\boldsymbol{y}} = \boldsymbol{y} + \Delta \boldsymbol{y}^{\sim \mathcal{U}(0,1)}$$ **Training: differentiable quantization**

$$\hat{\boldsymbol{y}} = \mathrm{round}(\boldsymbol{y})$$ **Inference: quantization (not differentiable)**

differentiable entropy

$$c = f_K \circ f_{K-1} \cdots f_1$$
$$p = f'_K \cdot f'_{K-1} \cdots f'_1$$

$\tilde{\boldsymbol{y}}$ or $\hat{\boldsymbol{y}}$

$$f_k(\boldsymbol{x}) = g_k\left(\boldsymbol{H}^{(k)}\boldsymbol{x} + \boldsymbol{b}^{(k)}\right) \qquad 1 \le k < K$$
$$f_K(\boldsymbol{x}) = \mathrm{sigmoid}\left(\boldsymbol{H}^{(K)}\boldsymbol{x} + \boldsymbol{b}^{(K)}\right)$$
$$g_k(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{a}^{(k)} \odot \tanh(\boldsymbol{x})$$
$$g'_k(\boldsymbol{x}) = 1 + \boldsymbol{a}^{(k)} \odot \tanh'(\boldsymbol{x})$$

$$\boldsymbol{H}^{(k)} = \mathrm{softplus}(\hat{\boldsymbol{H}}^{(k)})$$
$$\boldsymbol{a}^{(k)} = \tanh(\hat{\boldsymbol{a}}^{(k)})$$

$$R = \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}\left[-\log_2 p_{\hat{\boldsymbol{y}}}\left(Q(g_a(\boldsymbol{x};\phi_g))\right)\right]$$ **estimated bit-rate**

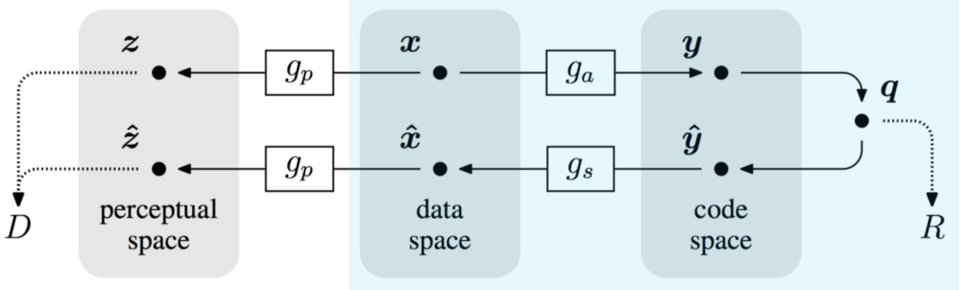
$$L(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{x}, \Delta \boldsymbol{y}}\Big[\underbrace{-\sum_i \log_2 p_{\tilde{y}_i}(g_a(\boldsymbol{x};\boldsymbol{\phi}) + \Delta \boldsymbol{y}; \boldsymbol{\psi}^{(i)})}_{\text{bit-rate}} + \underbrace{\boxed{\lambda}}_{\text{trade-off}} \underbrace{d\Big(g_p(g_s(g_a(\boldsymbol{x};\boldsymbol{\phi}) + \Delta \boldsymbol{y}; \boldsymbol{\theta})), g_p(\boldsymbol{x})\Big)}_{\text{distortion}}\Big]$$

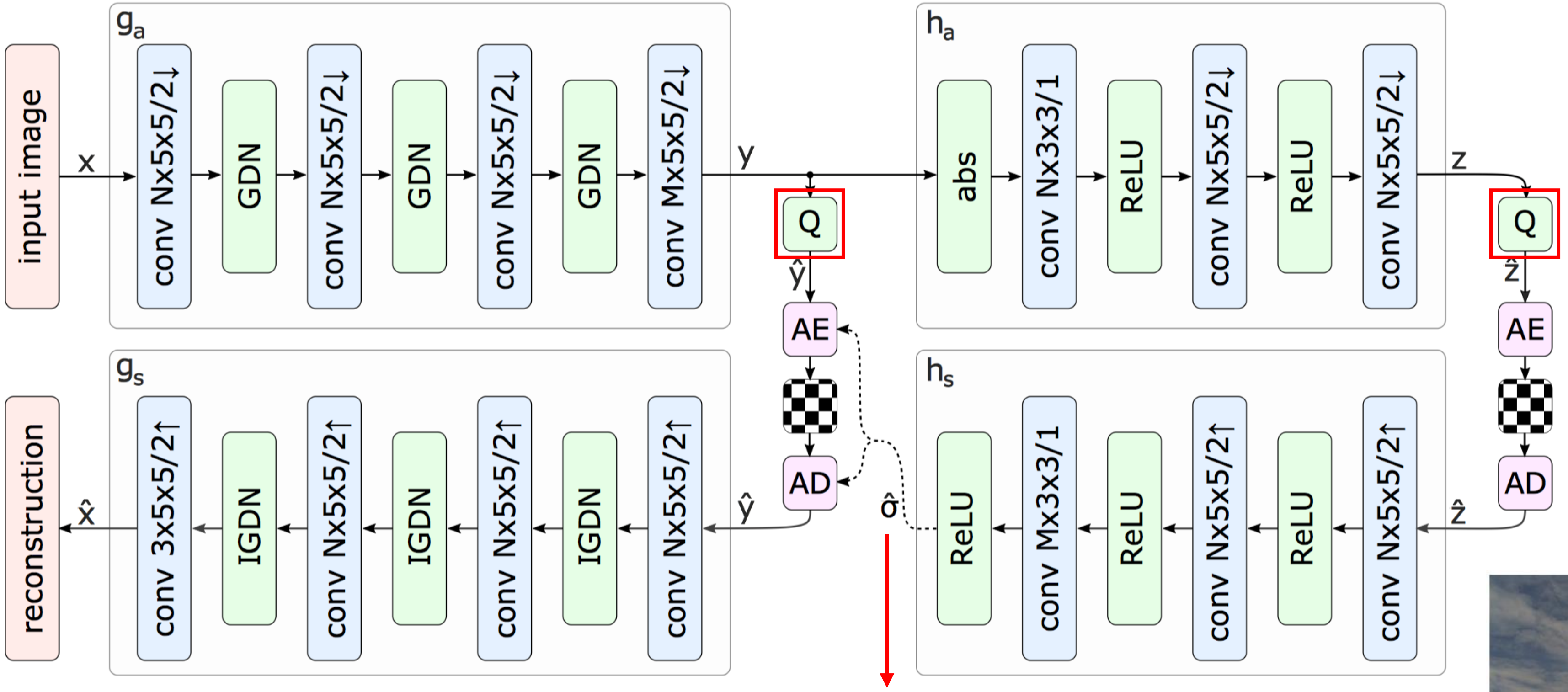
Optimized in an end-to-end manner

[1] Ballé, Johannes, et al. "End-to-end optimized image compression." in ICLR. 2017.

Learned Image Compression

- CNN transformer + **hyperpiror** entropy model [2]





differentiable quantization

differentiable entropy model

$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{z}}) \longleftrightarrow p_{\hat{y}_i}(\hat{y}_i \mid \hat{\sigma}_i) = \int_{\hat{y}_i-1/2}^{\hat{y}_i+1/2} \mathcal{N}(y \mid 0, \hat{\sigma}_i)\, \mathrm{d}y$$

discretized Gaussian distribution

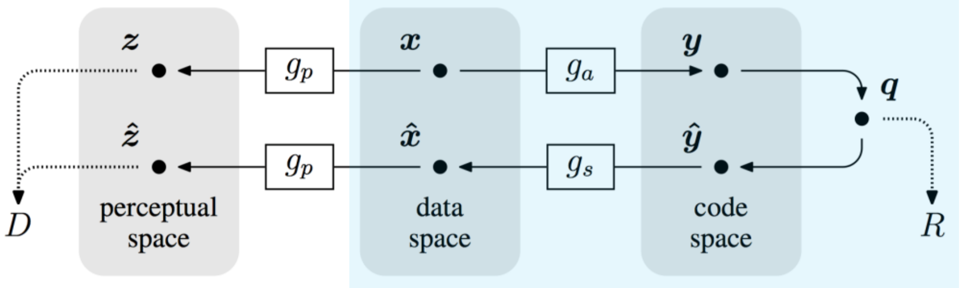
less dependency (less redundancy)

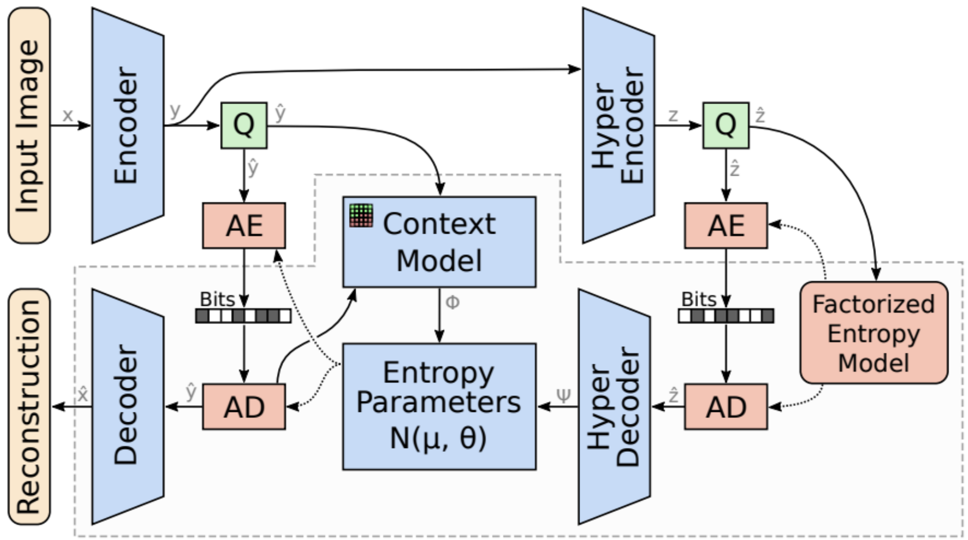
$\boldsymbol{y}$ (factorized) $\quad$ $\boldsymbol{y}/\boldsymbol{\sigma}$ (hyperprior)

[2] Ballé, Johannes, et al. "Variational image compression with a scale hyperprior." in ICLR. 2018.

Learned Image Compression

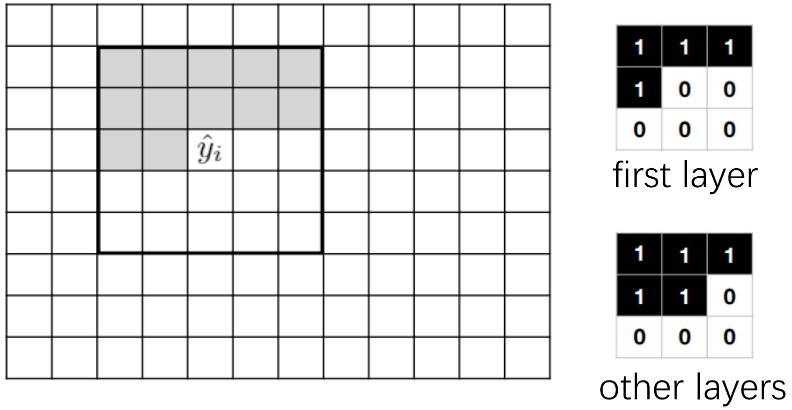
- CNN transformer + **autoregressive** entropy model [3]





Component	Symbol
Input Image	$\boldsymbol{x}$
Encoder	$f(\boldsymbol{x};\boldsymbol{\theta}_e)$
Latents	$\boldsymbol{y}$
Latents (quantized)	$\hat{\boldsymbol{y}}$
Decoder	$g(\hat{\boldsymbol{y}};\boldsymbol{\theta}_d)$
Hyper Encoder	$f_h(\boldsymbol{y};\boldsymbol{\theta}_{he})$
Hyper-latents	$\boldsymbol{z}$
Hyper-latents (quant.)	$\hat{\boldsymbol{z}}$
Hyper Decoder	$g_h(\hat{\boldsymbol{z}};\boldsymbol{\theta}_{hd})$
Context Model	$g_{cm}(\boldsymbol{y}_{<i};\boldsymbol{\theta}_{cm})$
Entropy Parameters	$g_{ep}(\cdot;\boldsymbol{\theta}_{ep})$
Reconstruction	$\hat{\boldsymbol{x}}$

Mask CNN [4]



Algorithm 1 Constructing 3D Masks

$central\_idx \leftarrow \lceil (f_W \cdot f_H \cdot f_D)/2 \rceil$
$current\_idx \leftarrow 1$
$mask \leftarrow f_W \times f_H \times f_D$-dimensional matrix of zeros
for $d \in \{1,\dots,f_D\}$ **do**
  for $h \in \{1,\dots,f_H\}$ **do**
    for $w \in \{1,\dots,f_W\}$ **do**
      if $current\_idx < central\_idx$ **then**
        $mask(w,h,d) = 1$
      else
        $mask(w,h,d) = 0$
      $current\_idx \leftarrow current\_idx + 1$

Due to the chain rule: $p(\boldsymbol{y}) = p(y_1)\cdot p(y_2|y_1)\cdot p(y_3|y_2,y_1)\dots p(y_N|y_{<N})$

$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}\mid\hat{\boldsymbol{z}}) = \prod_{i=1}^{N} \underline{p_{\hat{y}_i|\hat{y}_{<i},\hat{\boldsymbol{z}}}(\hat{y}_i\mid\hat{y}_{<i},\hat{\boldsymbol{z}})}$$

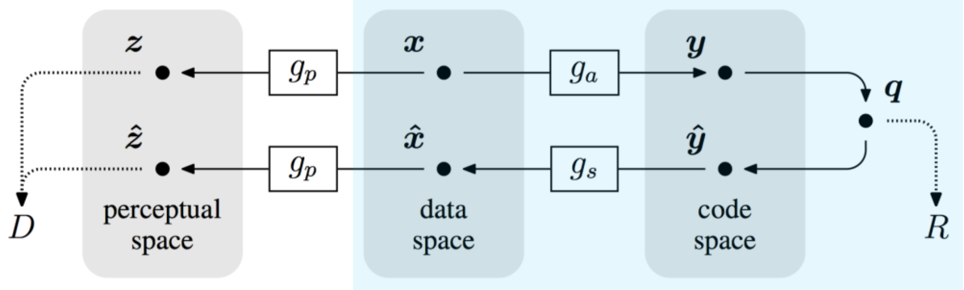
$$p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}}\mid\hat{\boldsymbol{z}},\boldsymbol{\theta}_{hd},\boldsymbol{\theta}_{cm},\boldsymbol{\theta}_{ep}) = \prod_i \underline{\Big(\mathcal{N}(\mu_i,\sigma_i^2) * \mathcal{U}(-\tfrac{1}{2},\tfrac{1}{2})\Big)(\hat{y}_i)}$$

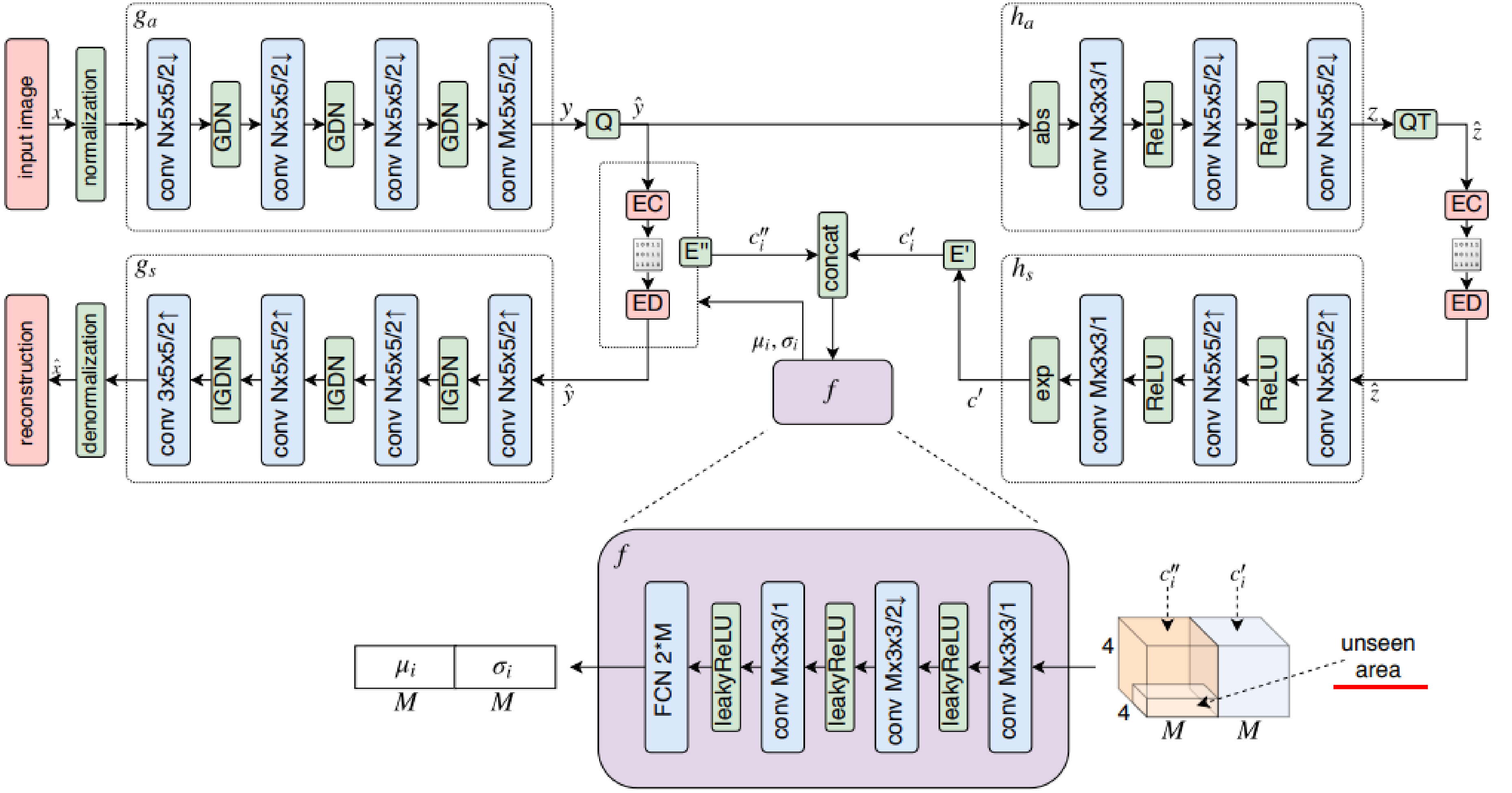
with $\mu_i,\sigma_i = g_{ep}(\boldsymbol{\psi},\boldsymbol{\phi}_i;\boldsymbol{\theta}_{ep}), \boldsymbol{\psi} = g_h(\hat{\boldsymbol{z}};\boldsymbol{\theta}_{hd})$, and $\boldsymbol{\phi}_i = g_{cm}(\underline{\hat{\boldsymbol{y}}_{<i}};\boldsymbol{\theta}_{cm})$

[3] Minnen, David, et al. "Joint autoregressive and hierarchical priors for learned image compression." in NeruIPS. 2018.
[4] Mentzer, Fabian, et al. "Conditional Probability Models for Deep Image Compression", in CVPR, 2018.

Learned Image Compression

- CNN transformer + **autoregressive** entropy model [5]





[5] Lee, Jooyoung, et al. "Context-adaptive Entropy Model for End-to-end Optimized Image Compression." in ICLR. 2019.

Learned Image Compression

$y_i \in$ {hot coffee, hot tea, cold coffee, cold tea} $\boldsymbol{y} = [y_1, y_2, y_3]$

- **Factorized** entropy model

$p_{y_i}(y_i) = 25\%$ for $y_i =$ hot coffee, hot tea, cold coffee, cold tea

$H\left(p_{y_i}\right) = 4 \times (-0.25 \log_2 0.25) = 2$

The expected number of bits to encode $\boldsymbol{y}$ is **6**

- **Hyperprior** entropy model $\boldsymbol{z} = [10℃, 15℃, 30℃]$

$p_{y_i|z_i}(y_i|z_i < 20℃) = 50\%$ for $y_i =$ hot coffee, hot tea $H = 2 \times (-0.5 \log_2 0.5) = 1$

$p_{y_i|z_i}(y_i|z_i \geq 20℃) = 50\%$ for $y_i =$ cold coffee, cold tea $H = 1$

The expected number of bits to encode $\boldsymbol{y}$ is **3**

- **Autoregressive** entropy model (joint with hyperprior)

$p_{y_i|y_{i-1}, z_i}(y_i|y_{i-1}, z_i)$

Don't drink coffee (or tea) in two consecutive days.

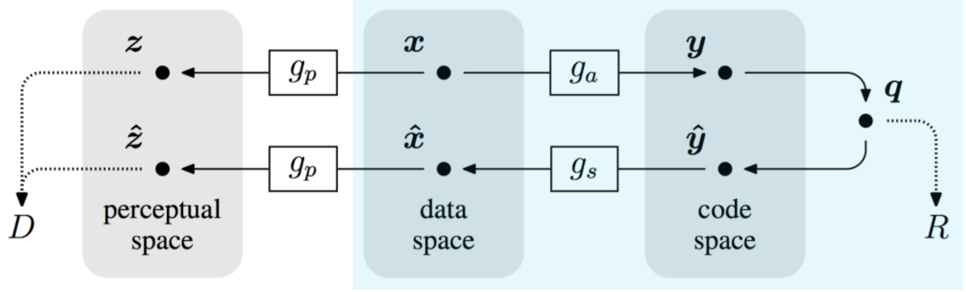
$\boldsymbol{z} = [10℃, 15℃, 30℃]$

$p(\boldsymbol{y} =$ [hot coffee, hot tea, cold coffee]$) = 0.5$

$p(\boldsymbol{y} =$ [hot tea, hot coffee, cold tea]$) = 0.5$

The expected number of bits to encode $\boldsymbol{y}$ is $H(\boldsymbol{y}) = 2 \times (-0.5 \log_2 0.5) =$ **1**

Learned Image Compression



- Another differentiable quantization method [4]

given centers $\mathcal{C} = \{c_1, \cdots, c_L\}$

$$\hat{z}_i = Q(z_i) := \arg\min_j \|z_i - c_j\|$$

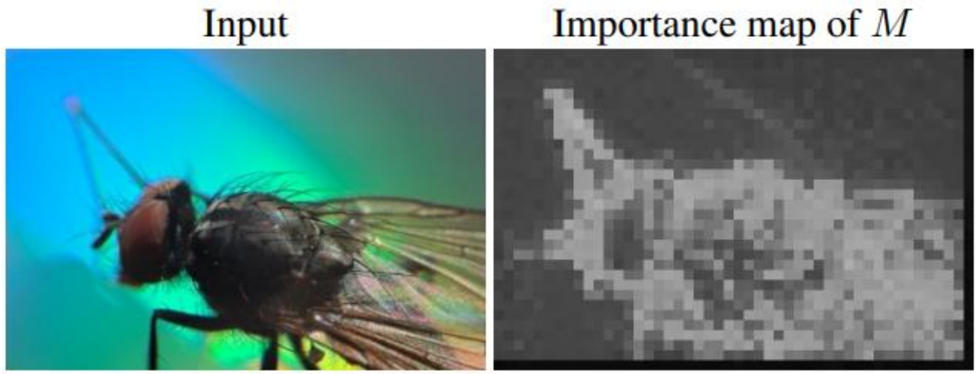
Inference

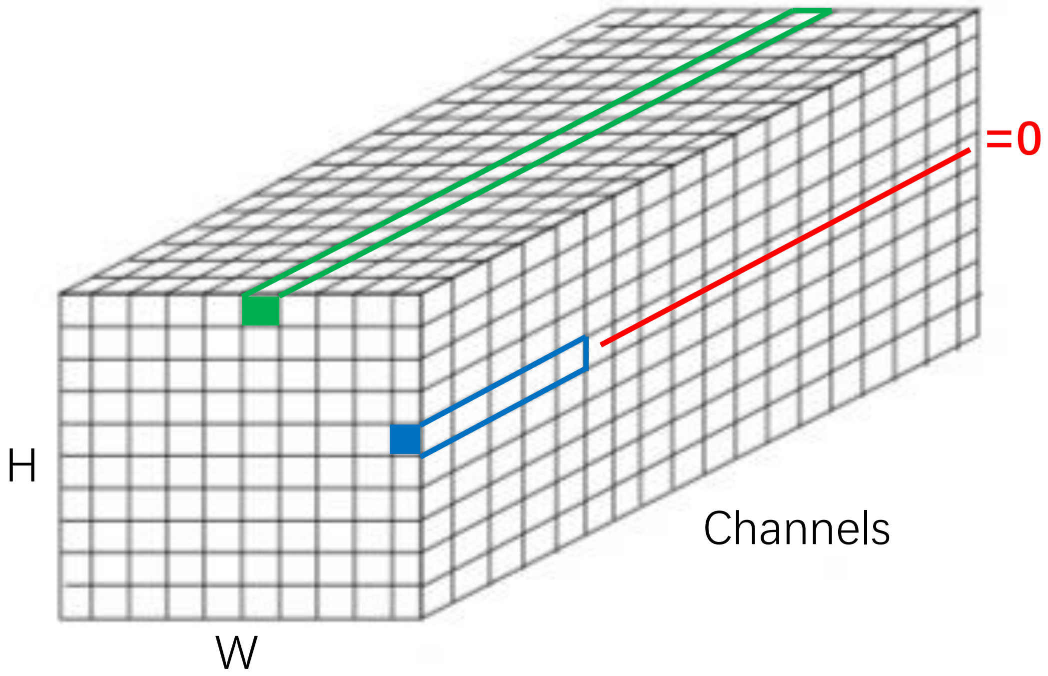
$$\tilde{z}_i = \sum_{j=1}^{L} \frac{\exp(-\sigma\|z_i - c_j\|)}{\sum_{l=1}^{L} \exp(-\sigma\|z_i - c_l\|)} c_j$$

Training: differentiable

$$\bar{z}_i = \text{tf.stopgradient}(\hat{z}_i - \tilde{z}_i) + \tilde{z}_i$$

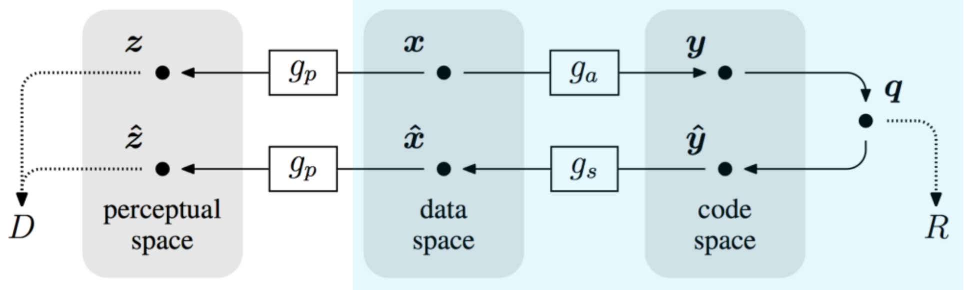
- Importance map [4]





[4] Mentzer, Fabian, et al. "Conditional Probability Models for Deep Image Compression", in CVPR, 2018.

Learned Image Compression



- Another differentiable quantization method [4]

given centers $\mathcal{C} = \{c_1, \cdots, c_L\}$

$$\hat{z}_i = Q(z_i) := \arg\min_j \|z_i - c_j\|$$

Inference

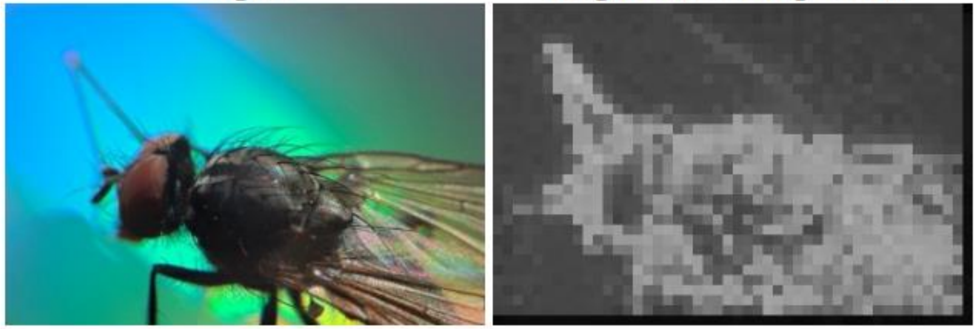
$$\tilde{z}_i = \sum_{j=1}^{L} \frac{\exp(-\sigma\|z_i - c_j\|)}{\sum_{l=1}^{L} \exp(-\sigma\|z_i - c_l\|)} c_j$$

Training: differentiable

$$\bar{z}_i = \text{tf.stopgradient}(\hat{z}_i - \tilde{z}_i) + \tilde{z}_i$$

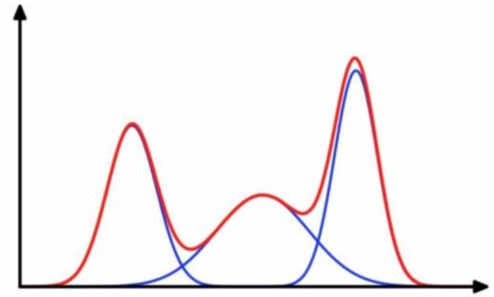
- Importance map [4]

Input　　Importance map of M



- Gaussian Mixture Model (GMM) for entropy [6]

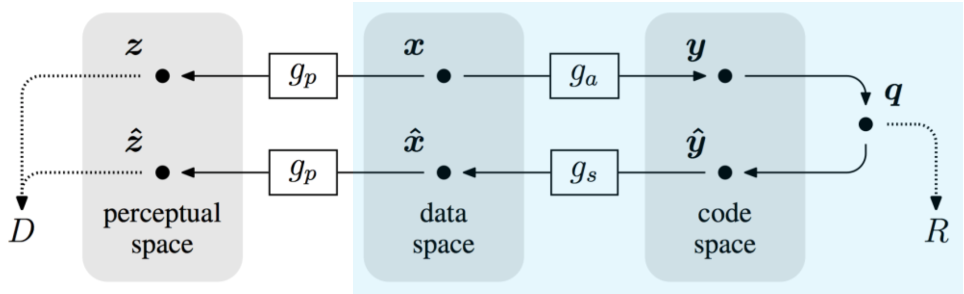
$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) \sim \sum_{k=1}^{K} \boldsymbol{w}^{(k)} \mathcal{N}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{2(k)})$$

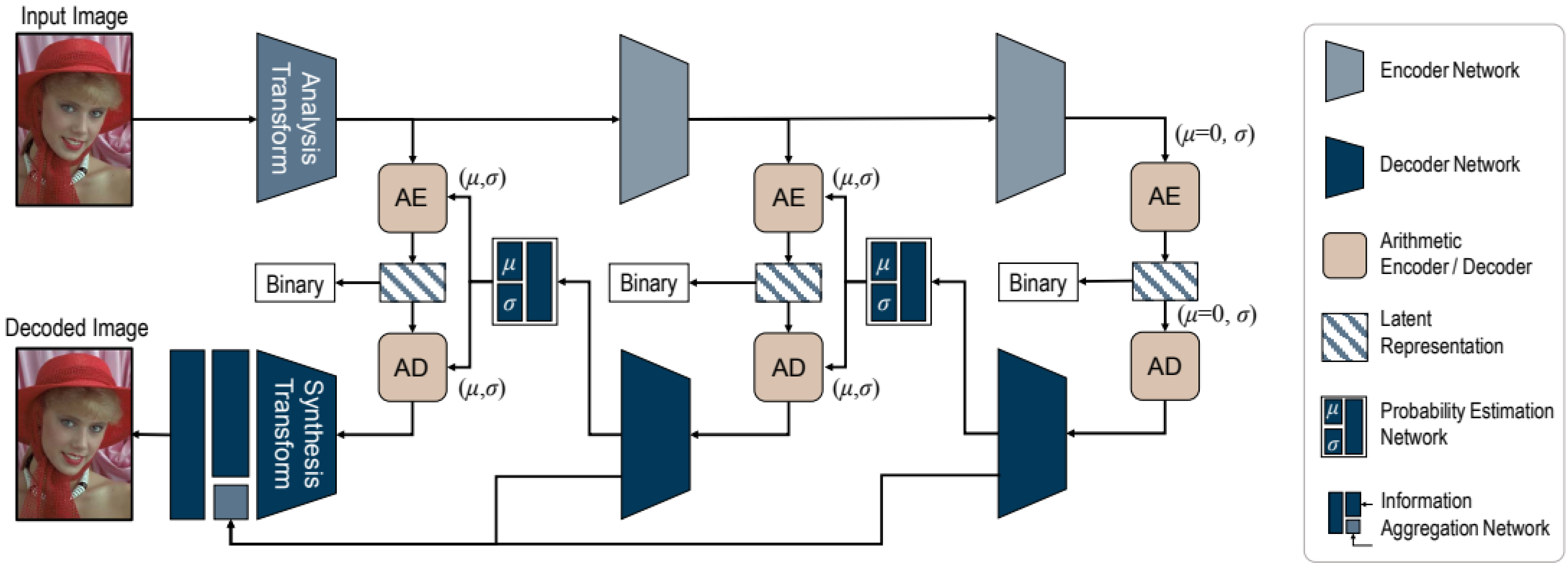


[4] Mentzer, Fabian, et al. “Conditional Probability Models for Deep Image Compression”, in CVPR, 2018.
[6] Cheng et al. ”Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules”, in CVPR. 2020.

Learned Image Compression

- CNN transformer + **coarse-to-fine** model [7]

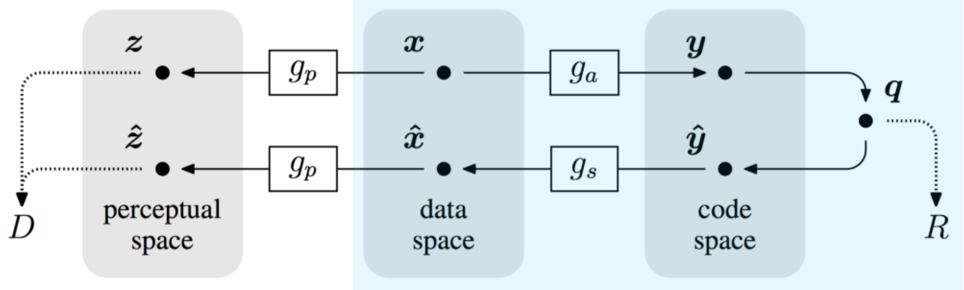


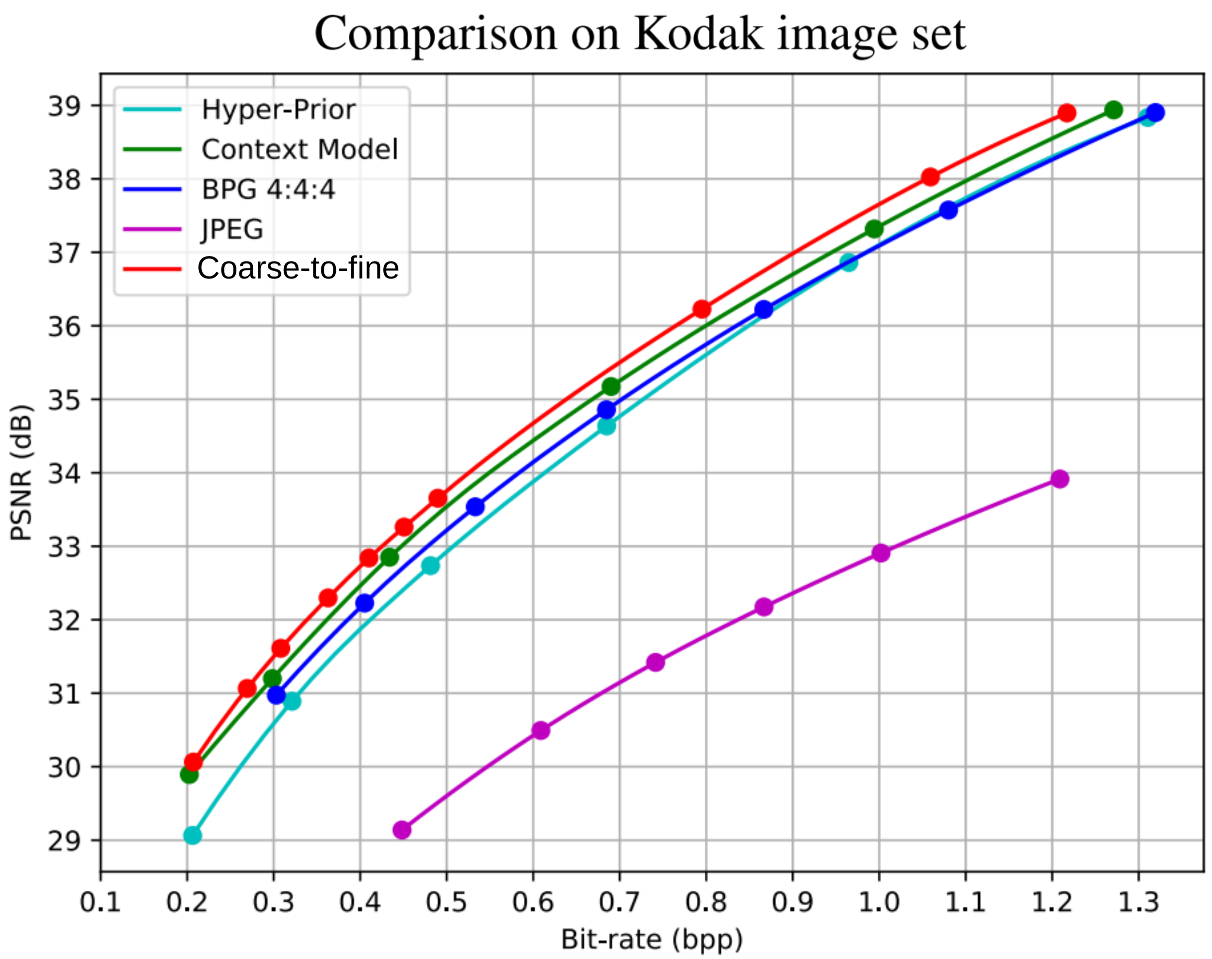


[7] Hu, Yueyu, et al. "Coarse-to-Fine Hyper-Prior Modeling for Learned Image Compression." in AAAI. 2020.

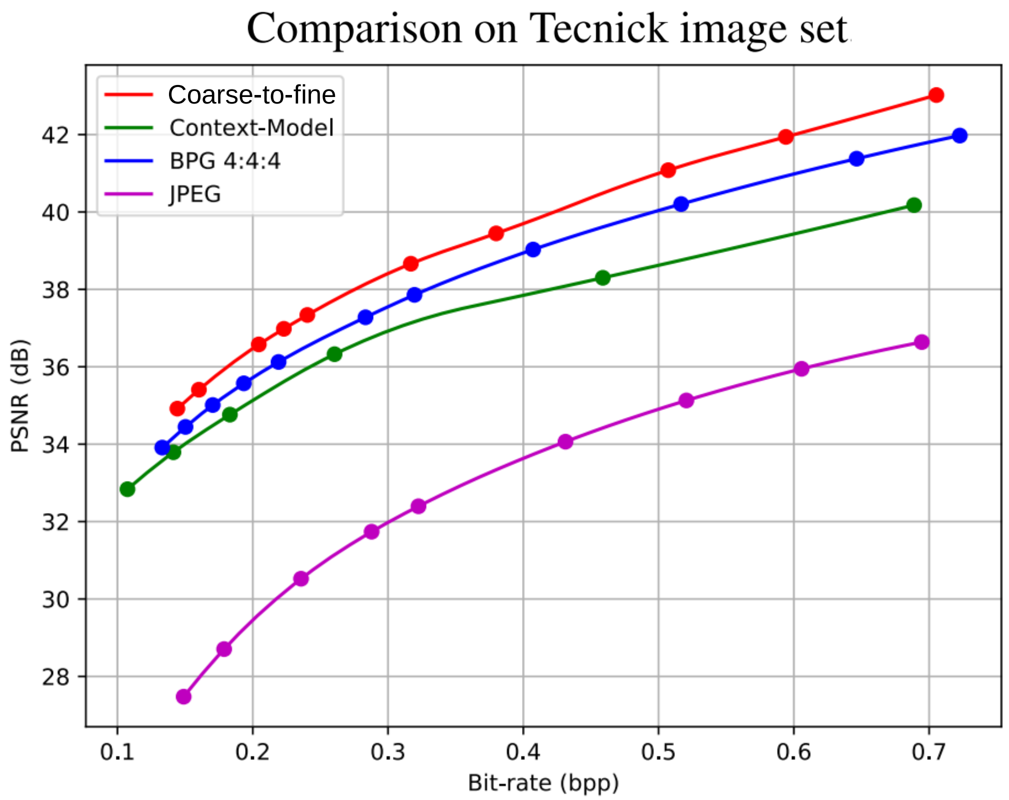
Learned Image Compression

- Performance





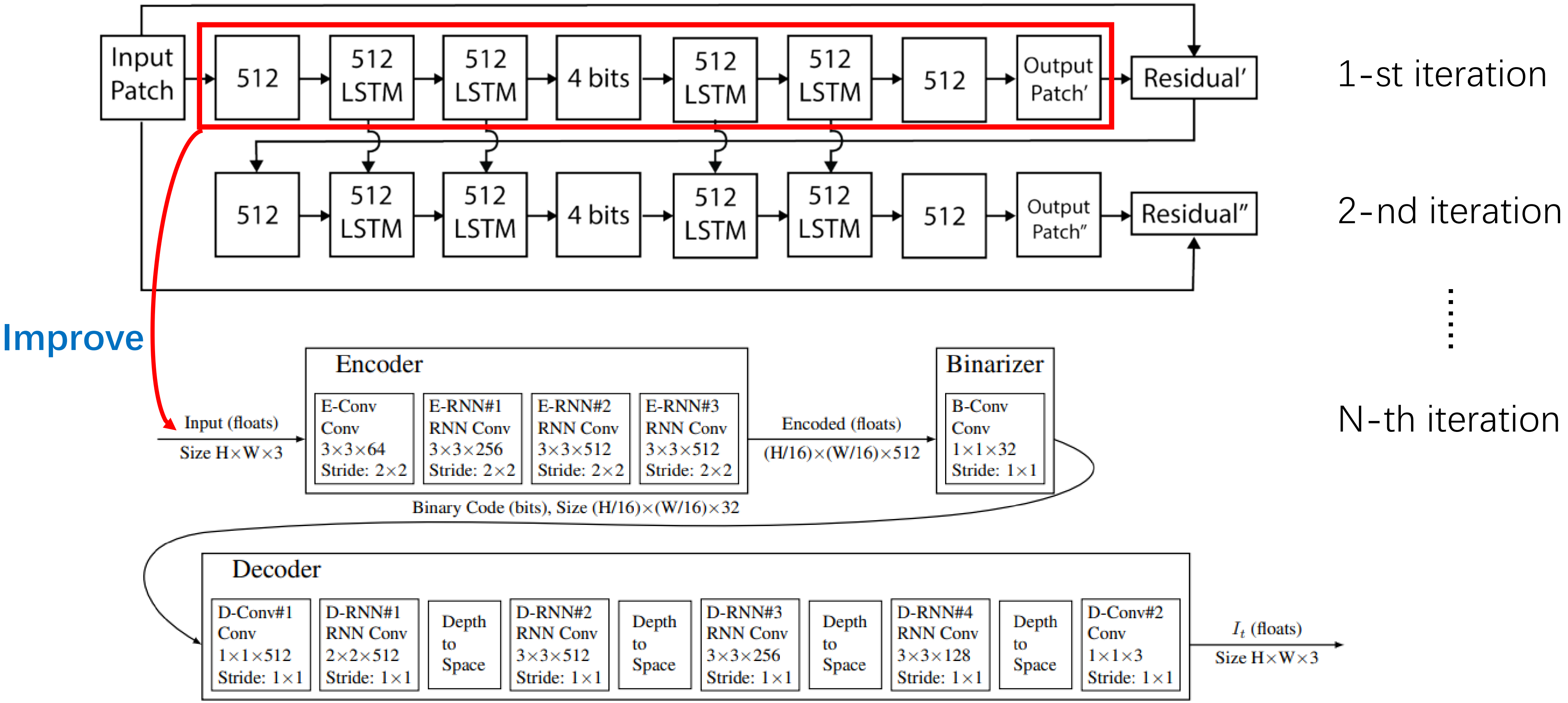
The context (autoregressive) and coarse-to-fine models outperform BPG 4:4:4 (latest traditional standard)



The rank may vary on different datasets

Learned Image Compression

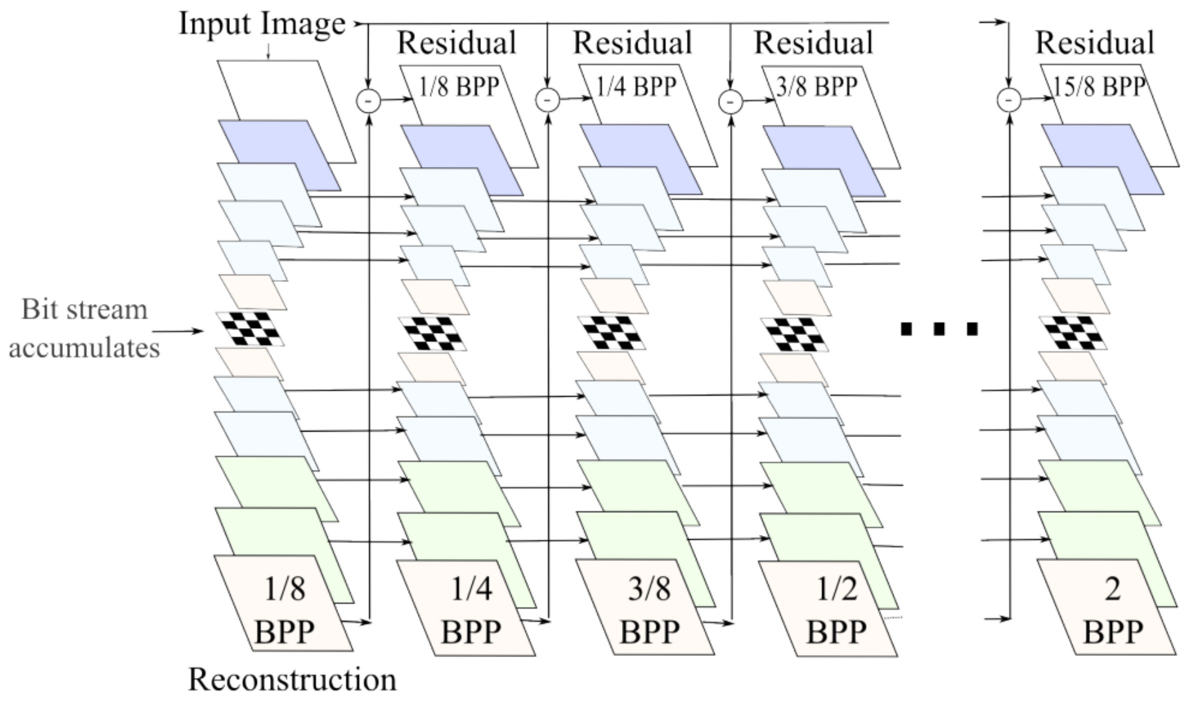
- Variable rate image compression: RNN-based methods [8, 9]



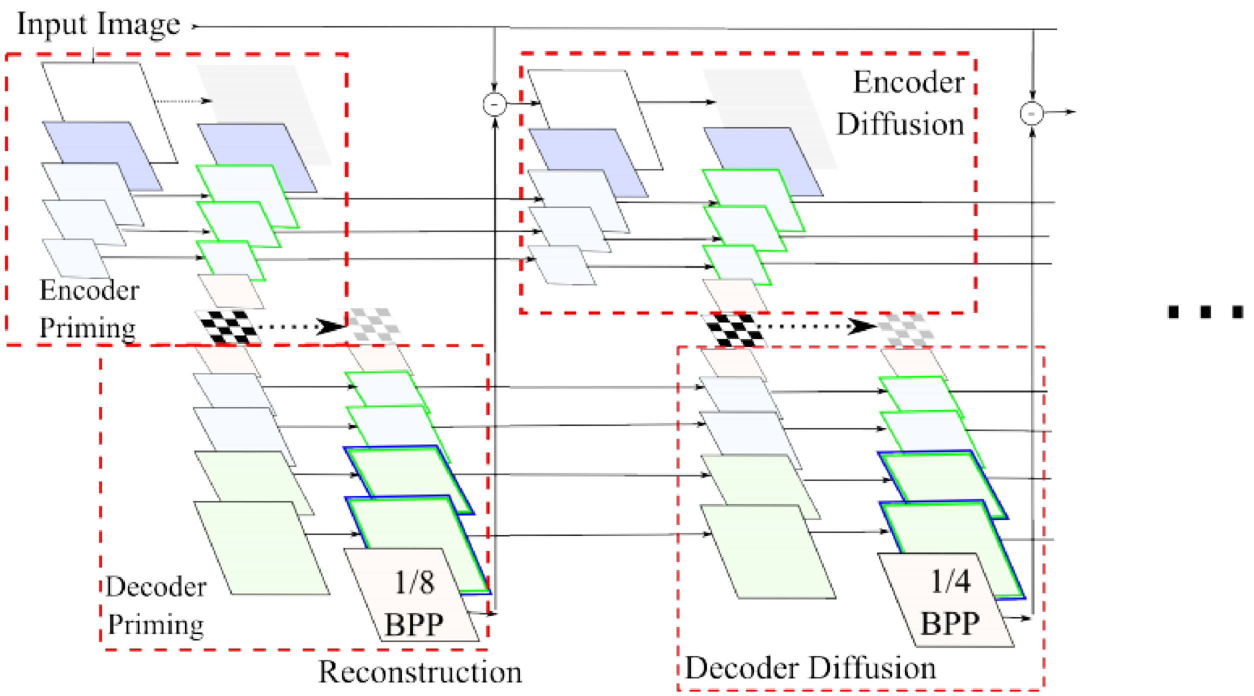
[8] Toderici, George, et al. "Variable Rate Image Compression with Recurrent Neural Networks." in ICLR. 2016.
[9] Toderici, George, et al. "Full Resolution Image Compression with Recurrent Neural Networks." in CVPR, 2017.

Learned Image Compression

- Variable rate image compression: RNN-based methods [10]



Basic framework



Increasing the depth of neural network

[10] Johnston, Nick, et al. "Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks." in CVPR. 2018.

Learned Image Compression

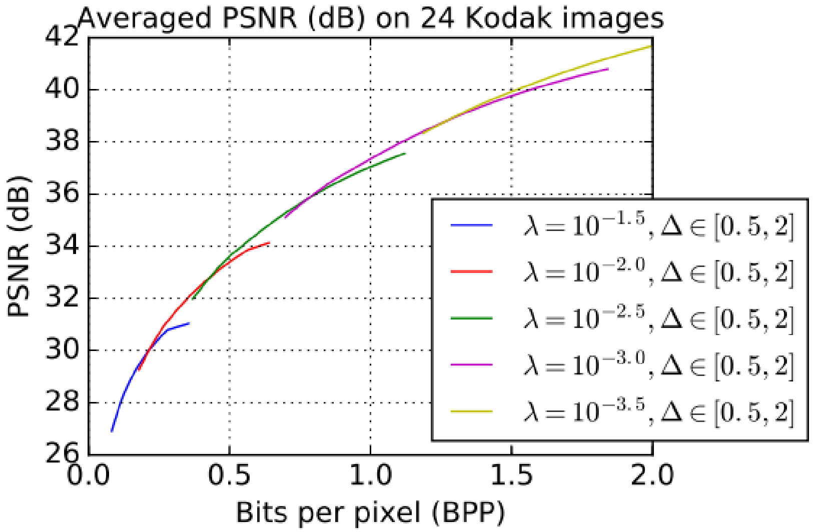
- Variable rate image compression: Conditional autoencoder [11]

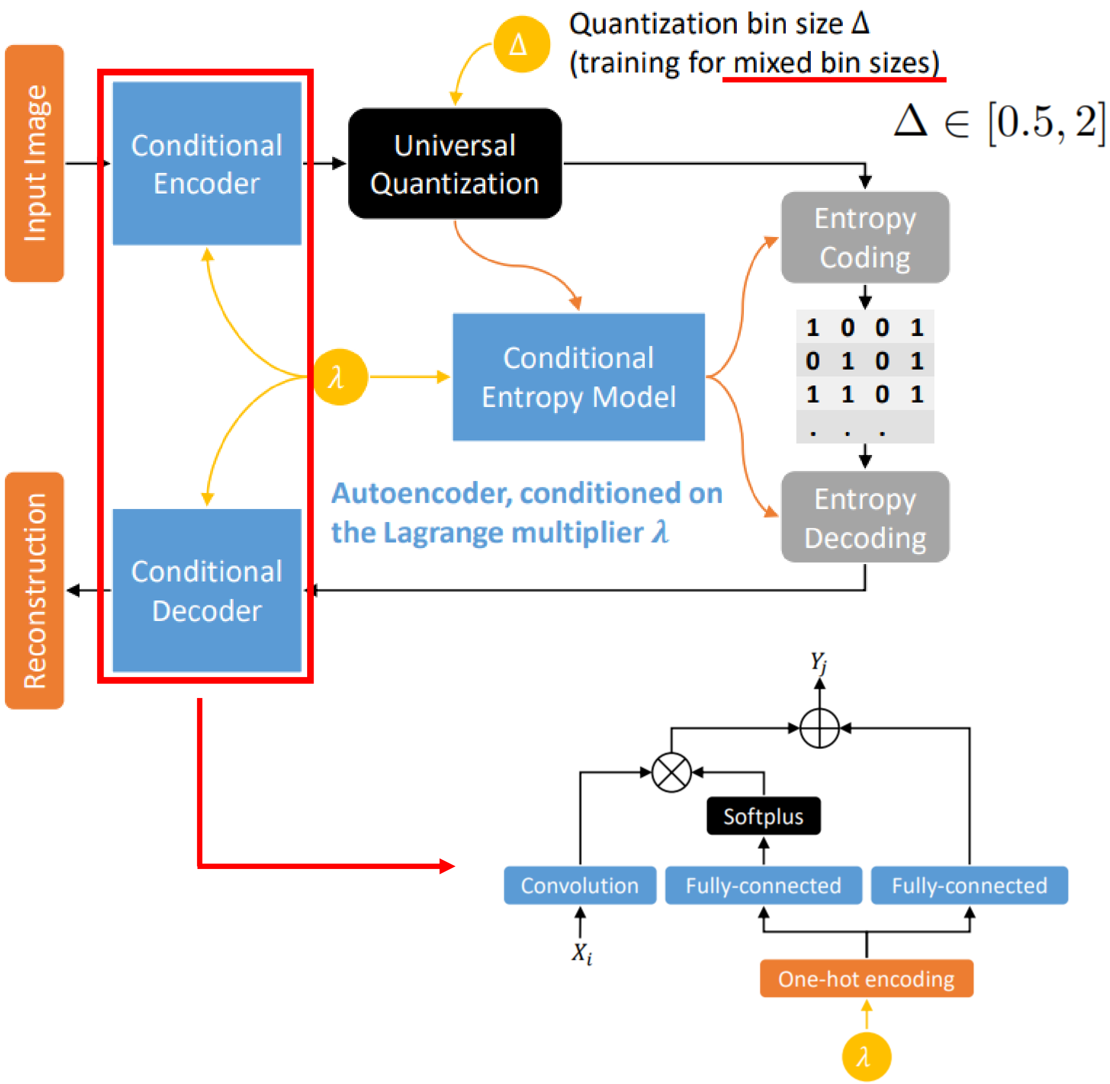
Loss function:

$$\min_{\phi,\theta}\{D_{\phi,\theta} + \lambda R_{\phi}\}$$

$$\min_{\phi,\theta}\sum_{\lambda\in\Lambda}\left(D_{\phi,\theta}(\lambda) + \lambda R_{\phi,\theta}(\lambda)\right)$$

$$\min_{\phi,\theta}\sum_{\lambda\in\Lambda}\mathbb{E}_{p(\Delta)}[D_{\phi,\theta}(\lambda,\Delta) + \lambda R_{\phi,\theta}(\lambda,\Delta)]$$

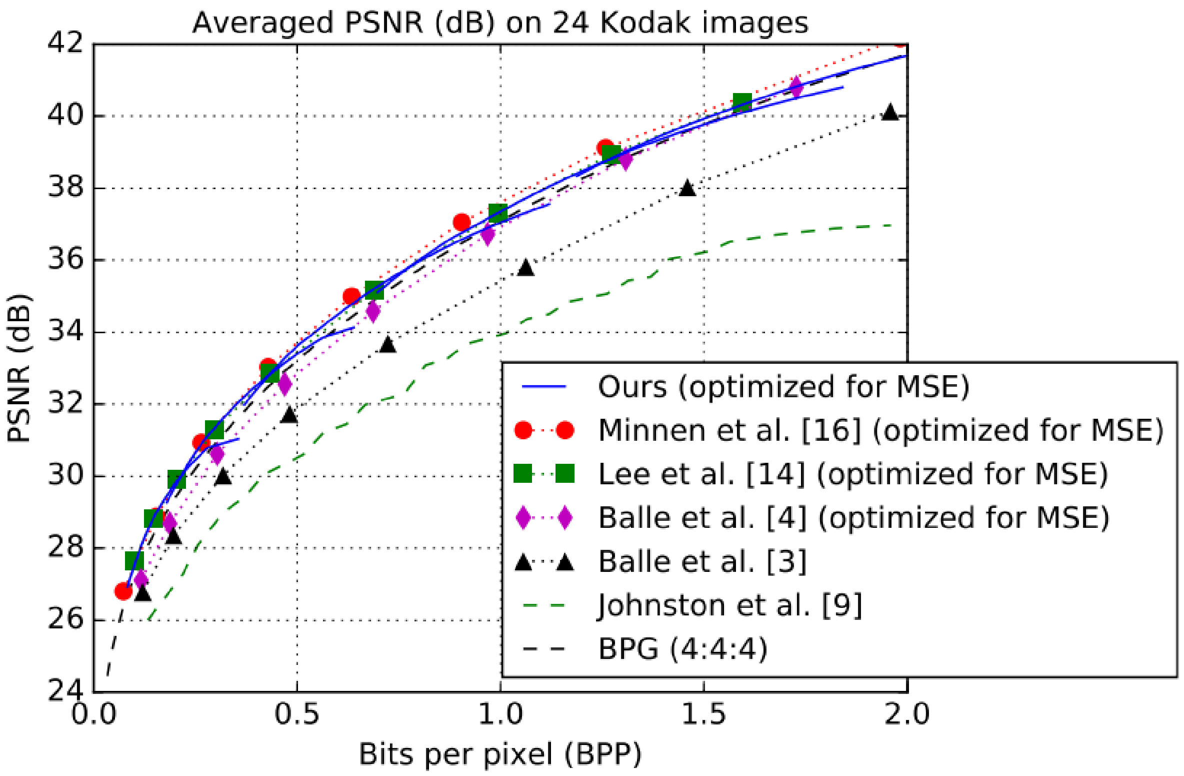


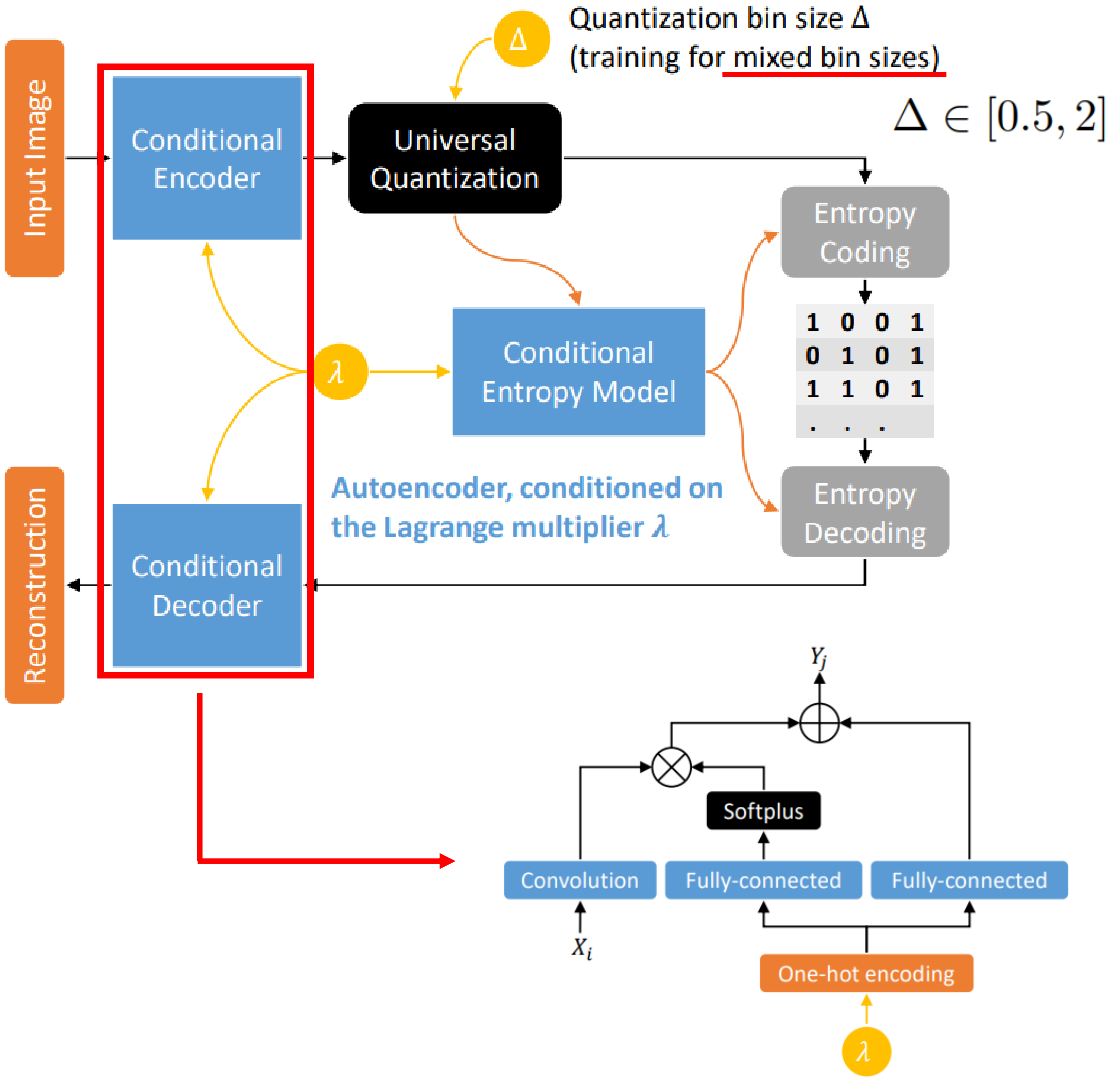


[11] Choi, Yoojin, et al. “Variable Rate Deep Image Compression With a Conditional Autoencoder.” in ICCV. 2019.

Learned Image Compression

- Variable rate image compression: Conditional autoencoder [11]

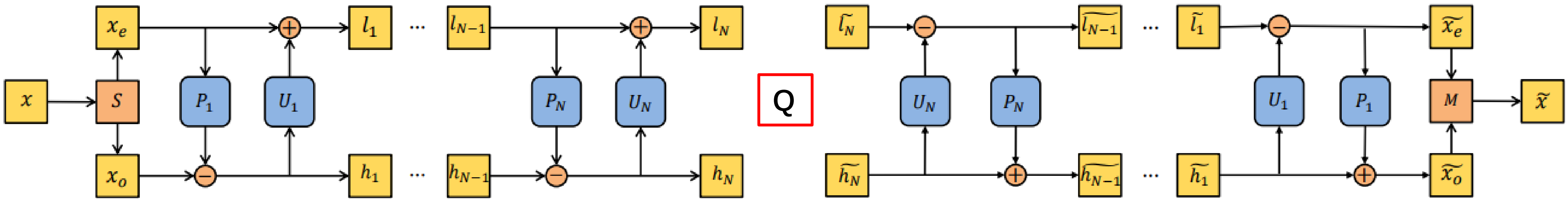




[11] Choi, Yoojin, et al. "Variable Rate Deep Image Compression With a Conditional Autoencoder." in ICCV. 2019.

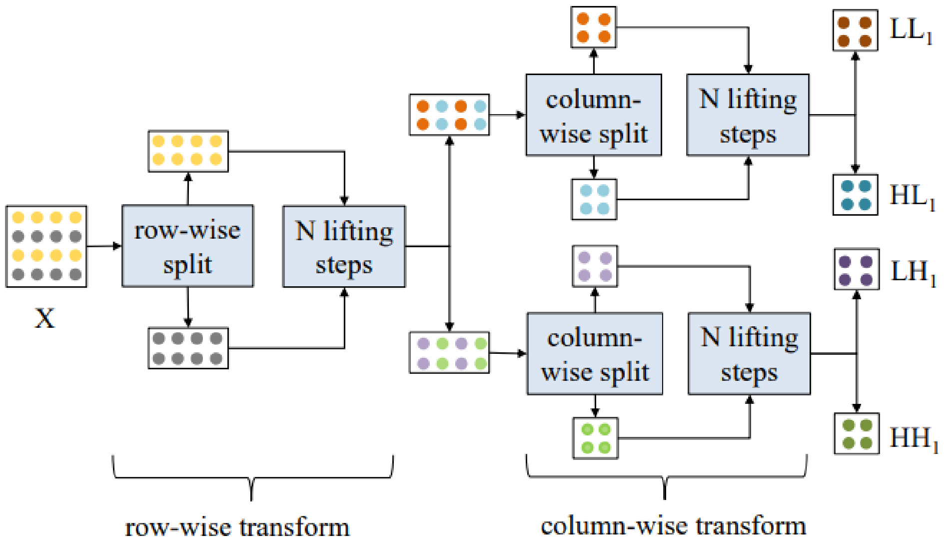
Learned Image Compression

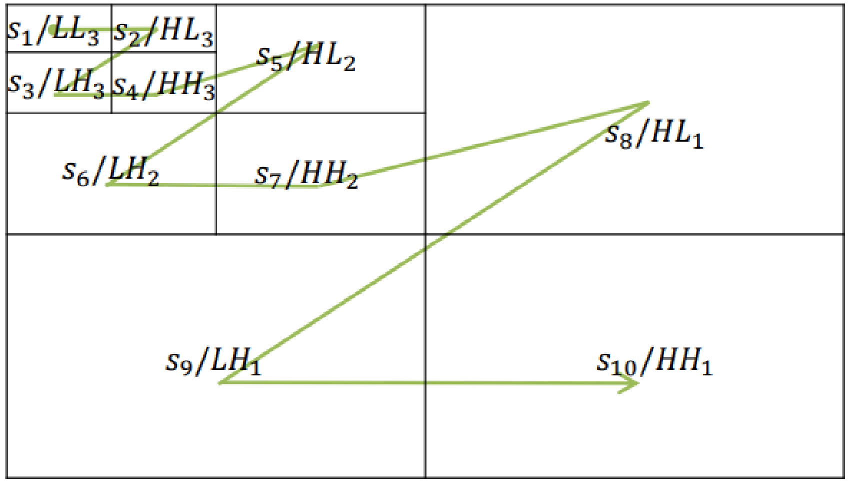
- Variable rate image compression: Wavelet-like transformer [12]



(a) Forward transform (b) Inverse transform

Invertible: achieving lossy and lossless compression by the same framework

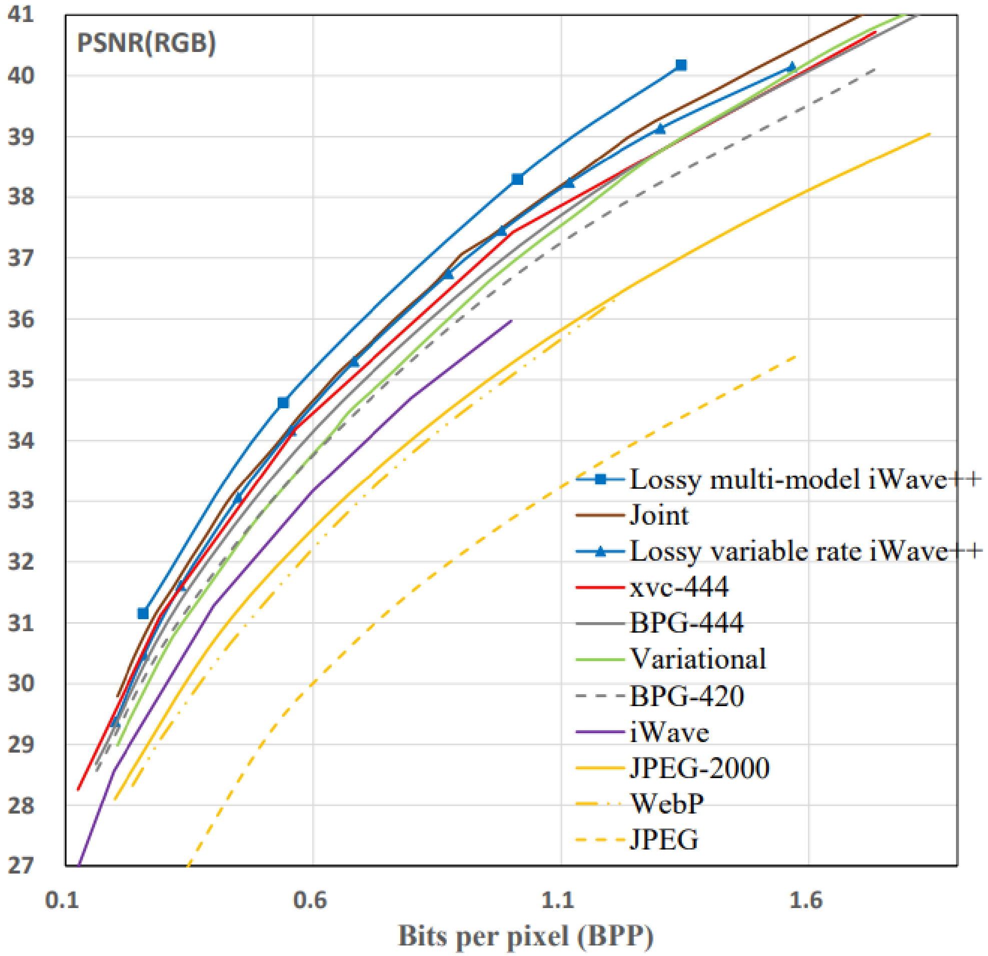




[12] Ma, Haichuan, et al. "End-to-End Optimized Versatile Image Compression With Wavelet-Like Transform." in IEEE T-PAMI. 2020.

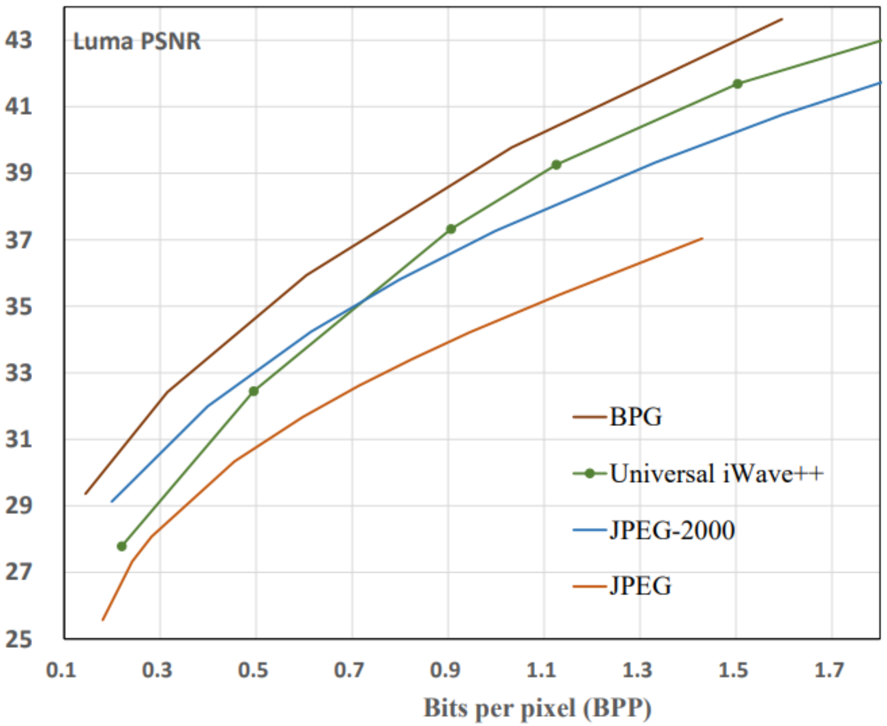
Learned Image Compression

- Variable rate image compression: Wavelet-like transformer [12]



(one model for each point)

(one model for all points)

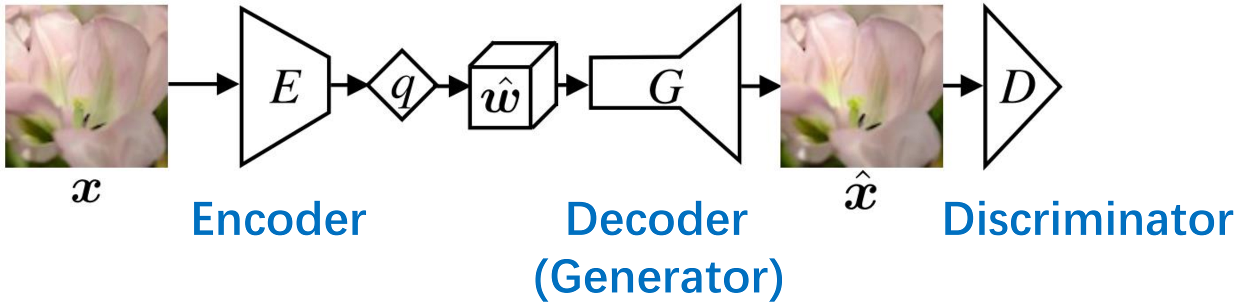


one model for both lossy and lossless compression

[12] Ma, Haichuan, et al. "End-to-End Optimized Versatile Image Compression With Wavelet-Like Transform." in IEEE T-PAMI. 2020.

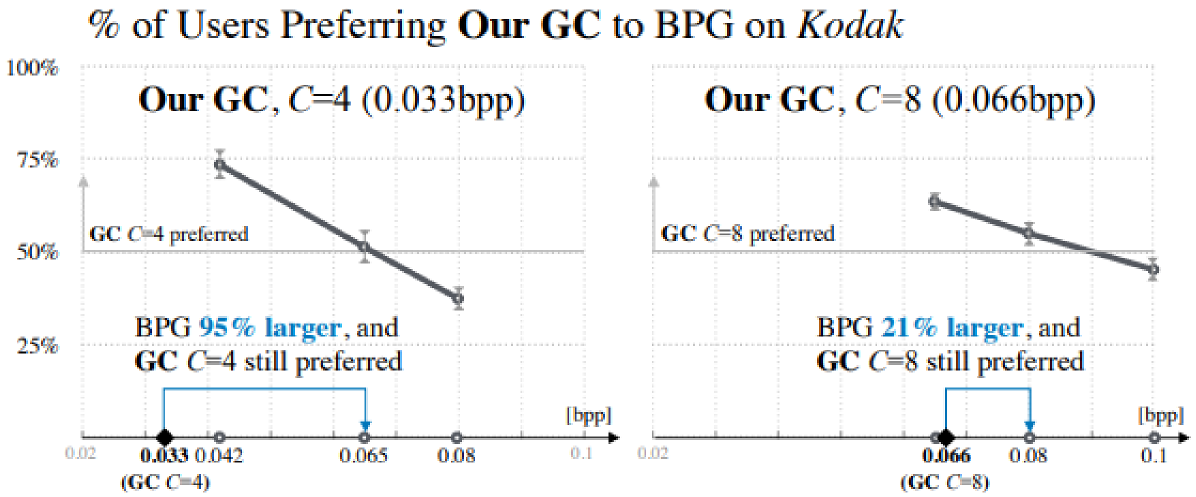
Learned Image Compression

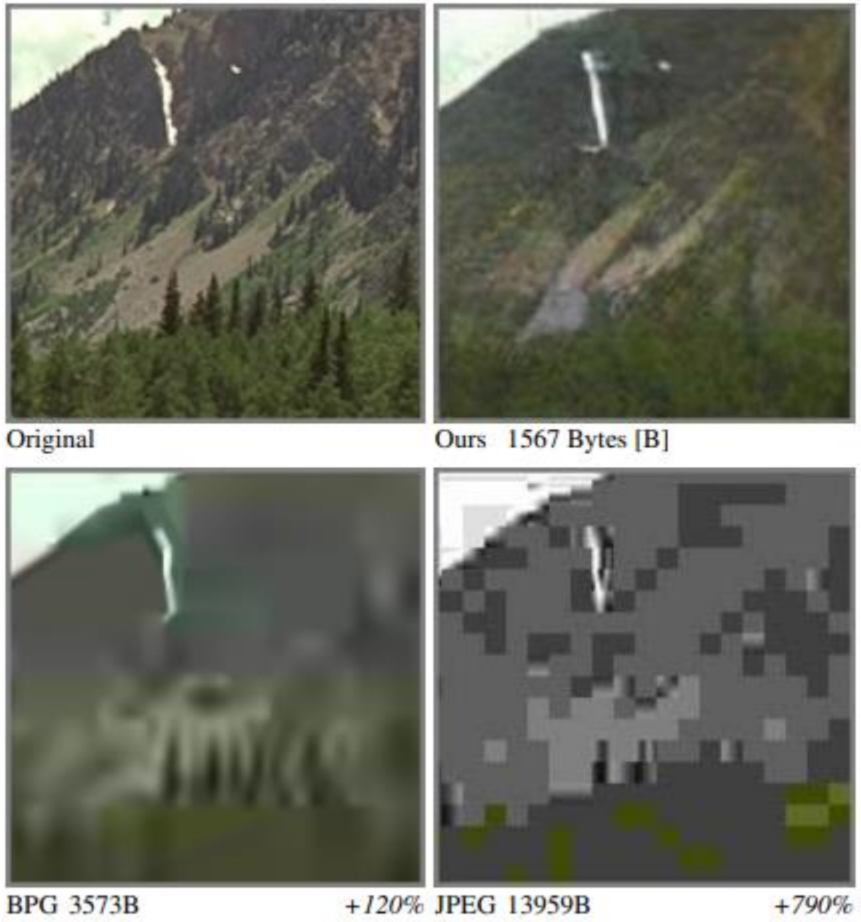
- Generative image compression: GAN-based methods [13]



$$\min_{E,G}\max_{D}\ \mathbb{E}[f(D(\hat{\boldsymbol{w}}))] + \mathbb{E}[g(D(G(\hat{\boldsymbol{w}})))]$$
GAN loss

$$+\ \lambda\mathbb{E}[d(\boldsymbol{x}, G(\hat{\boldsymbol{w}}))] + \beta H(\hat{\boldsymbol{w}}),$$
RD loss

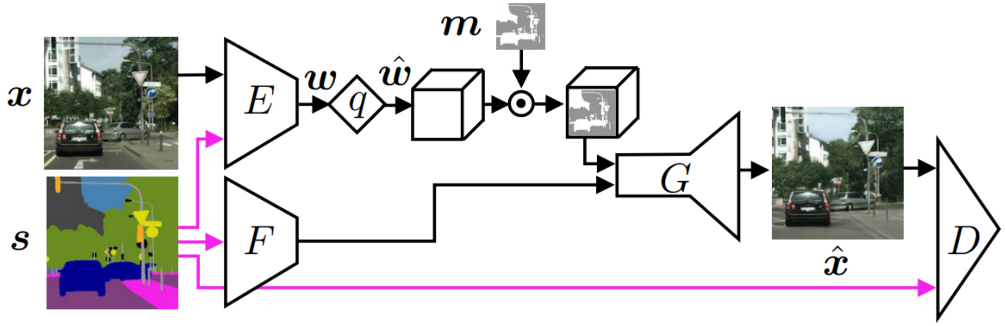




[13] Agustsson, Eirikur, et al., "Generative adversarial networks for extreme learned image compression." in ICCV. 2019.

Learned Image Compression

- Generative image compression: GAN-based methods [13]



Conditional GAN: $\mathcal{L}_{\text{cGAN}} := \max_{D} \mathbb{E}[f(D(\boldsymbol{x}, \boldsymbol{s}))] + \mathbb{E}[g(D(G(\boldsymbol{z}, \boldsymbol{s}), \boldsymbol{s}))]$

Selective generative compression (SC): binary heatmap $\boldsymbol{m}$

road (0.146bpp, -55%) | *car* (0.227bpp, -15%) | all synth. (0.035bpp, -89%)

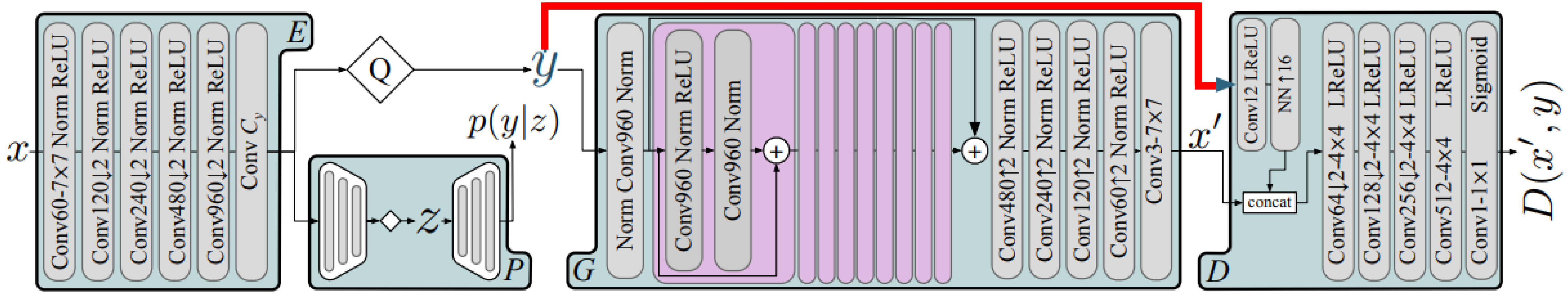
people (0.219bpp, -33%) | *building* (0.199bpp, -39%) | no synth. (0.326bpp, -0%)

[13] Agustsson, Eirikur, et al. "Generative adversarial networks for extreme learned image compression." in ICCV. 2019.

Learned Image Compression

- Generative image compression: GAN-based methods [14]

High-Fidelity Generative Image Compression



Conditional discriminator:

$$\mathcal{L}_{EGP} = \mathbb{E}_{x\sim p_X}[\lambda r(y) + d(x,x') - \beta \underline{\log(D(x',y))}],$$
$$\mathcal{L}_D = \mathbb{E}_{x\sim p_X}[-\log(1-\underline{D(x',y))}] + \mathbb{E}_{x\sim p_X}[-\underline{\log(D(x,y))}].$$

[14] Mentzer, Fabian, et al. "High-Fidelity Generative Image Compression." in NeurIPS. 2020.

Learned Image Compression

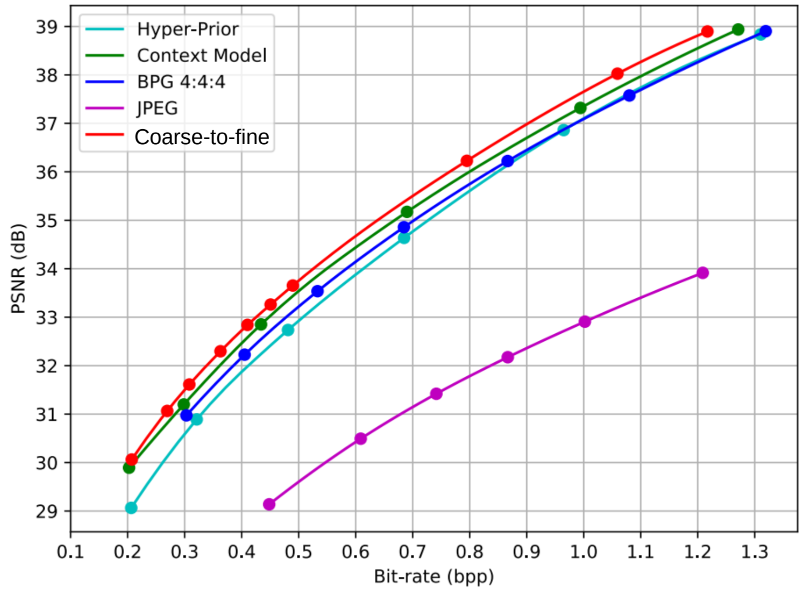
- Generative image compression: GAN-based methods [14]

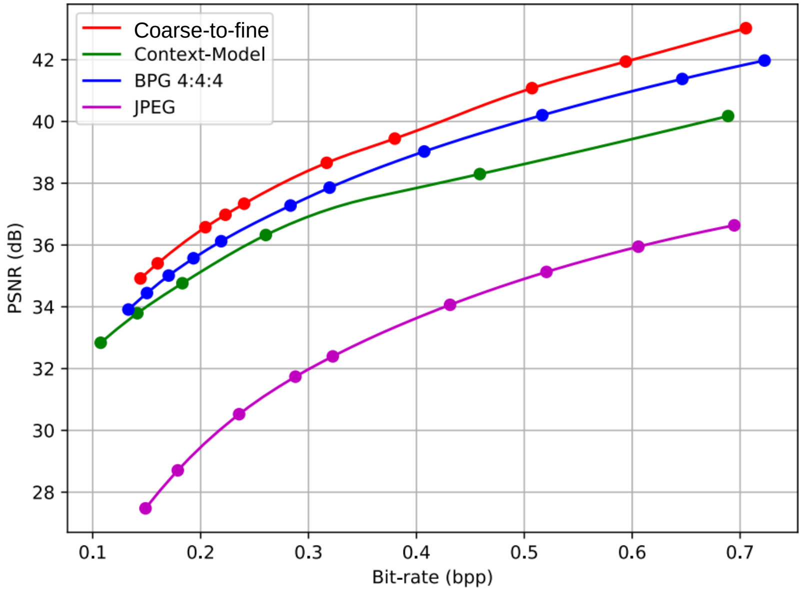
[14] Mentzer, Fabian, et al., "High-Fidelity Generative Image Compression." in NeurIPS. 2020.

Learned Image Compression

Conclusion:

- CNN-based methods
  - Factorized entropy model
  - Hyperprior entropy model
  - Autoregressive entropy model
  - Coarse-to-fine entropy model
  - Conditional auto-encoder (variable bit-rates)
  - Invertible auto-encoder (lossy and lossless by one framework)
- RNN-based methods
  - Variable bit-rate
- GAN-based methods
  - Photo-realistic compressed image with low bit-rate





The state-of-the-art learned image compression methods successfully outperform the latest traditional compression standard BPG 4:4:4

Learned Image Compression

- Will learning-based compression be standardized?
- Can learning-based method be compatible with traditional standards (e.g., JPEG)?

JPEG initiates standardisation of image compression based on AI

The 89th JPEG meeting was held online from 5 to 9 October 2020.

During this meeting multiple JPEG standardisation activities and explorations were discussed and progressed. Notably, the call for evidence on learning-based image coding was successfully completed and evidence was found that this technology promises several new functionalities while offering at the same time superior compression efficiency, beyond the state of the art.

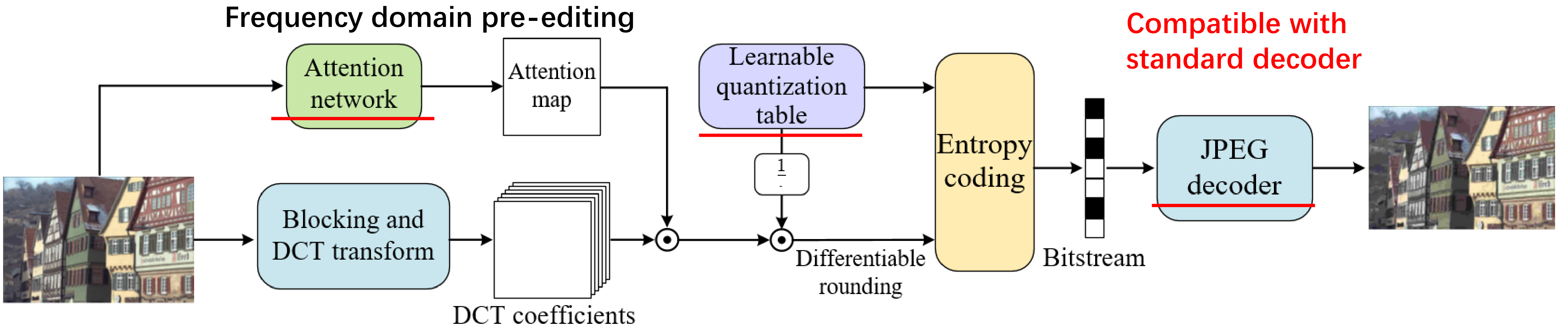
JPEG AI

At the 89th meeting the submissions to the Call for Evidence on learning-based image coding were presented and discussed. Four submissions were received in response to the Call for Evidence. The results of the subjective evaluation of the submissions to the Call for Evidence were reported and discussed in detail by experts. It was agreed that there is strong evidence that learning-based image coding solutions can outperform the already defined anchors in terms of compression efficiency, when compared to state-of- the-art conventional image coding architecture. Thus, it was decided to create a new standardisation activity for a JPEG AI on learning-based image coding system, that applies machine learning tools to achieve substantially better compression efficiency compared to current image coding systems, while offering unique features desirable for an efficient distribution and consumption of images. This type of approach should allow to obtain an efficient compressed domain representation not only for visualisation, but also for machine learning based image processing and computer vision. JPEG AI releases to the public the results of the objective and subjective evaluations as well as a first version of common test conditions for assessing the performance of leaning-based image coding systems.

Learned Image Compression

- Will learning-based compression be standardized?
- Can learning-based method be compatible with traditional standards (e.g., JPEG)?

We made an attempt: [15]

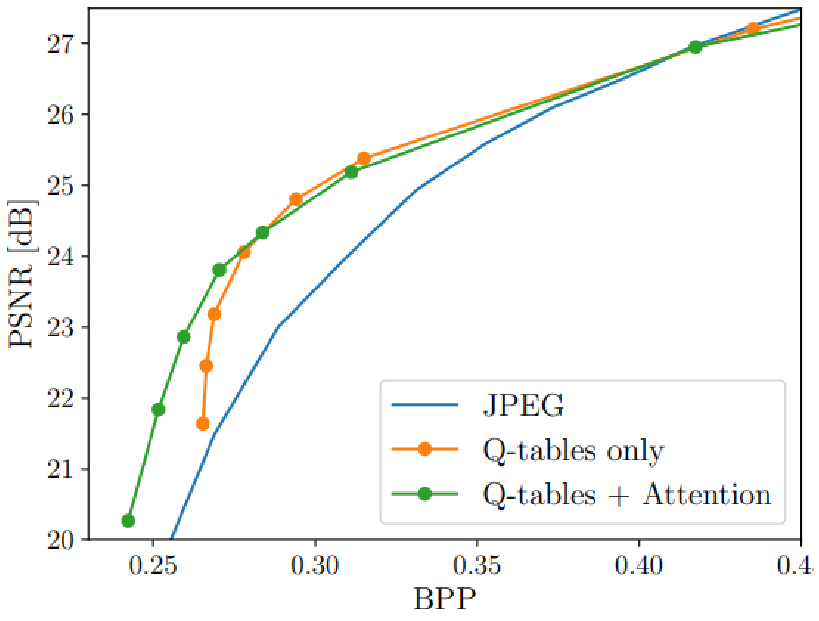


[15] Strümpler, Yannick, et al. "Learning to Improve Image Compression without Changing the Standard Decoder." in ECCVW. 2020.

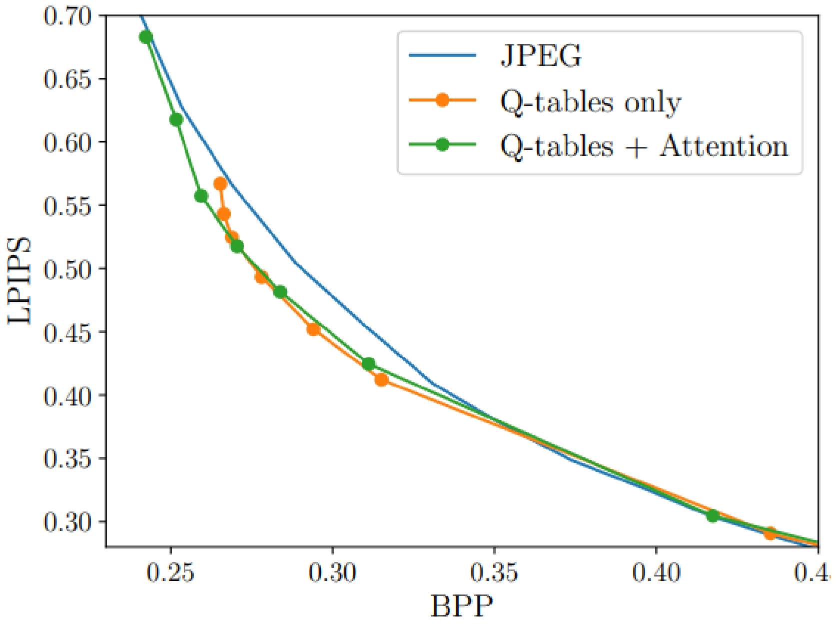
Learned Image Compression

- Will learning-based compression be standardized?
- Can learning-based method be compatible with traditional standards (e.g., JPEG)?

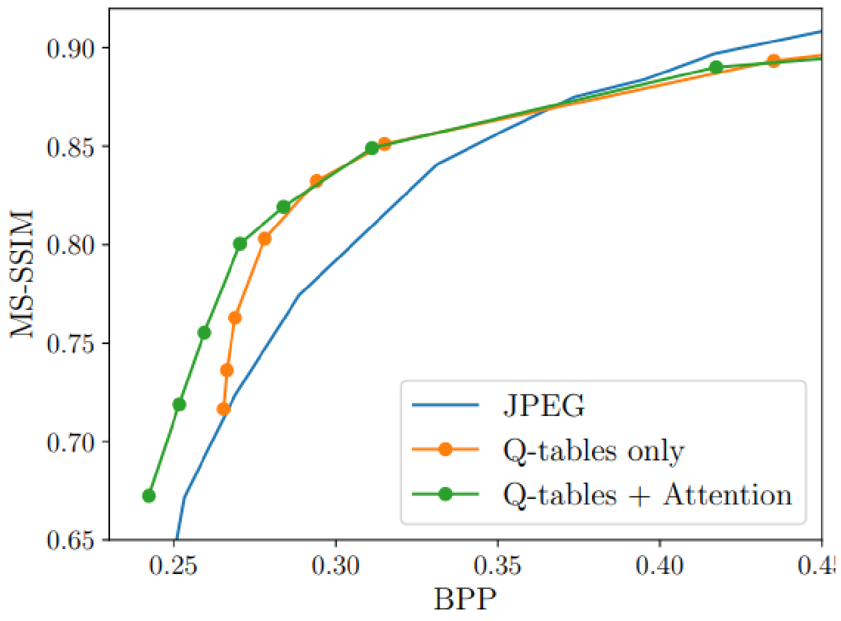
We made an attempt: [15]



PSNR on Kodak



LPIPS on Kodak



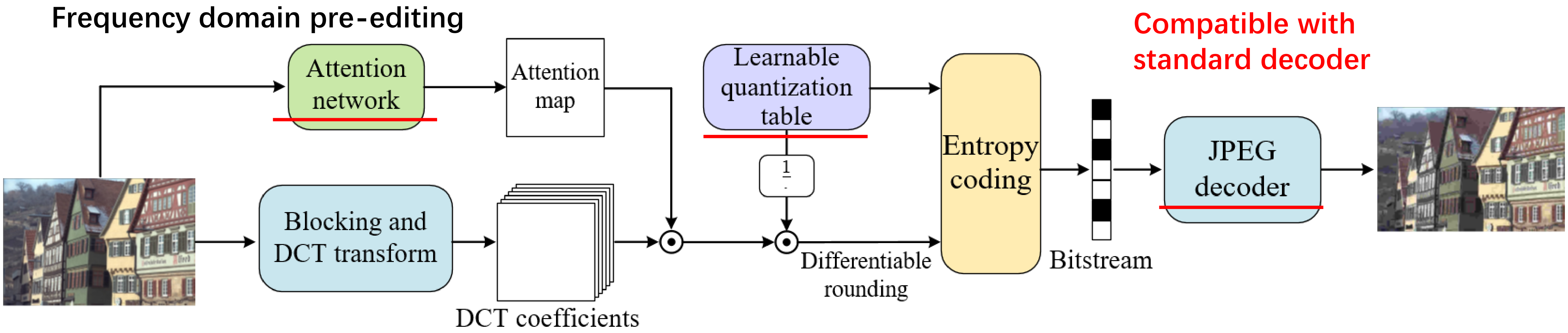
MS-SSIM on Kodak

[15] Strümpler, Yannick, et al. "Learning to Improve Image Compression without Changing the Standard Decoder." in ECCVW. 2020.

Learned Image Compression

- Will learning-based compression be standardized?
- Can learning-based method be compatible with traditional standards (e.g., JPEG)?

We made an attempt: [15]



- We achieve better rate-distortion performance without changing the standard decoder
- The compressed image can be decoded (viewed) on any common device, e.g., mobile, ipad, PC, etc.

[15] Strümpler, Yannick, et al. "Learning to Improve Image Compression without Changing the Standard Decoder." in ECCVW. 2020.

Learned Image Compression

- Open source codes:
  - Ballé et al., (factorized), Ballé et al., (hyperprior):
  https://github.com/tensorflow/compression (TensorFlow)
  - Ballé et al., (factorized), Ballé et al., (hyperprior), Minnen et al., (autoregressive):
  https://interdigitalinc.github.io/CompressAI/index.html (PyTorch)
  - Lee et al., (context-adaptive):
  https://github.com/JooyoungLeeETRI/CA_Entropy_Model
  - Mentzer et al., (autoregressive + importance map):
  https://github.com/fab-jul/imgcomp-cvpr
  - Cheng et al., (GMM entropy model):
  https://github.com/ZhengxueCheng/Learned-Image-Compression-with-GMM-and-Attention
  - Hu et al., (coarse-to-fine):
  https://github.com/huzi96/Coarse2Fine-ImaComp
  - Ma et al., (wavelet-like transformer):
  https://github.com/mahaichuan/Versatile-Image-Compression
  - Mentzer et al., (generative compression):
  https://github.com/tensorflow/compression/tree/master/models/hific

Learned Image Compression

Thanks for your attention

Q & A

Dong Xu
University of Sydney, Australia
dong.xu@sydney.edu.au

Guo Lu
Beijing Institute of Technology, China
luguo2014@sjtu.edu.cn

Ren Yang
ETH Zurich, Switzerland
ren.yang@vision.ee.ethz.ch

Radu Timofte
ETH Zurich, Switzerland
radu.timofte@vision.ee.ethz.ch