Learning for Image Compression with Deep Neural Networks

Ren Yang Dr. Radu Timofte

Computer Vision Laboratory, D-ITET, ETH Zurich

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

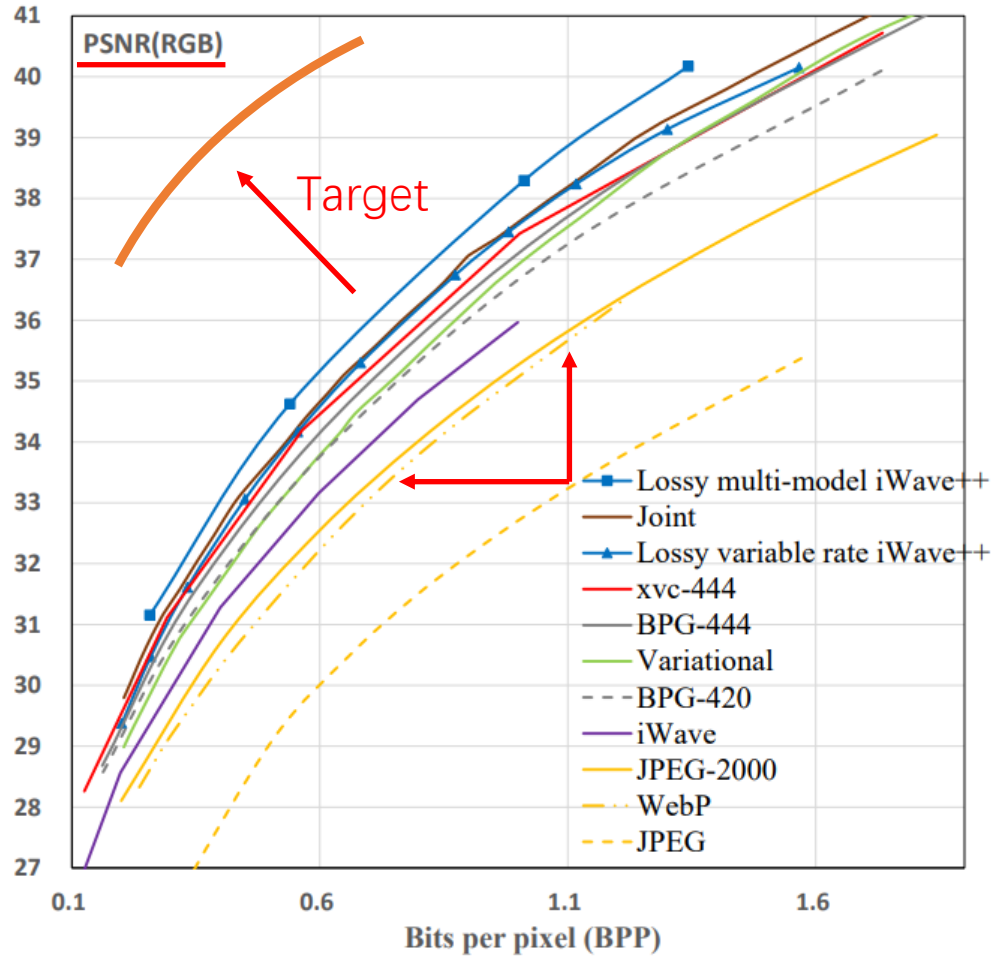
Lossy compression (.jpg): 9 MB

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)



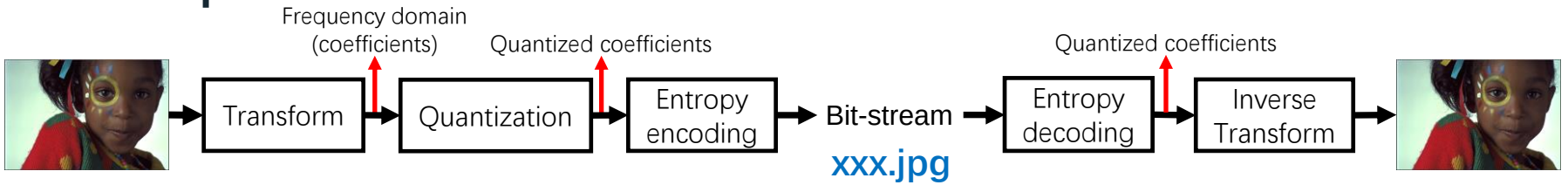
Better quality at the same bit-rate

Lower bit-rate at the same quality

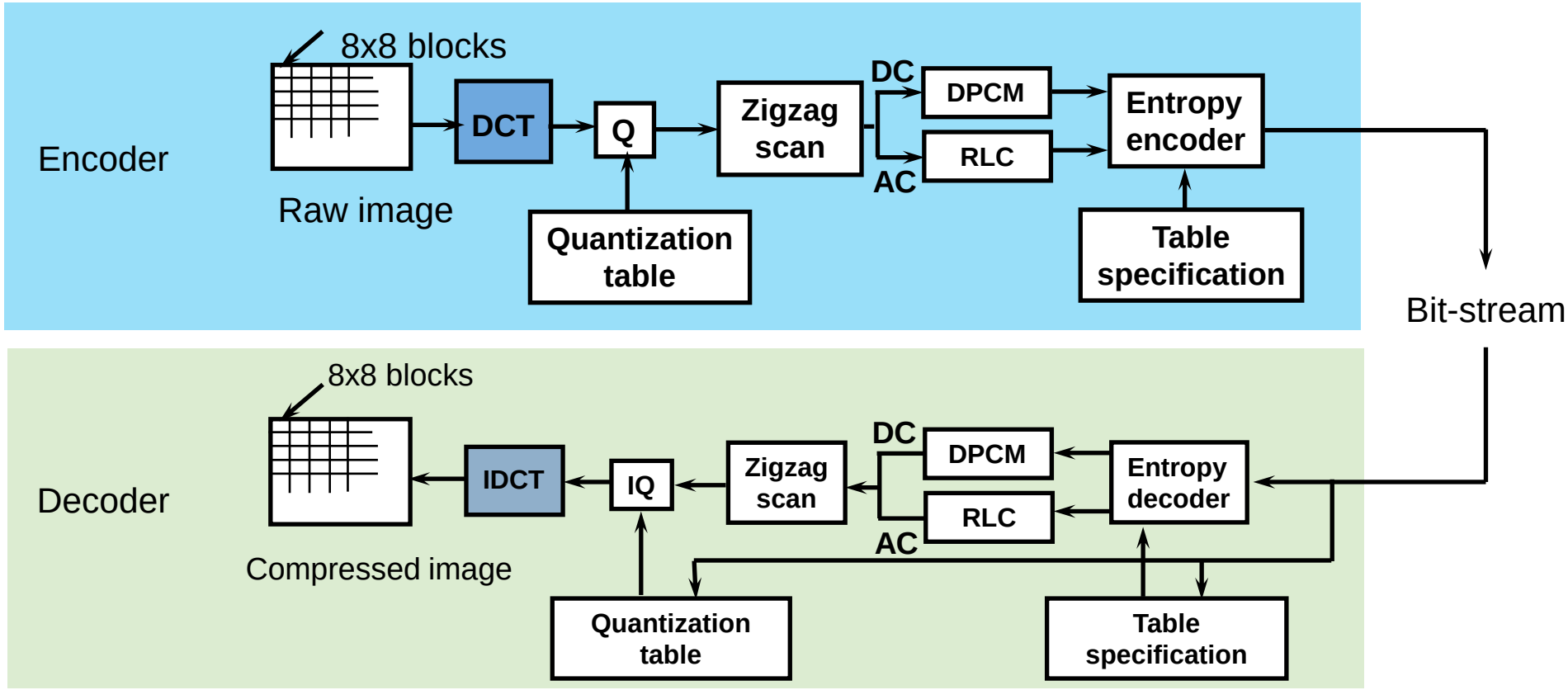
Higher bit-rate (file size)

Traditional Image Compression

- Classical Architecture:



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



Entropy coding

Entropy:

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

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

Cross entropy:

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

real estimated

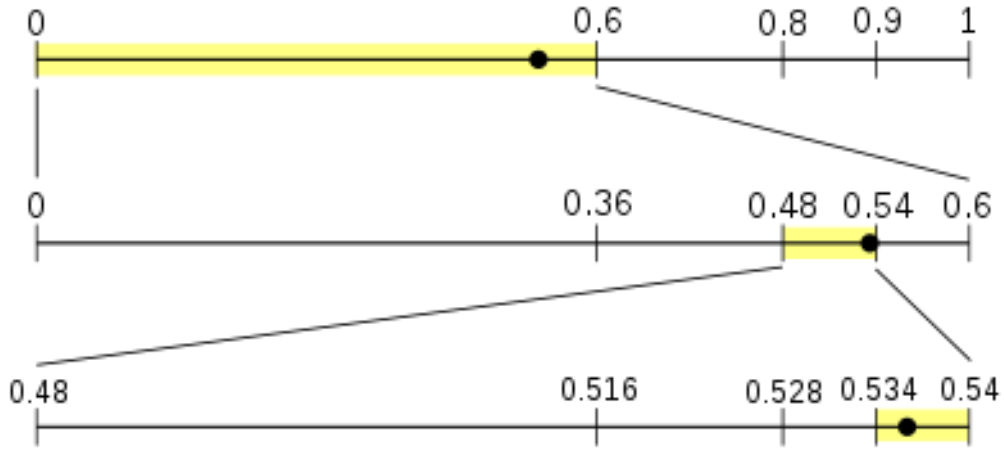
(Adaptive) arithmetic coding is theoretically able to losslessly compress data at

- 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



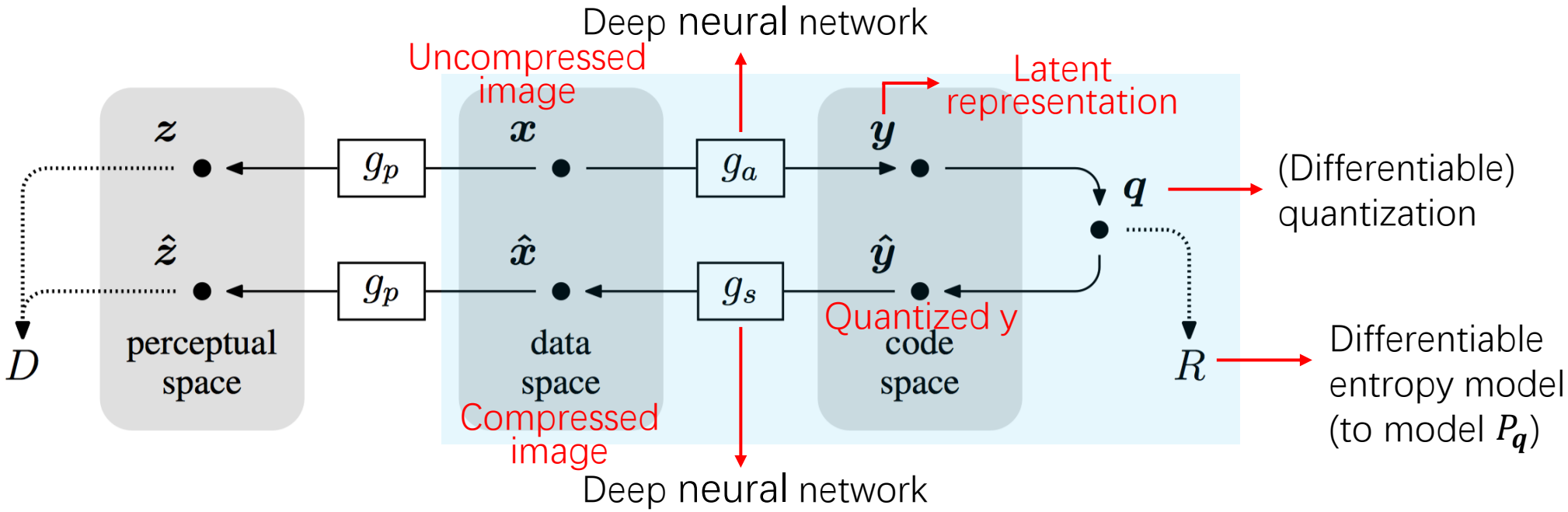
0.10001010 **8 bits**
(0.5390625 decimal)

Adaptive arithmetic coding:

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

Learned Image Compression

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

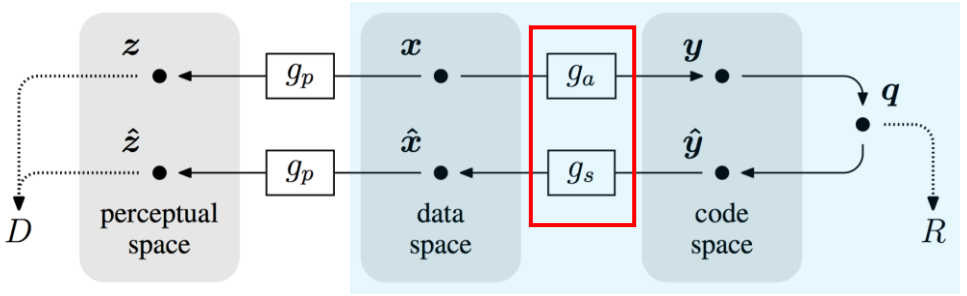


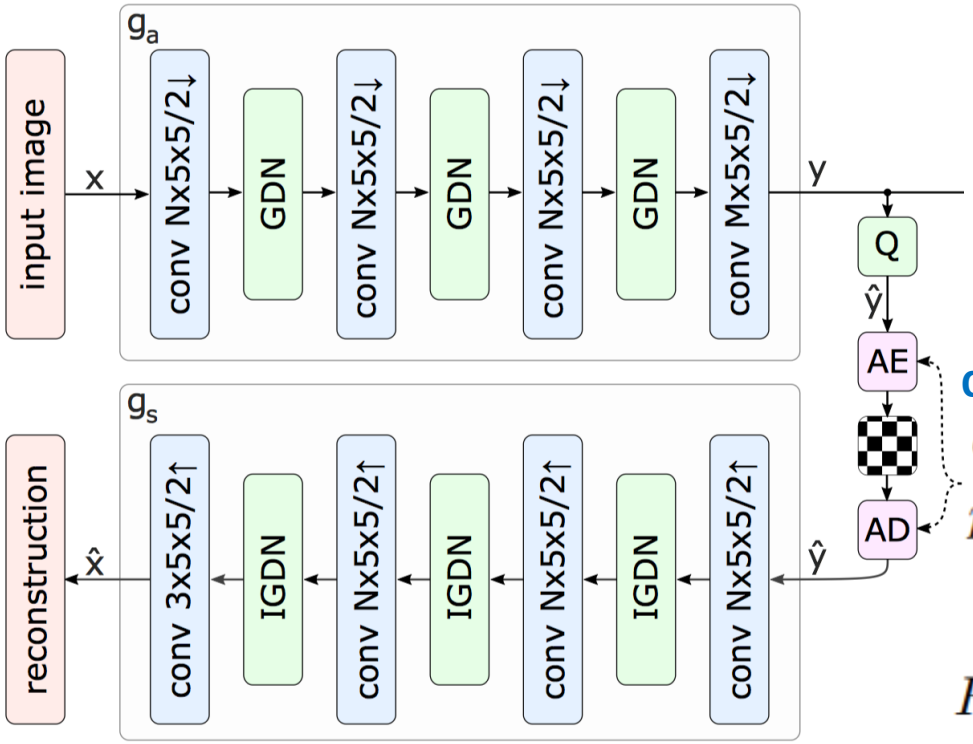
$$L[g_a, g_s, P_{\boldsymbol{q}}] = \underbrace{-\mathbb{E}\left[\log_2 P_{\boldsymbol{q}}\right]}_{R} + \lambda\, \mathbb{E}\left[d(\boldsymbol{x}, \hat{\boldsymbol{x}})\right]$$

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

Learned Image Compression

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





$$\tilde{\boldsymbol{y}} = \boldsymbol{y} + \Delta\boldsymbol{y} \quad \Delta\boldsymbol{y} \sim \mathcal{U}(0,1)$$

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 $\hat{\boldsymbol{y}}$

$$f_k(\underline{\boldsymbol{x}}) = g_k\left(\boldsymbol{H}^{(k)}\underline{\boldsymbol{x}} + \boldsymbol{b}^{(k)}\right) \qquad 1 \le k < K$$

$$f_K(\underline{\boldsymbol{x}}) = \mathrm{sigmoid}\left(\boldsymbol{H}^{(K)}\underline{\boldsymbol{x}} + \boldsymbol{b}^{(K)}\right)$$

$$g_k(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{a}^{(k)} \odot \tanh(\boldsymbol{x})$$

$$g'_k(\boldsymbol{x}) = 1 + \boldsymbol{a}^{(k)} \odot \tanh'(\boldsymbol{x})$$

$$\boldsymbol{H}^{(k)} = \mathrm{softplus}\left(\hat{\boldsymbol{H}}^{(k)}\right)$$

$$\boldsymbol{a}^{(k)} = \tanh\left(\hat{\boldsymbol{a}}^{(k)}\right)$$

$$R = \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}\left[-\log_2 p_{\hat{\boldsymbol{y}}}\left(Q(g_a(\boldsymbol{x};\boldsymbol{\phi}_g))\right)\right]$$

estimated bit-rate

$$L(\boldsymbol{\theta},\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{x},\Delta\boldsymbol{y}}\left[-\sum_i \log_2 p_{\tilde{y}_i}(g_a(\boldsymbol{x};\boldsymbol{\phi}) + \Delta\boldsymbol{y};\boldsymbol{\psi}^{(i)}) + \lambda d\left(g_p(g_s(g_a(\boldsymbol{x};\boldsymbol{\phi}) + \Delta\boldsymbol{y};\boldsymbol{\theta})), g_p(\boldsymbol{x})\right)\right]$$

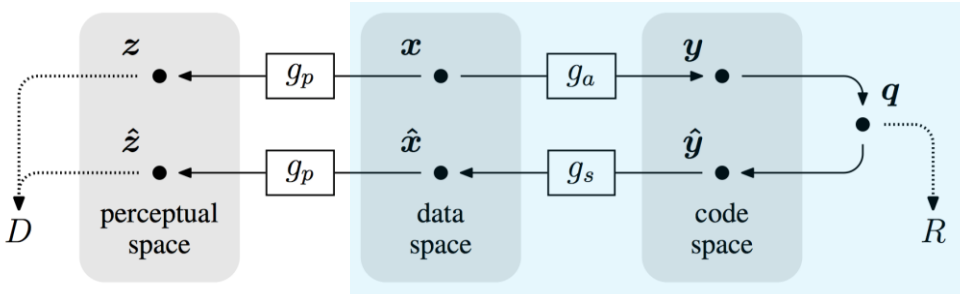
bit-rate | trade-off | distortion

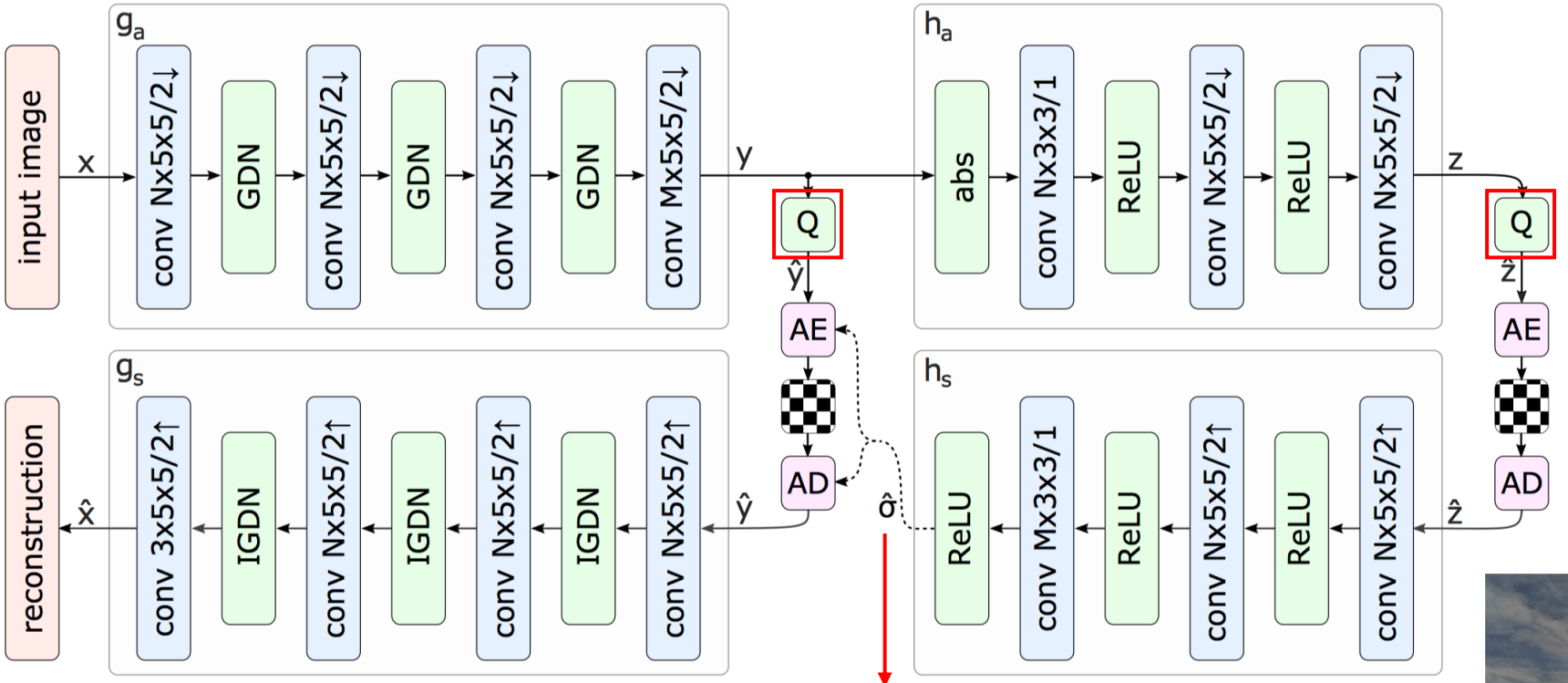
Optimized in an end-to-end manner

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

Learned Image Compression

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





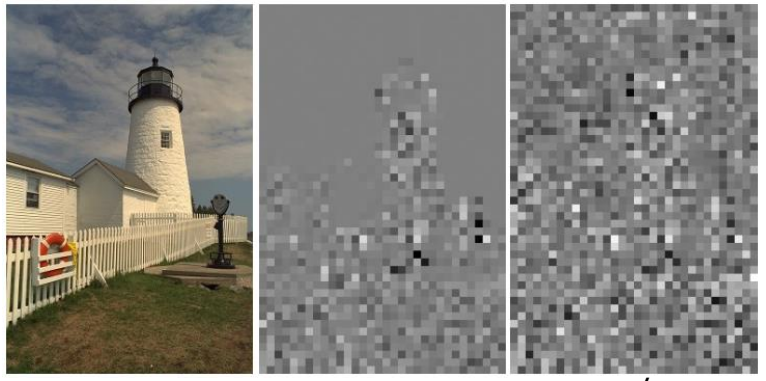
differentiable quantization

differentiable entropy model

less dependency (less redundancy)

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

discretized Gaussian distribution

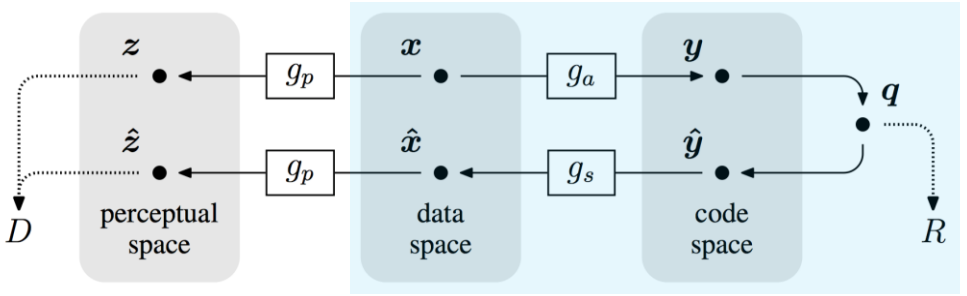


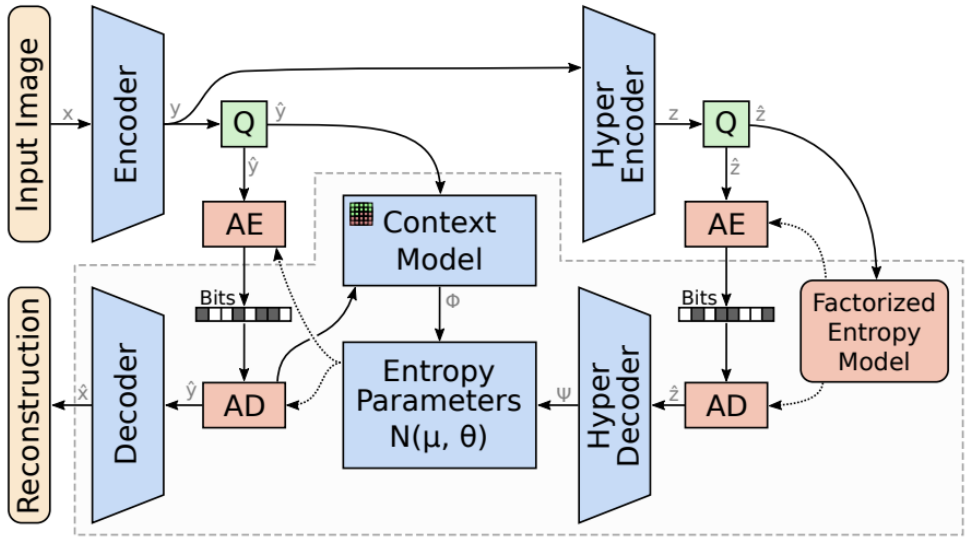
$\boldsymbol{y}$ (factorized) $\boldsymbol{y/\sigma}$ (hyperprior)

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

Learned Image Compression

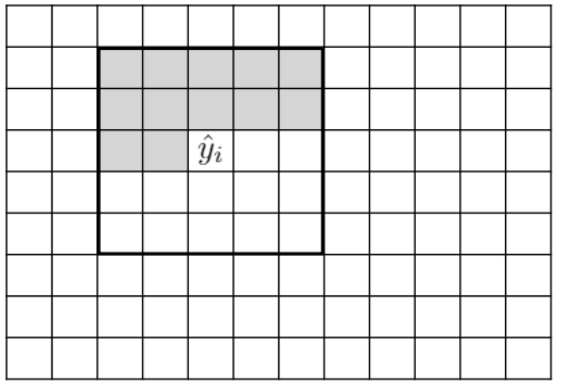
- CNN transformer + **autoregressive** entropy model [3]

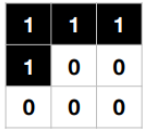




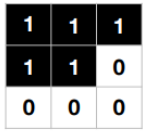
Component	Symbol
Input Image	$\boldsymbol{x}$
Encoder	$f(\boldsymbol{x};\boldsymbol{\theta}_e)$
Latents	$\boldsymbol{y}$
Latents (quantized)	$\hat{\boldsymbol{y}}$
Decoder	$g(\hat{\boldsymbol{y}};\boldsymbol{\theta}_d)$
Hyper Encoder	$f_h(\boldsymbol{y};\boldsymbol{\theta}_{he})$
Hyper-latents	$\boldsymbol{z}$
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]





first layer



other layers

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

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

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

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 $\boldsymbol{\phi}_i = g_{cm}(\hat{\boldsymbol{y}}_{<i}; \boldsymbol{\theta}_{cm})$

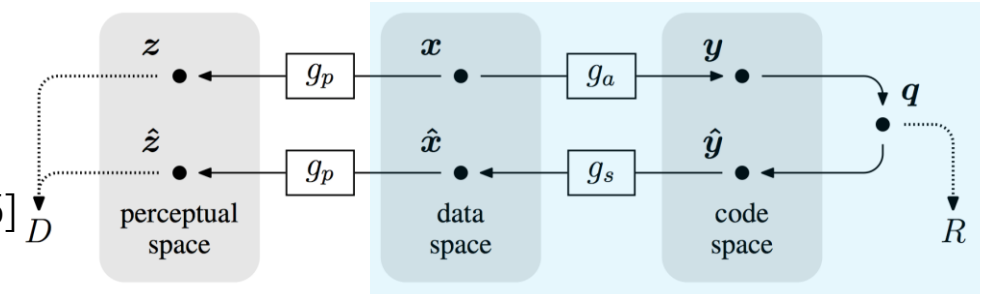
Algorithm 1 Constructing 3D Masks

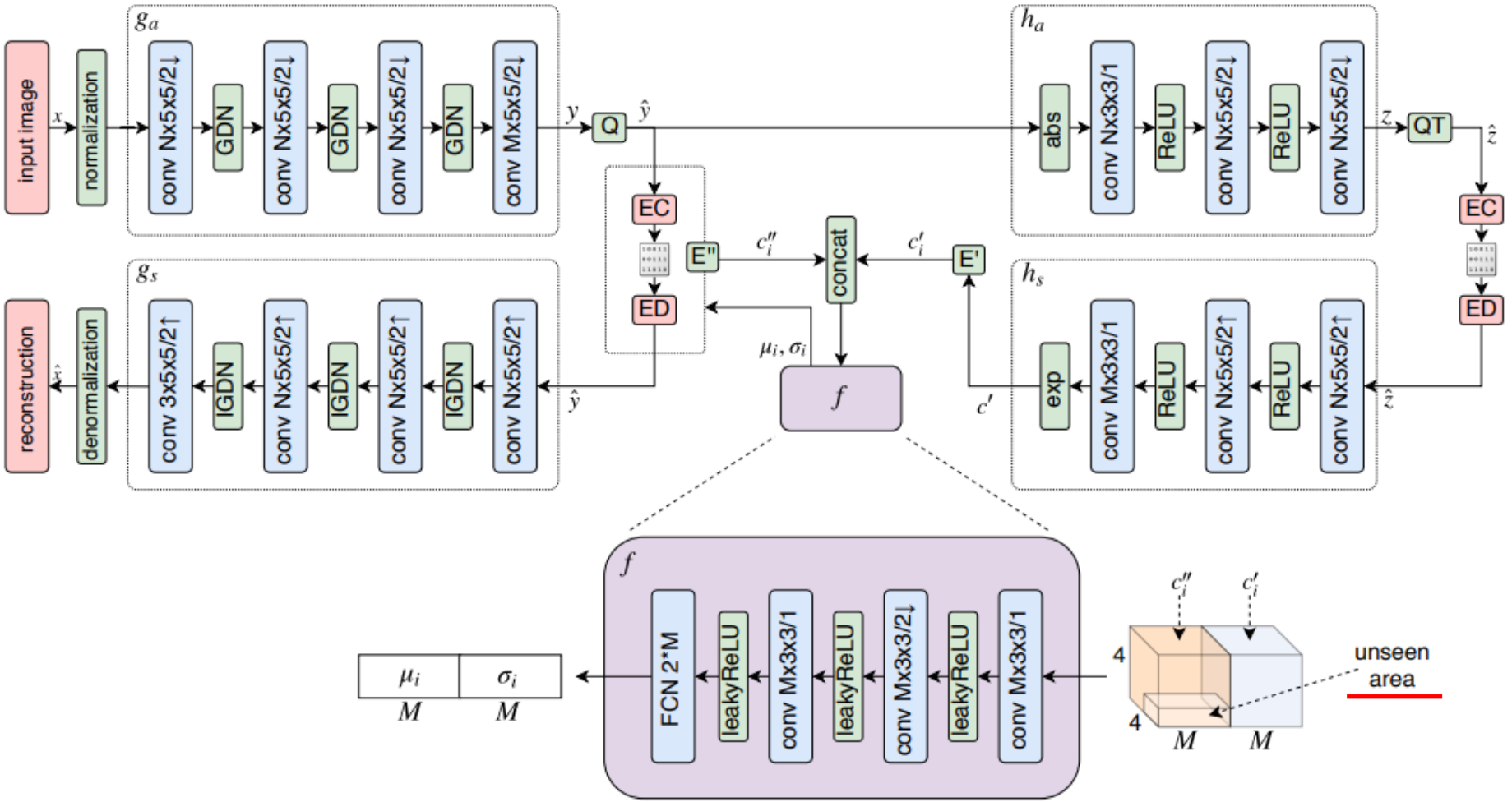
```
central_idx ← ⌈(f_W · f_H · f_D)/2⌉
current_idx ← 1
mask ← f_W × f_H × f_D-dimensional matrix of zeros
for d ∈ {1, ..., f_D} do
    for h ∈ {1, ..., f_H} do
        for w ∈ {1, ..., f_W} do
            if current_idx < central_idx then
                mask(w, h, d) = 1
            else
                mask(w, h, d) = 0
            current_idx ← current_idx + 1
```


[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

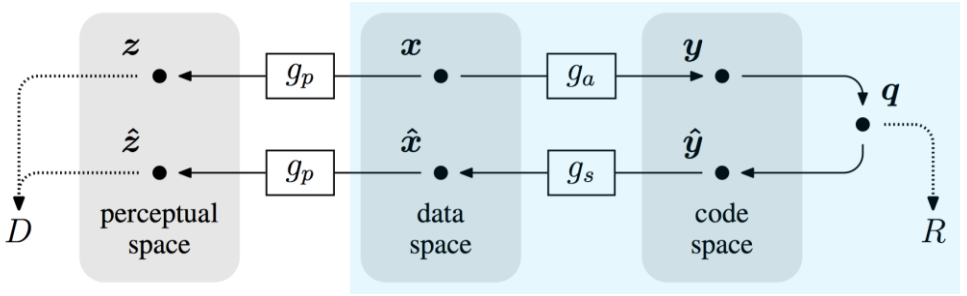
- CNN transformer + **autoregressive** entropy model [5]





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

Learned Image Compression



- Another differentiable quantization method [4]

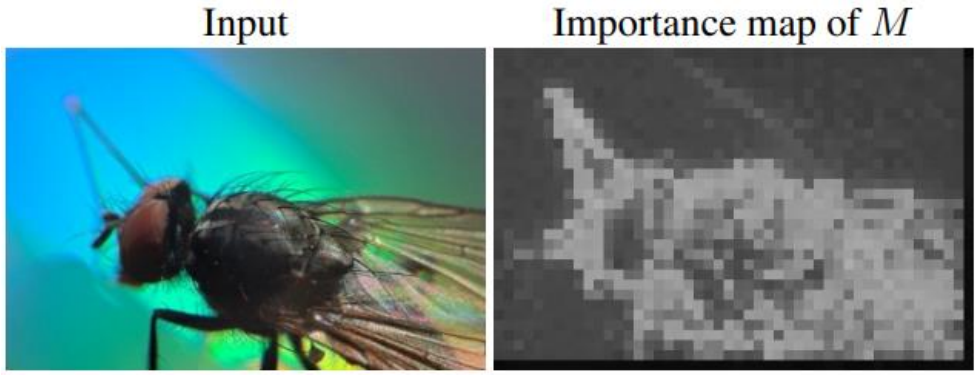
given centers $\mathcal{C} = \{c_1, \cdots, c_L\}$

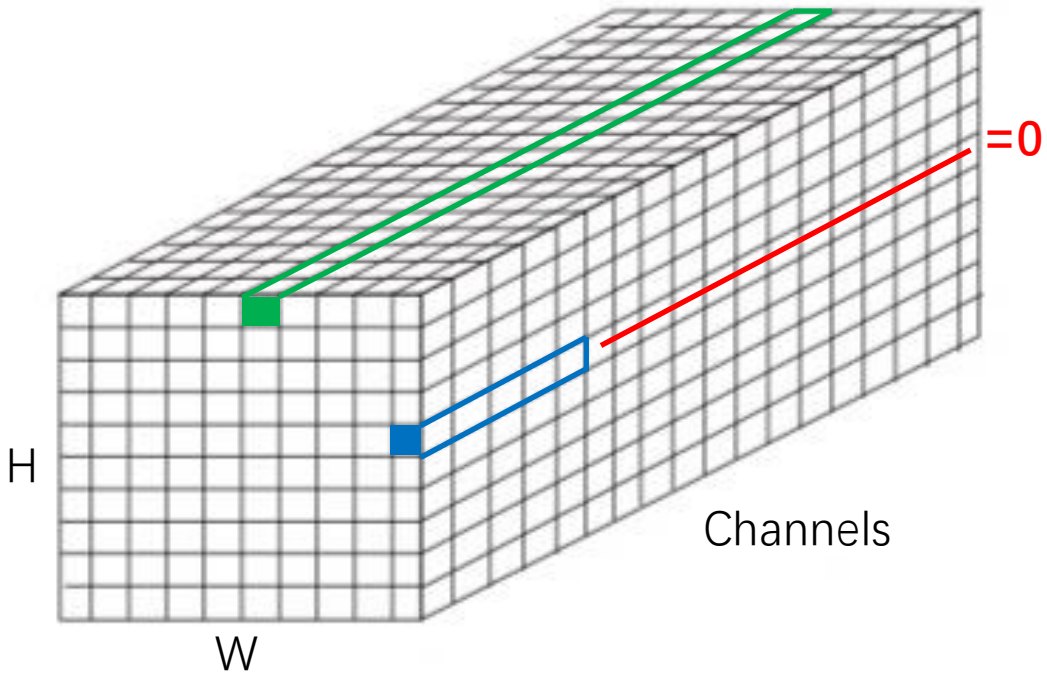
$$\hat{z}_i = Q(z_i) := \arg\min_j \|z_i - c_j\|$$ **Inference**

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

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

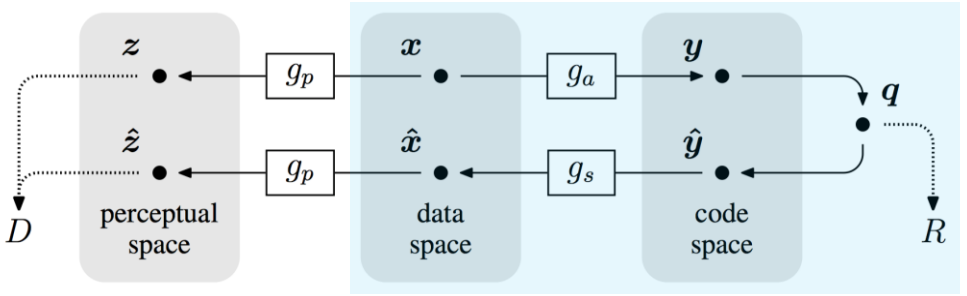
- Importance map [4]





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

Learned Image Compression



- Another differentiable quantization method [4]

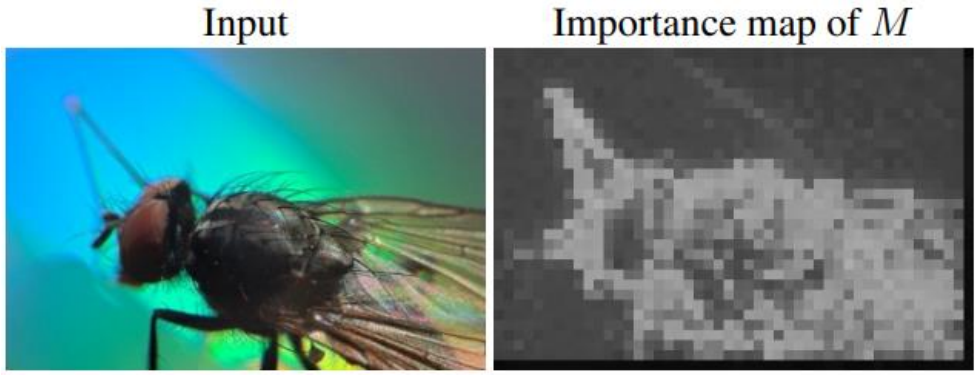
given centers $\mathcal{C} = \{c_1, \cdots, c_L\}$

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

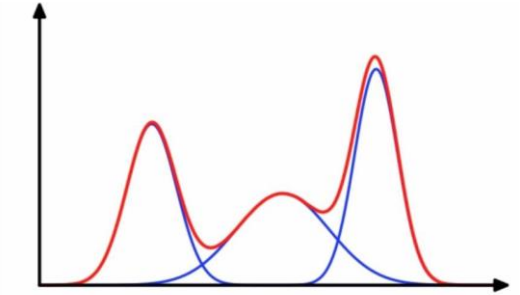
$$\bar{z}_i = \text{tf.stopgradient}(\hat{z}_i - \tilde{z}_i) + \tilde{z}_i$$

- Importance map [4]



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

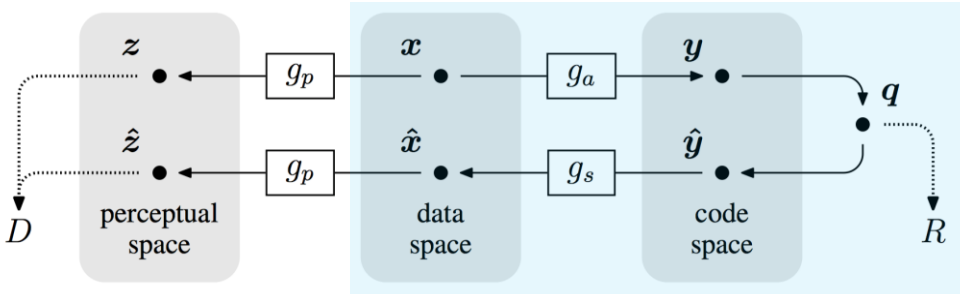
$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) \sim \sum_{k=1}^{K} \boldsymbol{w}^{(k)} \mathcal{N}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{2(k)})$$

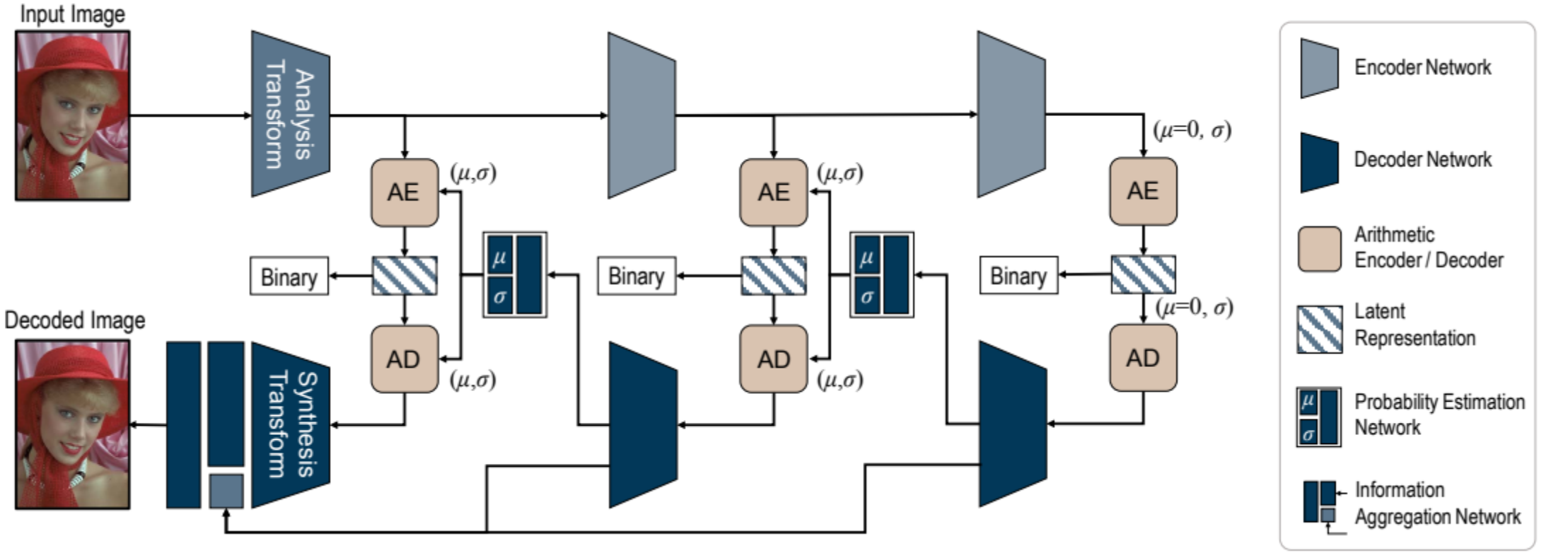


[4] Mentzer, Fabian, et al. “Conditional Probability Models for Deep Image Compression”, in CVPR, 2018.
[6] Cheng et al. ”Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules”, in CVPR. 2020.

Learned Image Compression

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

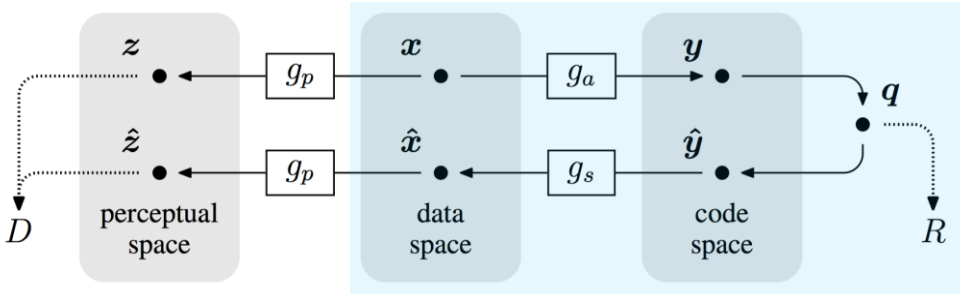


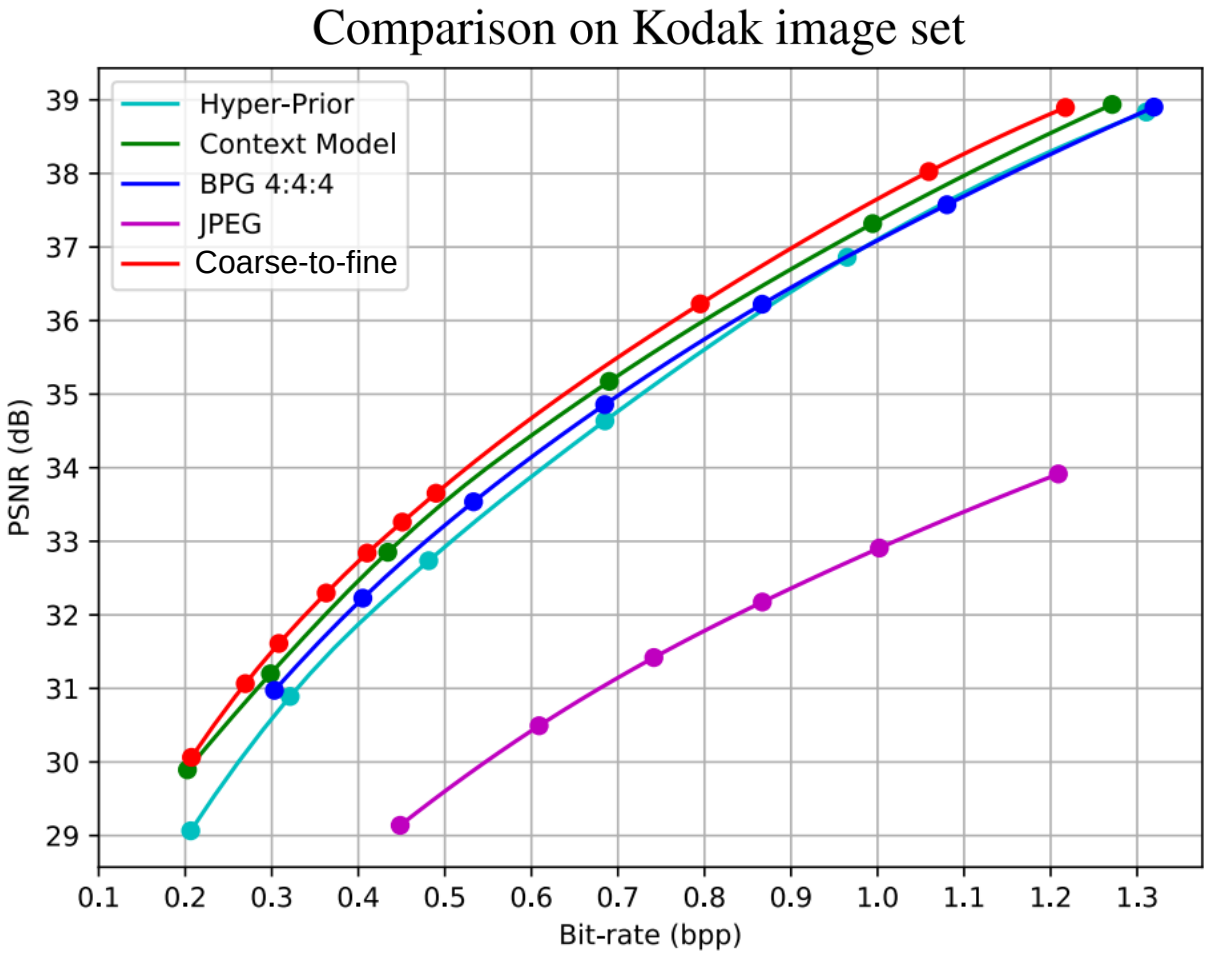


[7] Hu, Yueyu, et al. "Coarse-to-Fine Hyper-Prior Modeling for Learned Image Compression." in AAAI. 2020.

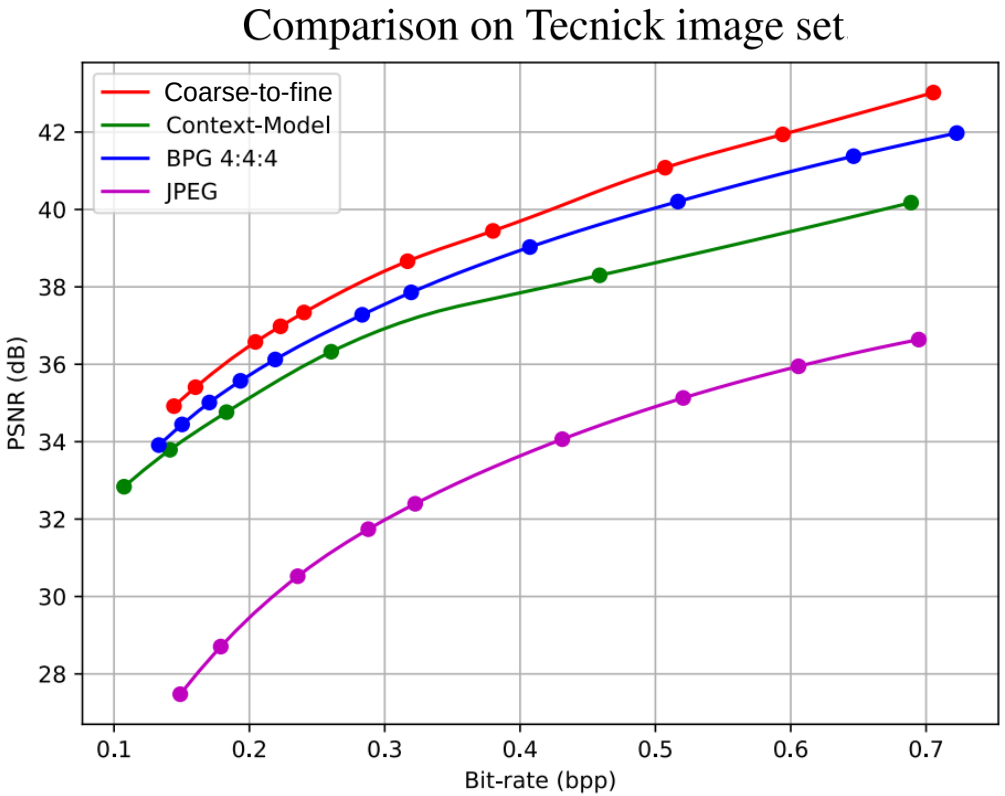
Learned Image Compression

- Performance





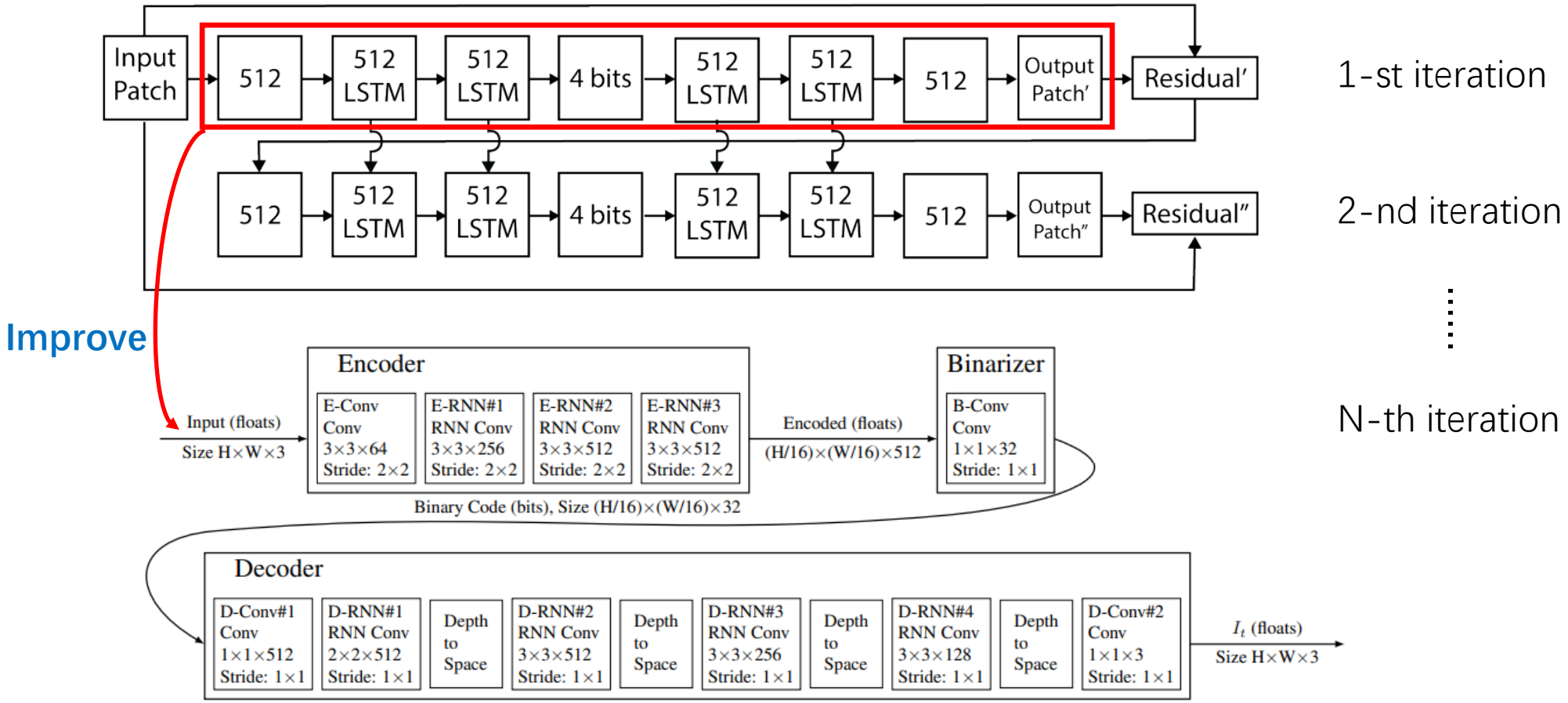
The context (autoregressive) and coarse-to-fine models outperform BPG 4:4:4 (latest traditional standard)



The rank may vary on different datasets

Learned Image Compression

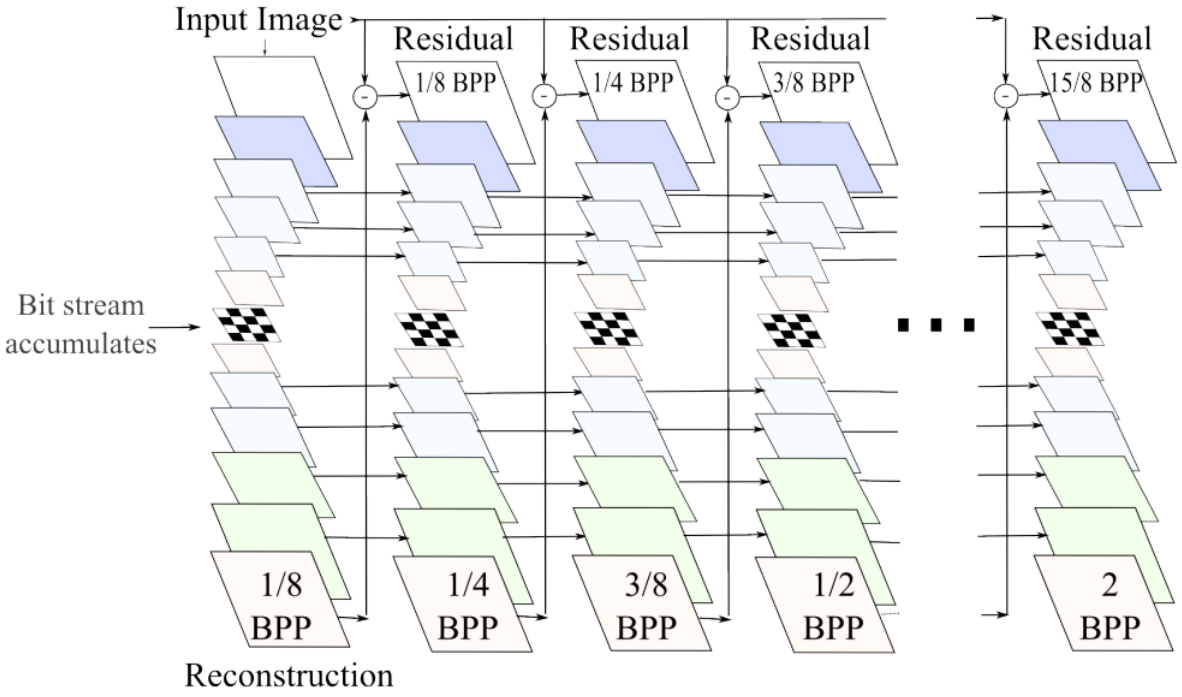
- Variable rate image compression: RNN-based methods [8, 9]



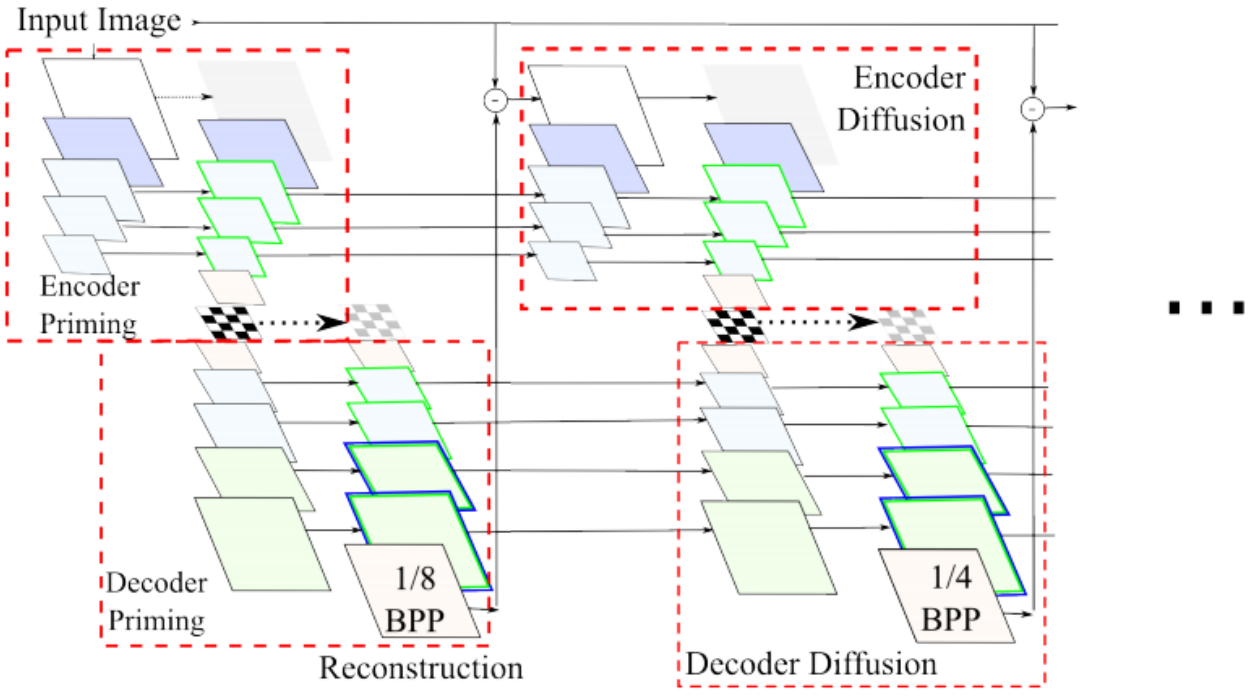
[8] Toderici, George, et al. "Variable Rate Image Compression with Recurrent Neural Networks." in ICLR. 2016.
[9] Toderici, George, et al. "Full Resolution Image Compression with Recurrent Neural Networks." in CVPR, 2017.

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.

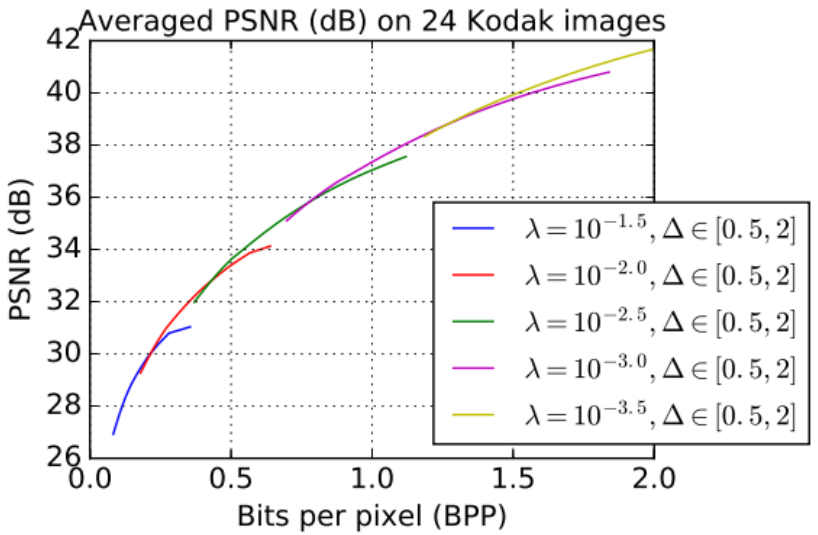
Learned Image Compression

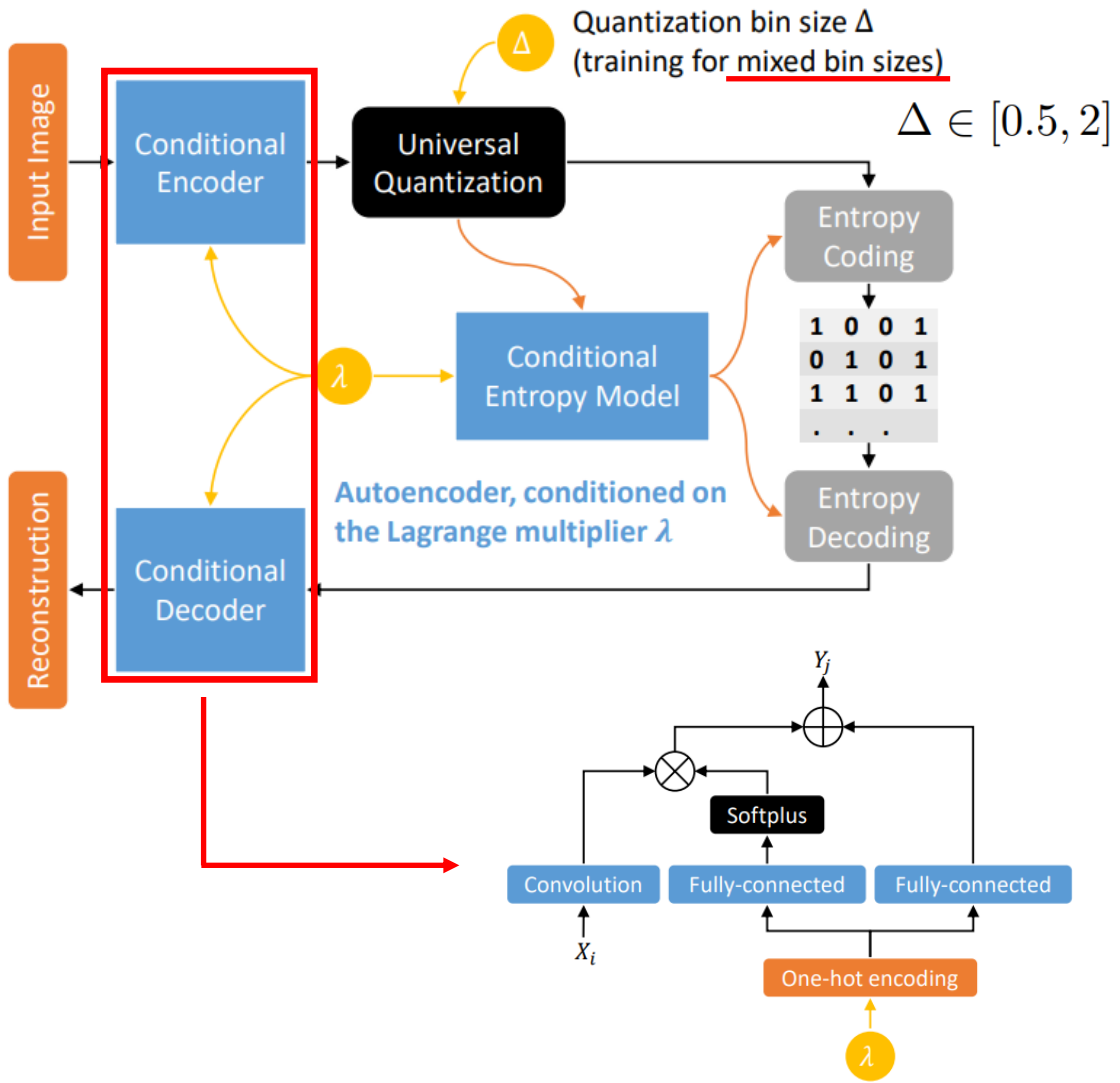
- Variable rate image compression: Conditional autoencoder [11]

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

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

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

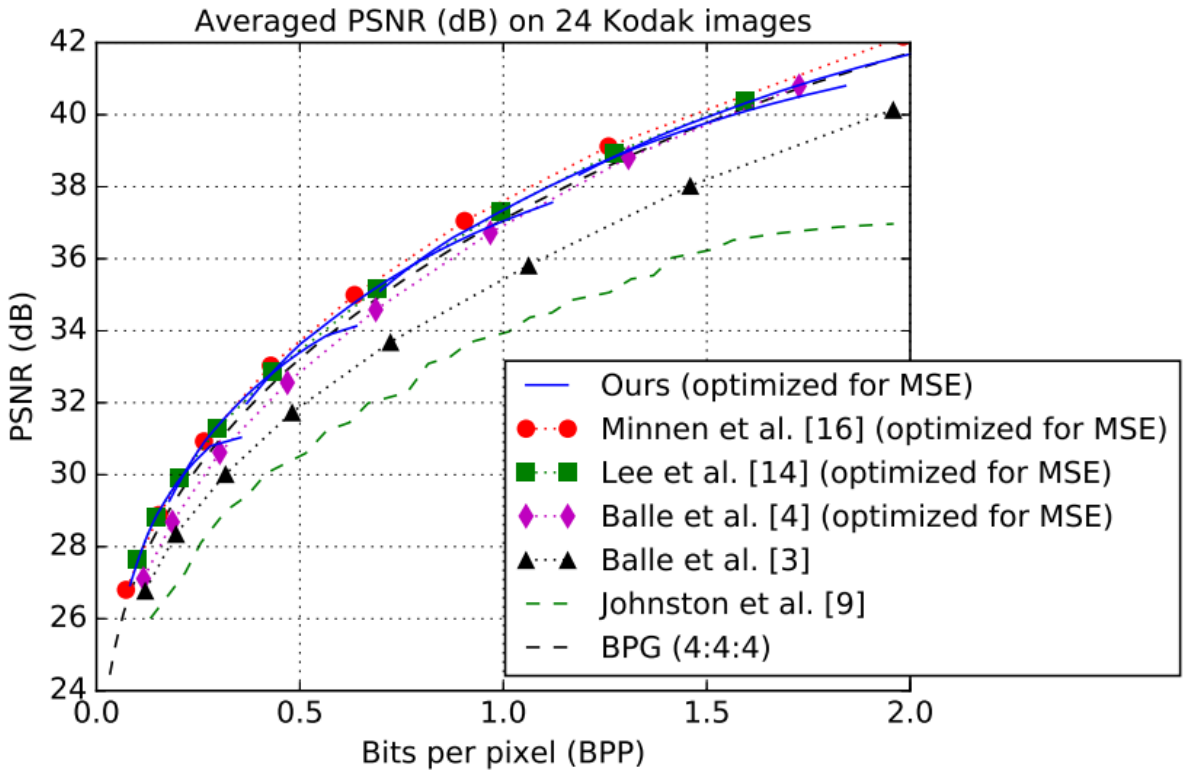


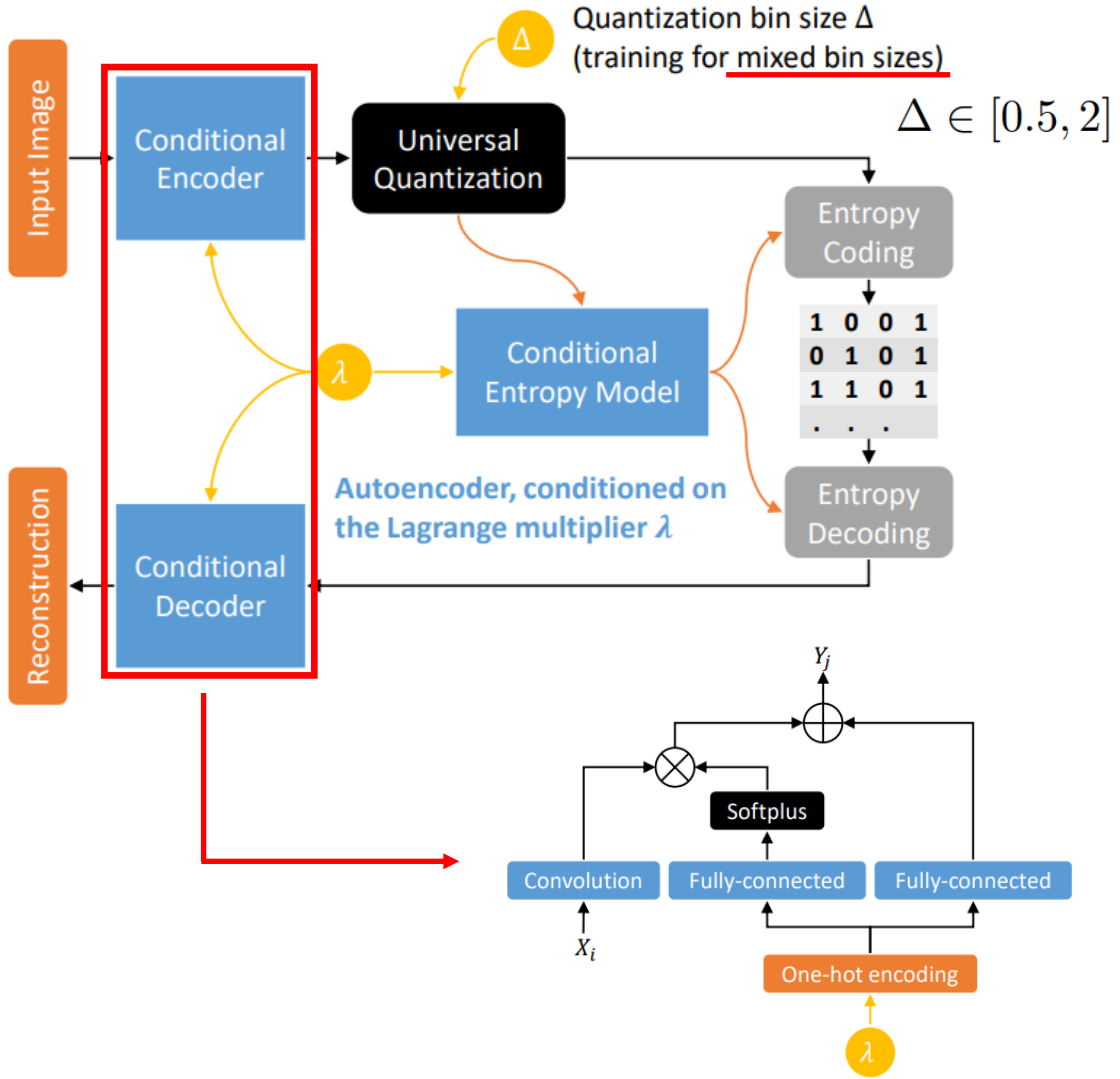


[11] Choi, Yoojin, et al. "Variable Rate Deep Image Compression With a Conditional Autoencoder." in ICCV. 2019.

Learned Image Compression

- Variable rate image compression: Conditional autoencoder [11]

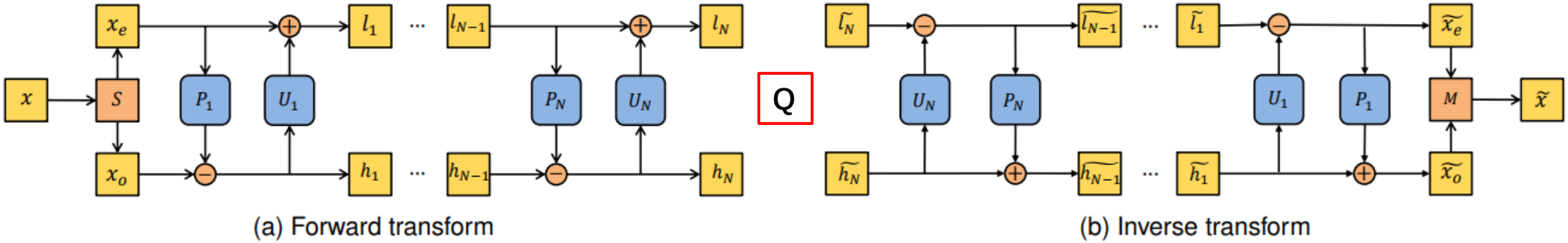




[11] Choi, Yoojin, et al. "Variable Rate Deep Image Compression With a Conditional Autoencoder." in ICCV. 2019.

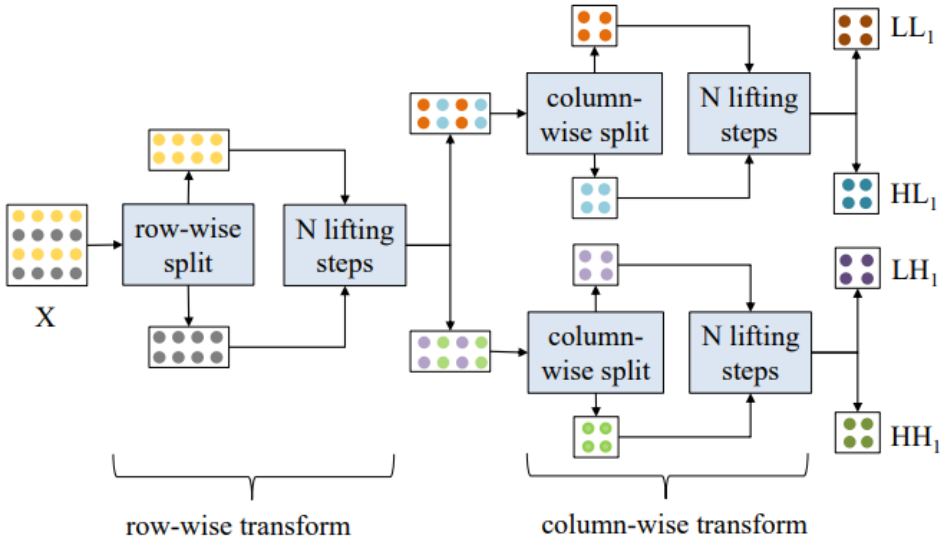
Learned Image Compression

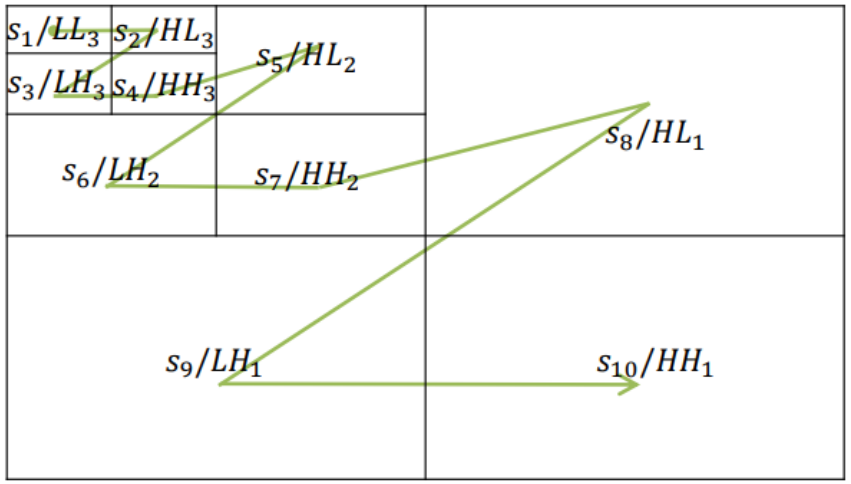
- Variable rate image compression: Wavelet-like transformer [12]



(a) Forward transform (b) Inverse transform

Invertible: achieving lossy and lossless compression by the same framework

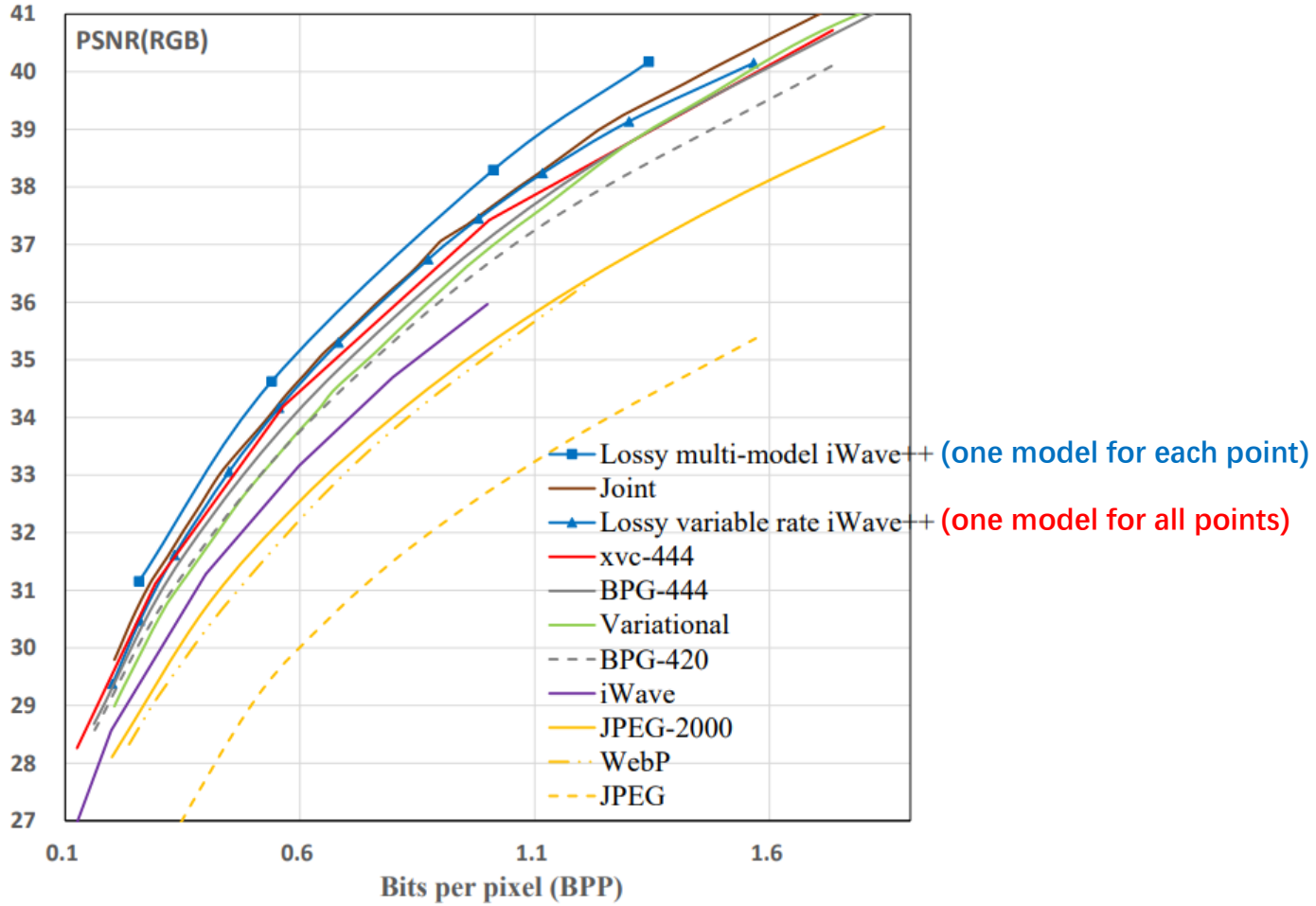


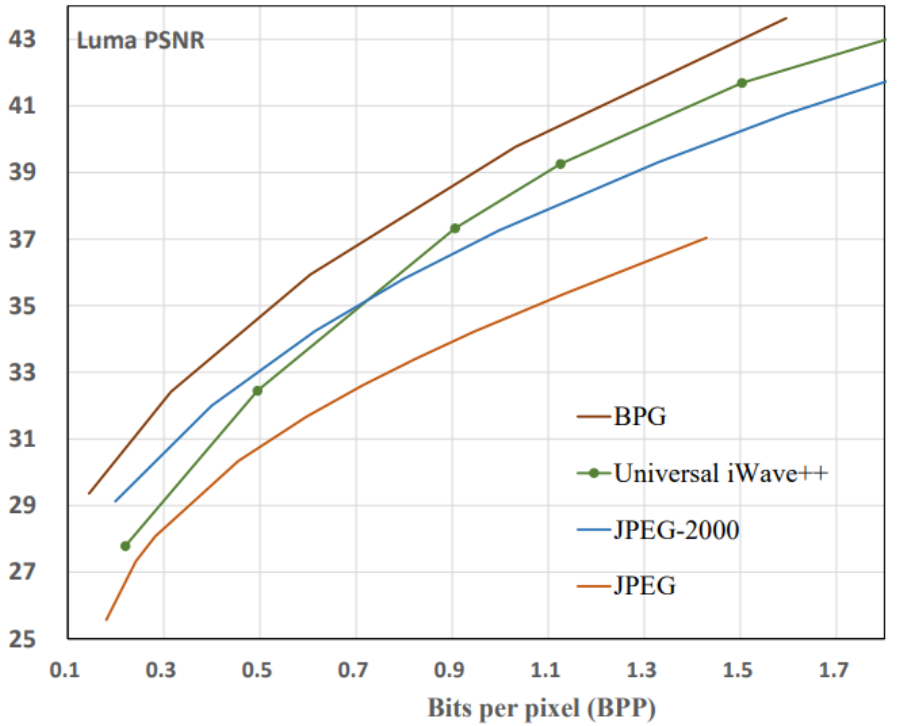


[12] Ma, Haichuan, et al. “End-to-End Optimized Versatile Image Compression With Wavelet-Like Transform.” in IEEE T-PAMI. 2020.

Learned Image Compression

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



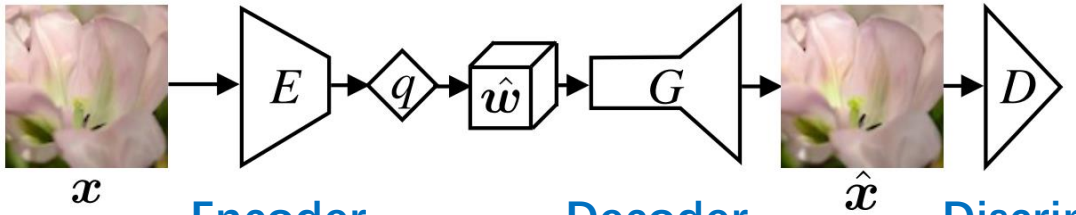


one model for both lossy and lossless compression

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

Learned Image Compression

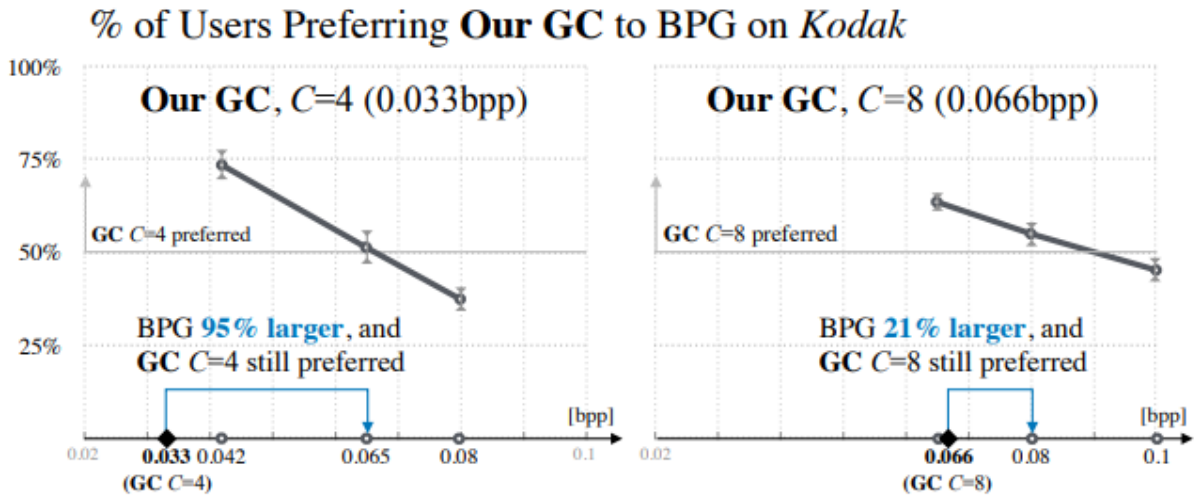
- Generative image compression: GAN-based methods [13]

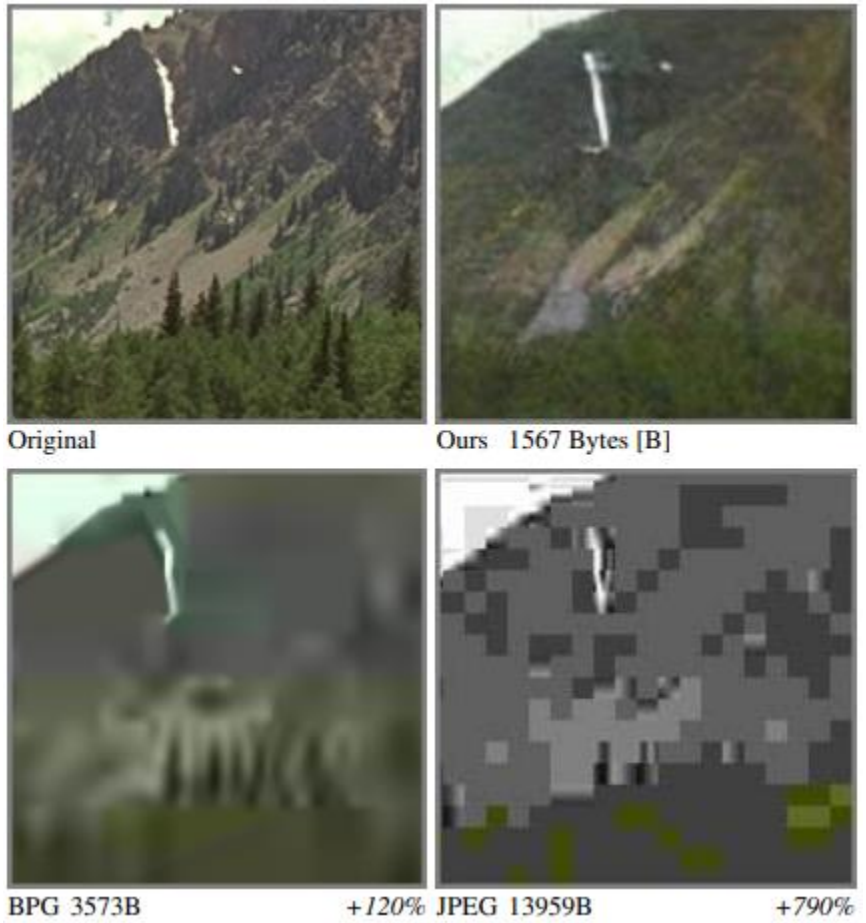


Encoder | Decoder (Generator) | Discriminator

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

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

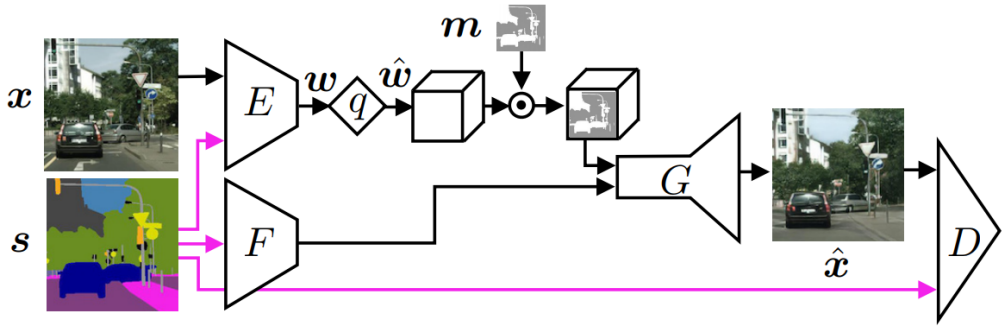




[13] Agustsson, Eirikur, et al., "Generative adversarial networks for extreme learned image compression." in ICCV. 2019.

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 $\boldsymbol{m}$

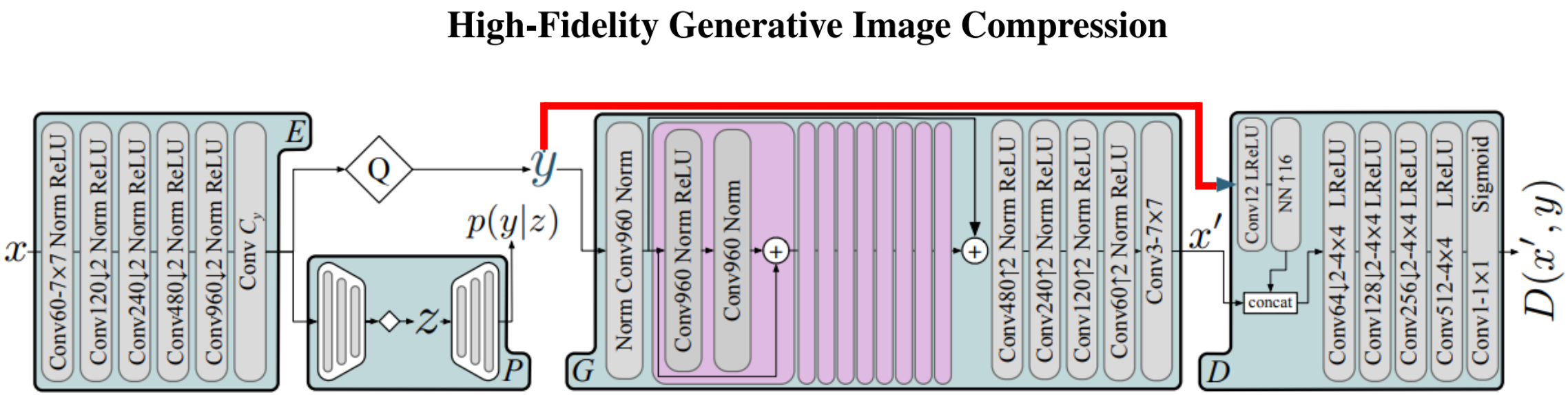
road (0.146bpp, -55%) | *car* (0.227bpp, -15%) | all synth. (0.035bpp, -89%)

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]



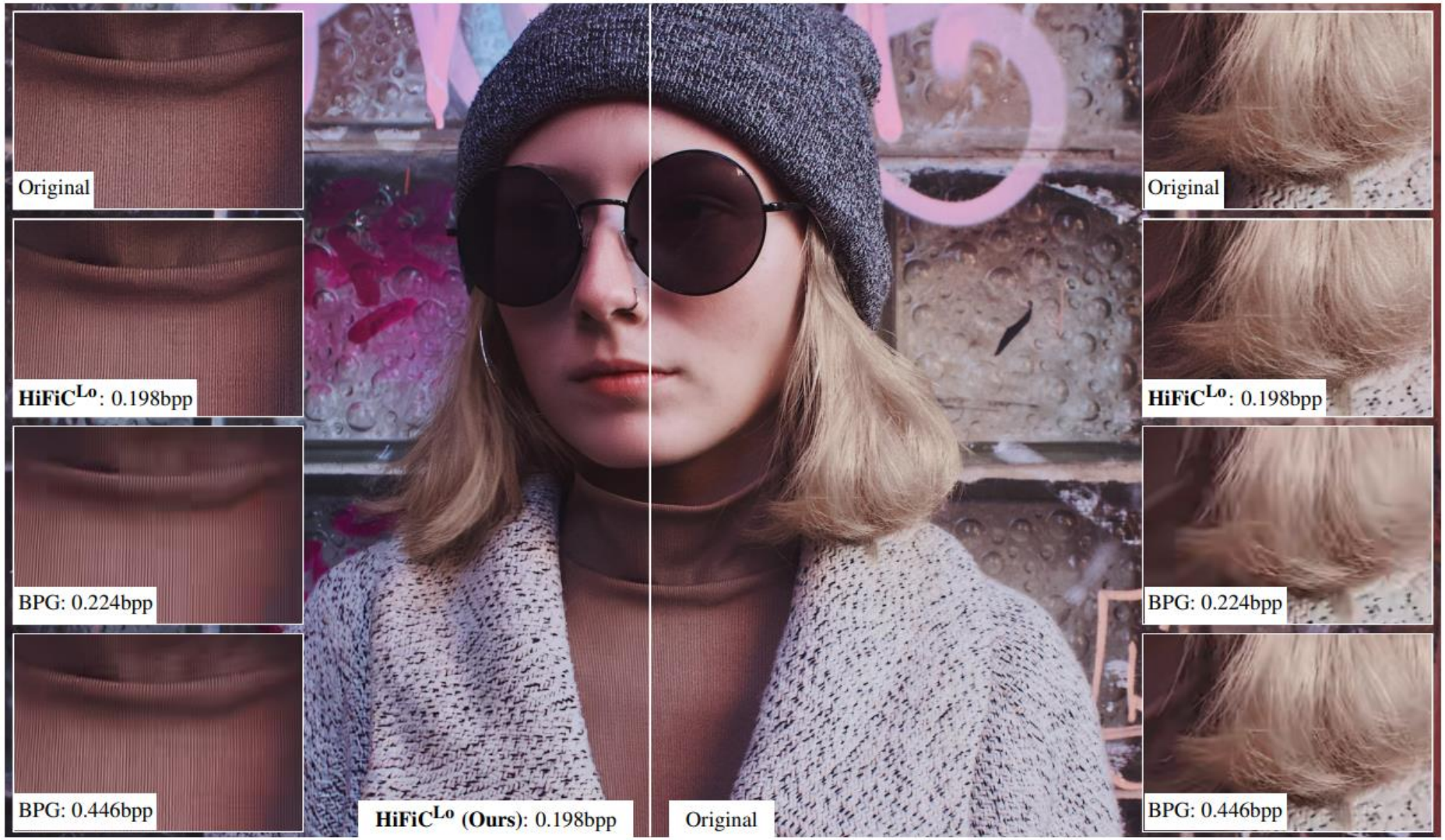
Conditional discriminator:

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

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

Learned Image Compression

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

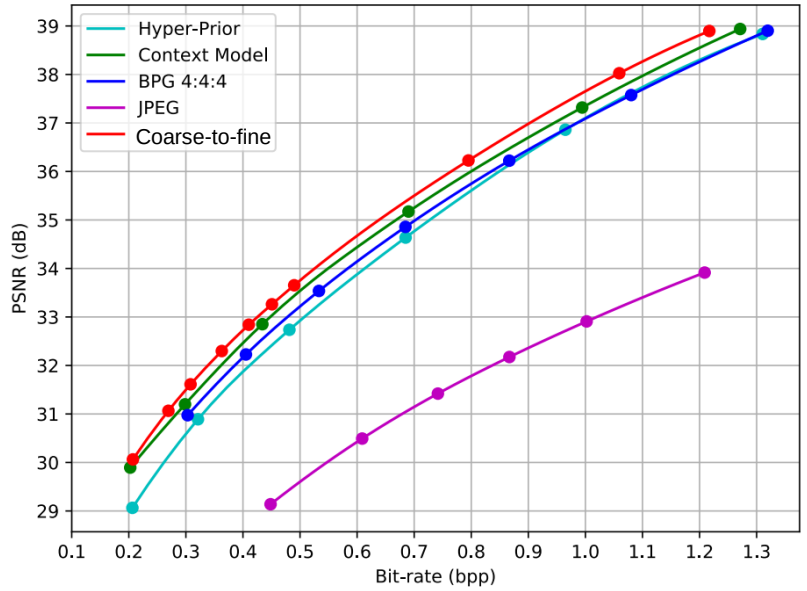


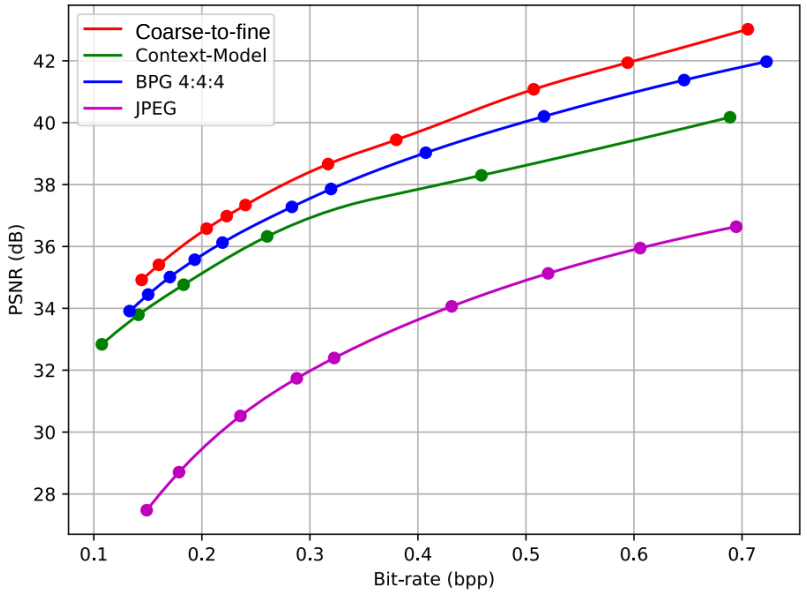
[14] Mentzer, Fabian, et al., "High-Fidelity Generative Image Compression." in NeurIPS. 2020.

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?

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.

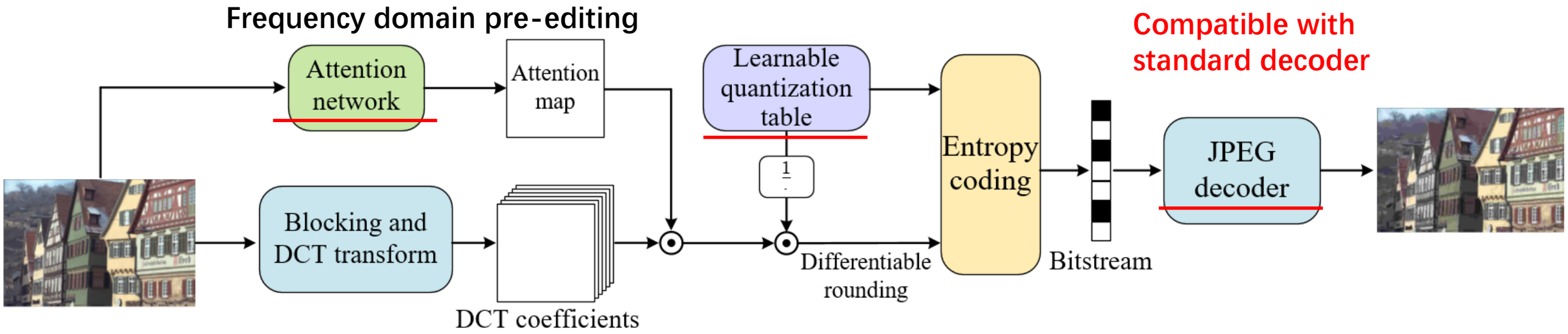
JPEG AI

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.

New Trends in Learned Image Compression

- Learning-based compression compatible with traditional standards (e.g., JPEG)

We made an attempt: [15]

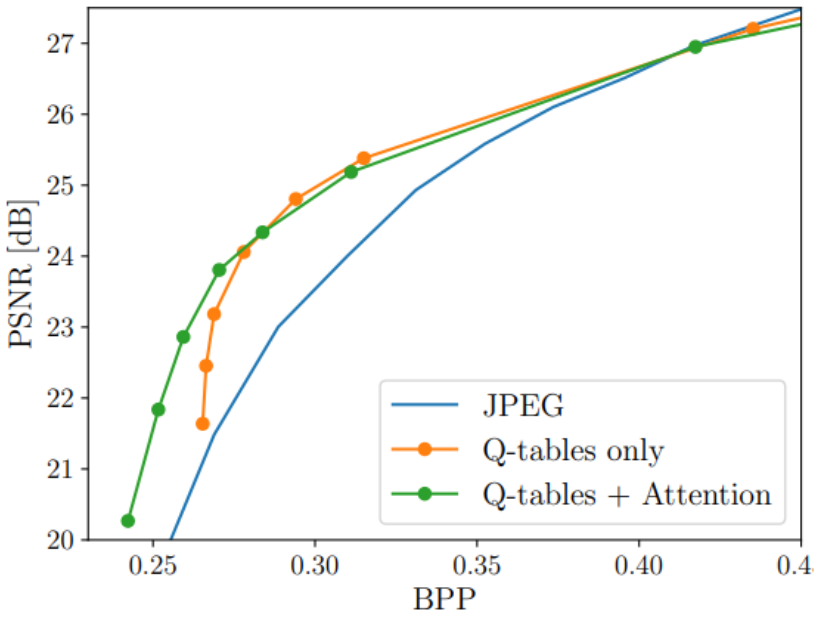


[15] Strümpler, Yannick, et al. "Learning to Improve Image Compression without Changing the Standard Decoder." in ECCVW. 2020.

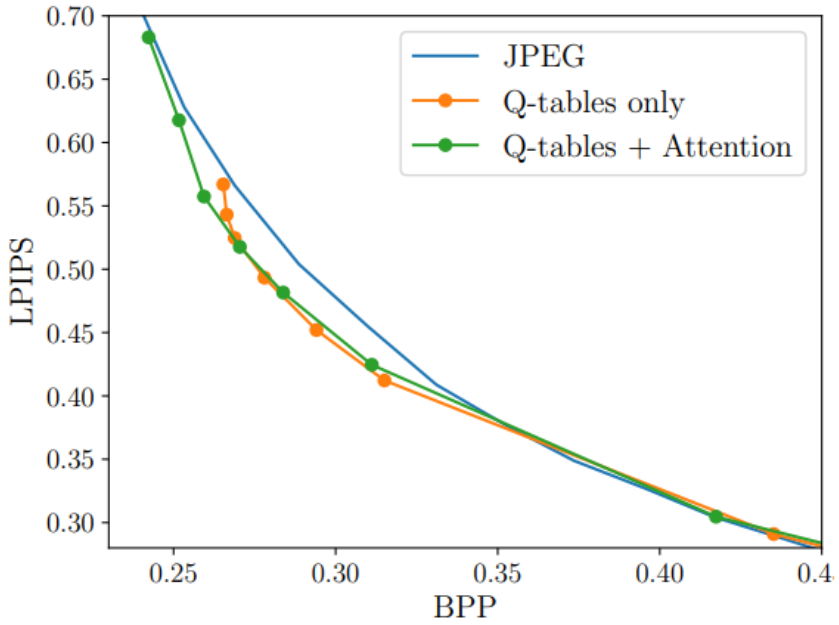
New Trends in Learned Image Compression

- Learning-based compression compatible with traditional standards (e.g., JPEG)

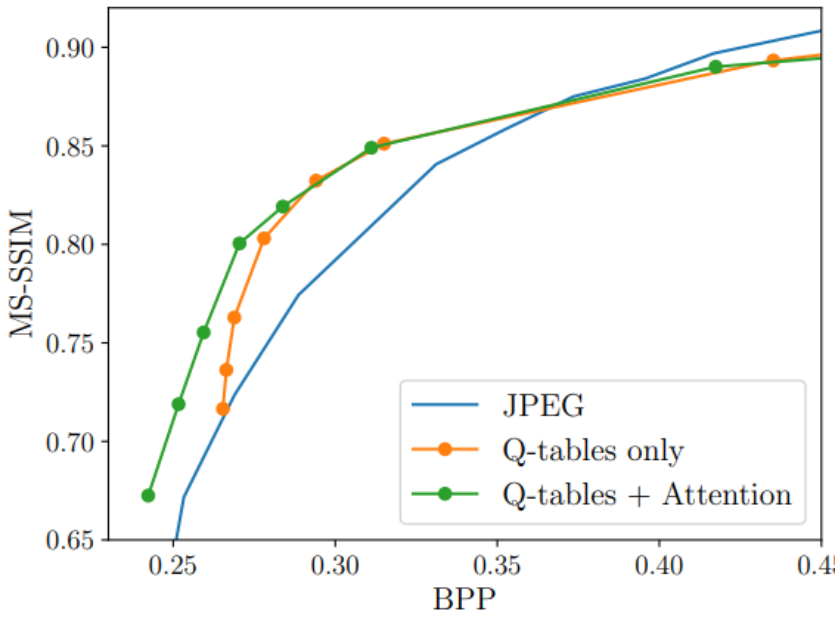
We made an attempt: [15]



PSNR on Kodak



LPIPS on Kodak



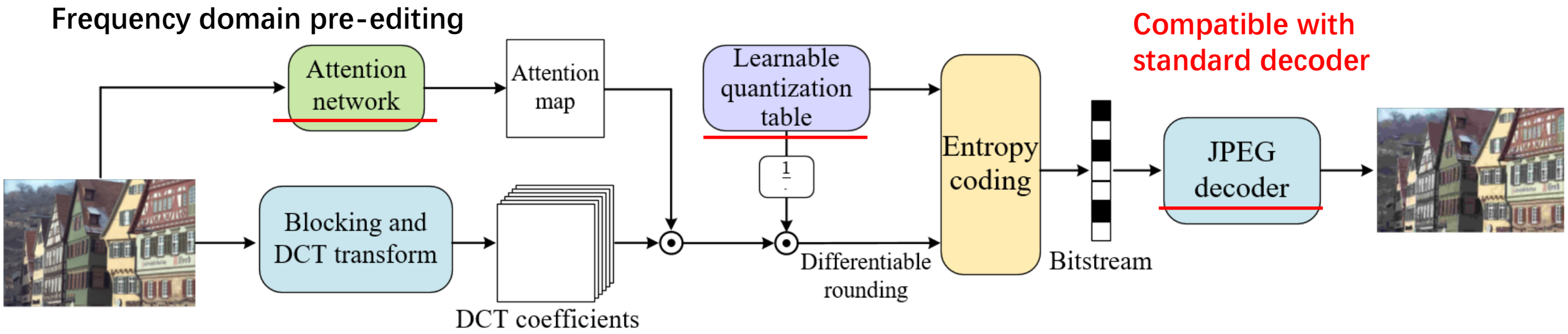
MS-SSIM on Kodak

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

New Trends in Learned Image Compression

- Learning-based compression 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.

New Trends in Learned Image Compression

- Learning-based compression compatible with traditional standards (e.g., JPEG)

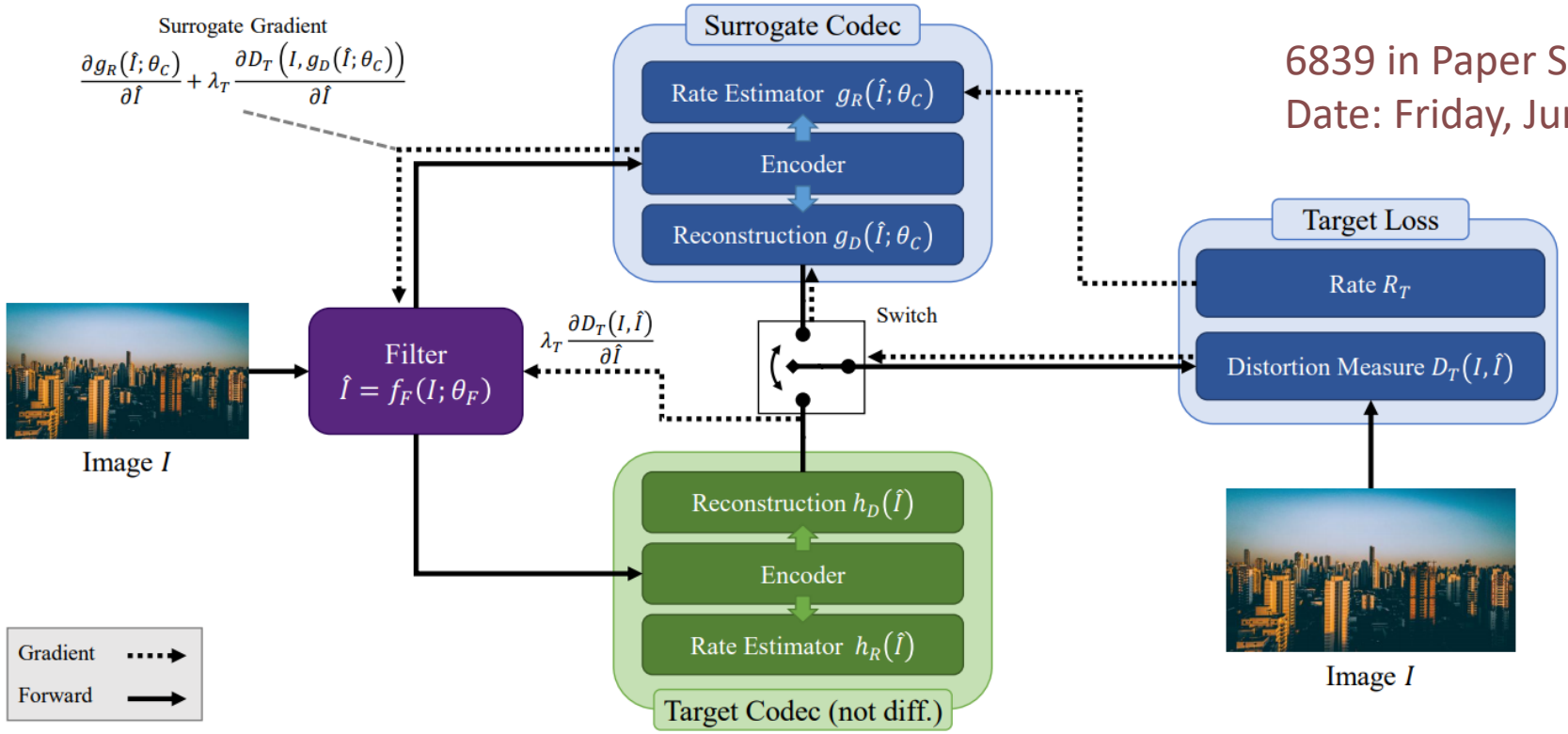


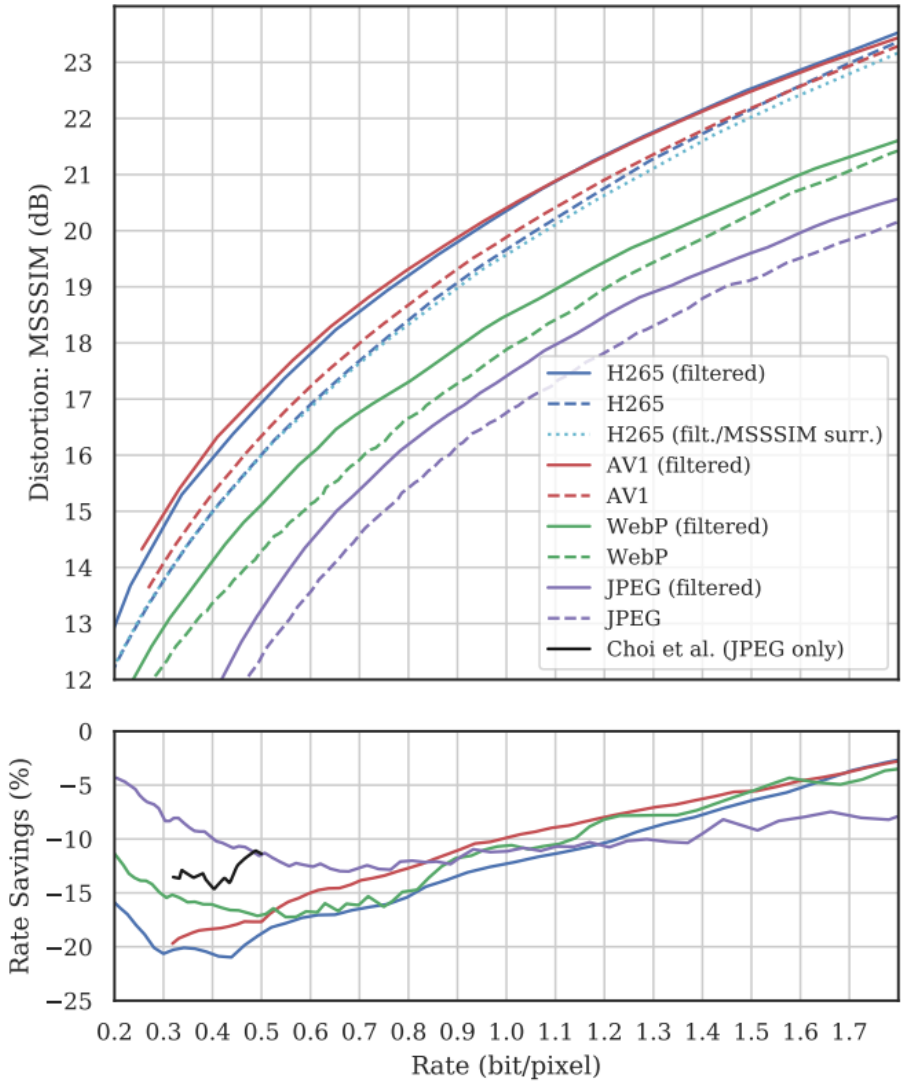
Figure 1. **Structural overview of our method.** The goal is to obtain a trained filter $f_F(I;\theta_F)$ to optimise the input image I for encoding by a target codec. This target is typically not differentiable. A surrogate codec is used instead. It provides a differentiable rate estimate. For the reconstruction there are two options as indicated by the switch. The first is to take the surrogate's reconstruction, the second to invoke the target during the forward pass. The gradient flows back accordingly either through the surrogate's reconstruction or directly into the filter. The target distortion measure D_T can be chosen freely. The surrogate codec is pre-trained with a distortion measure similar to the one of the target codec so as to imitate its behaviour. At testing time, the filter is applied to the image before it's encoded by the target codec.

6839 in Paper Session Twelve
Date: Friday, June 25, 2021 6:00 – 8:30

[16] Klopp, Jan P., et al. "How to Exploit the Transferability of Learned Image Compression to Conventional Codecs." in CVPR. 2021.

New Trends in Learned Image Compression

- Learning-based compression compatible with traditional standards (e.g., JPEG)

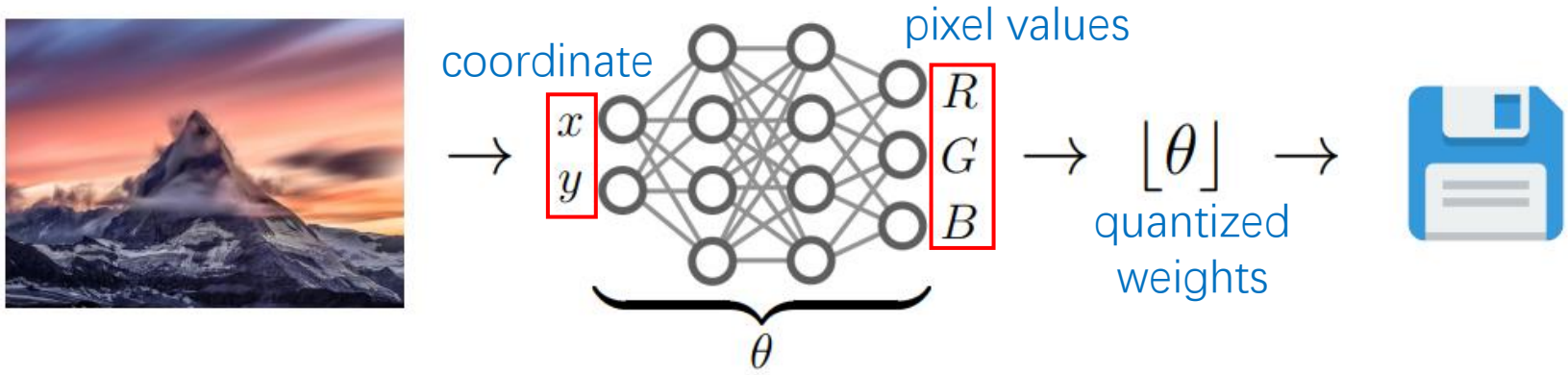


6839 in Paper Session Twelve
Date: Friday, June 25, 2021 6:00 – 8:30

[16] Klopp, Jan P., et al. "How to Exploit the Transferability of Learned Image Compression to Conventional Codecs." in CVPR. 2021.

New Trends in Learned Image Compression

- Learned Image Compression with Implicit Neural Network [17]



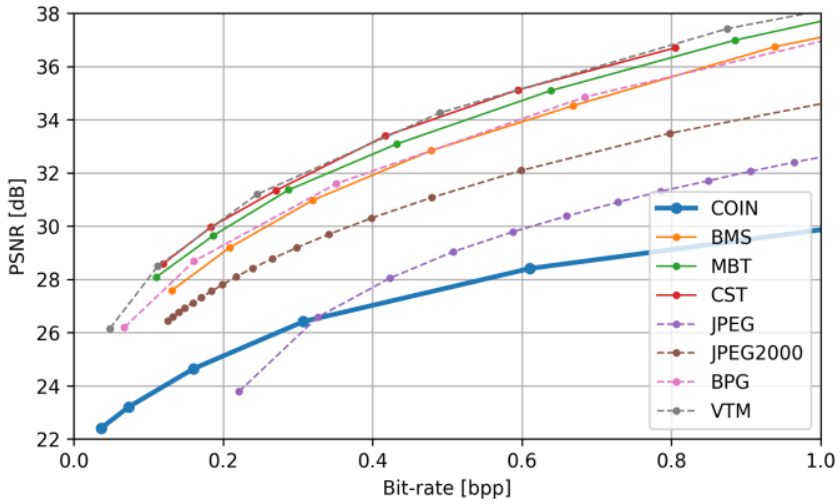


Figure 2: Rate distortion plots on the Kodak dataset.

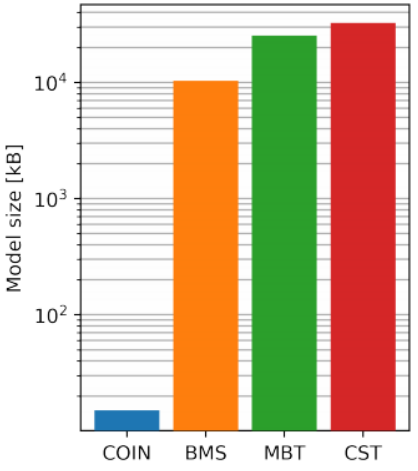


Figure 3: Model sizes at 0.3bpp.

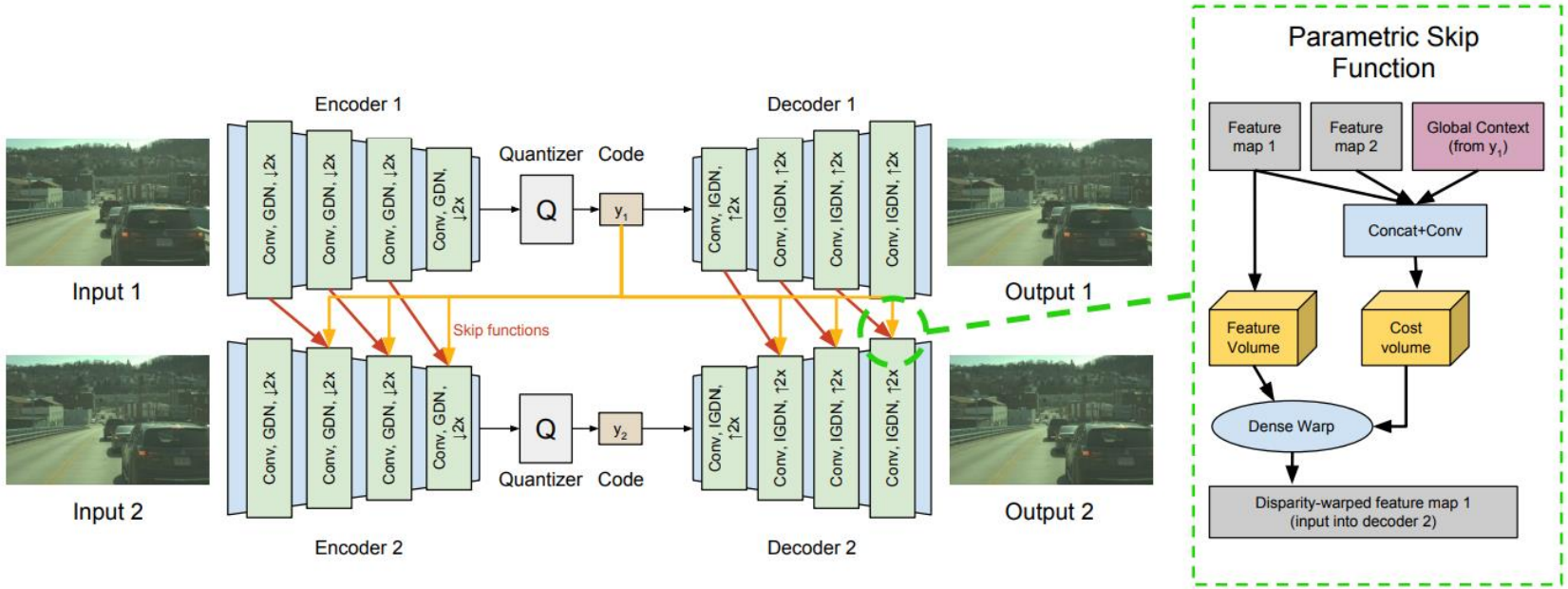
[17] Dupont, Emilien, et al. "COIN: COmpression with Implicit Neural Representations." in ICLR Workshop. 2021.

New Trends in Learned Image Compression

- Learning for Stereo Image Compression

Many applications such as autonomous vehicles and 3D movies involve the use of stereo camera pairs. Stereo image compression can be seen as in-between the work of image and video compression. [15]

DSIC: Deep Stereo Image Compression [18]



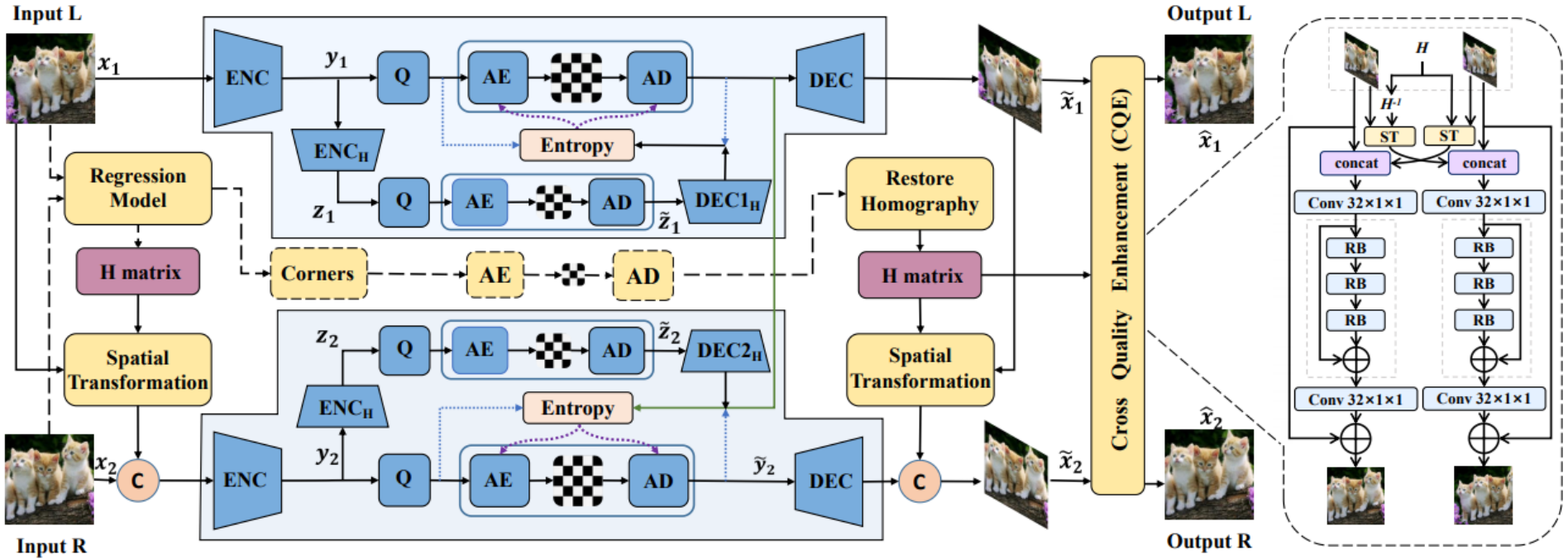
[18] Liu, Jerry, et al. "DSIC: Deep Stereo Image Compression." in ICCV. 2019.

New Trends in Learned Image Compression

- Learning for Stereo Image Compression

6772 in Paper Session Two
Date: Monday, June 21, 2021 22:00 – 24:30

Deep Homography for Efficient Stereo Image Compression [19]



Stereo images are captured at the same time → only angle change w/o temporal motion

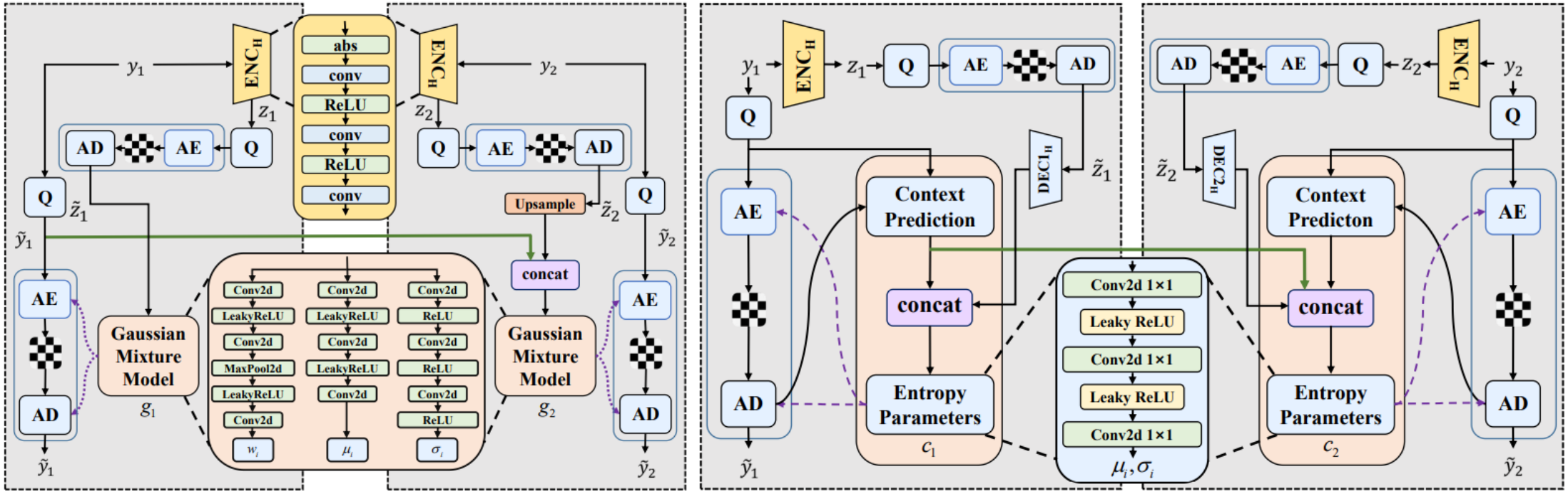
$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

[19] Deng, Xin, et al. "Deep Homography for Efficient Stereo Image Compression." in CVPR. 2021.

New Trends in Learned Image Compression

- Learning for Stereo Image Compression

6772 in Paper Session Two
Date: Monday, June 21, 2021 22:00 – 24:30



(a) GMM-based stereo entropy model.

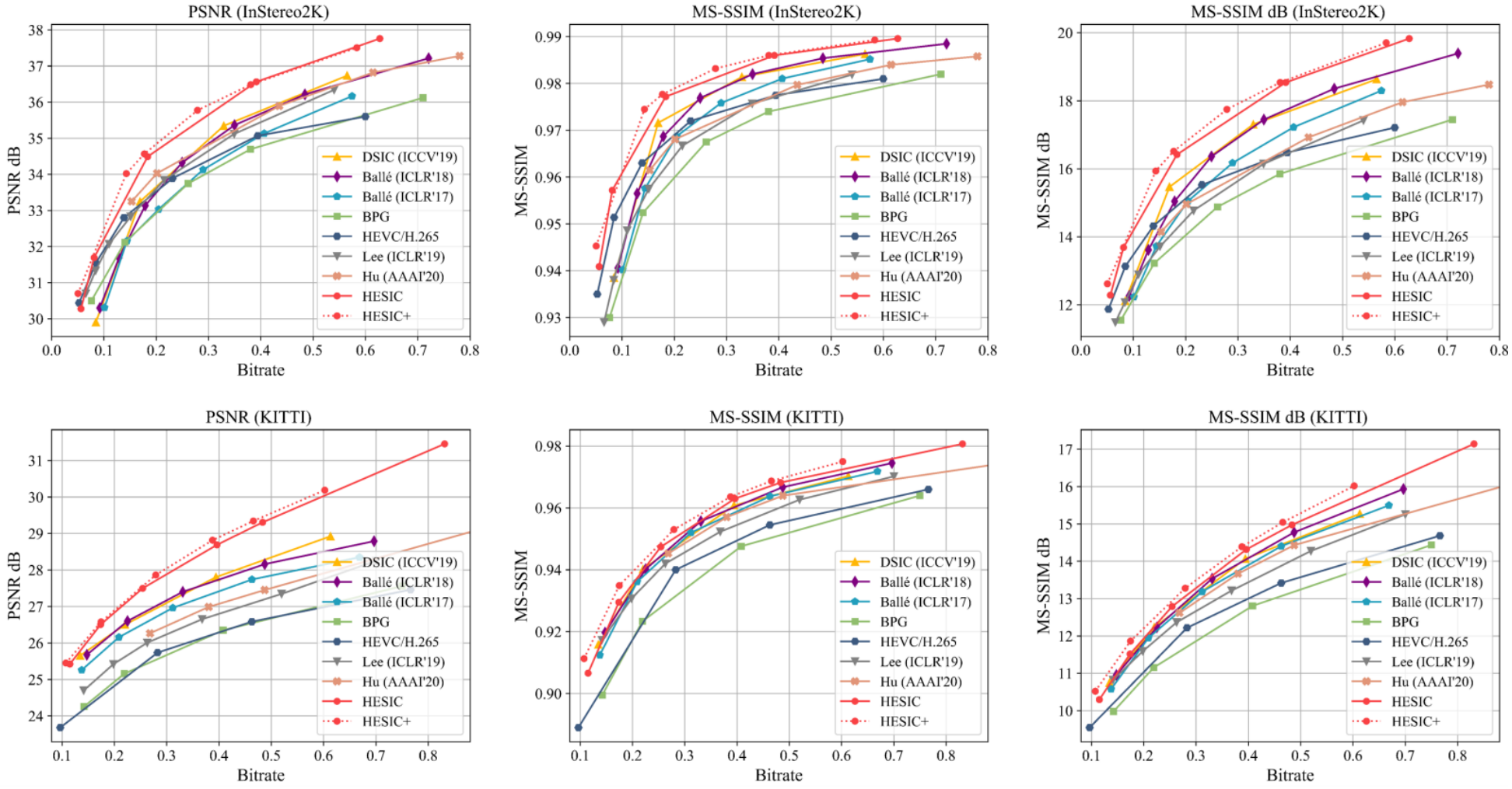
(b) Context-based stereo entropy model.

[19] Deng, Xin, et al. "Deep Homography for Efficient Stereo Image Compression." in CVPR. 2021.

New Trends in Learned Image Compression

- Learning for Stereo Image Compression

6772 in Paper Session Two
Date: Monday, June 21, 2021 22:00 – 24:30



[19] Deng, Xin, et al. "Deep Homography for Efficient Stereo Image Compression." in CVPR. 2021.

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
  - Dupont et al., (compression with implicit neural network): https://github.com/EmilienDupont/coin
  - Deng et al., (stereo image compression): https://github.com/ywz978020607/HESIC

Thanks for attention

Q & A

Ren Yang
ETH Zurich, Switzerland

ren.yang@vision.ee.ethz.ch

Radu Timofte
ETH Zurich, Switzerland

radu.timofte@vision.ee.ethz.ch