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

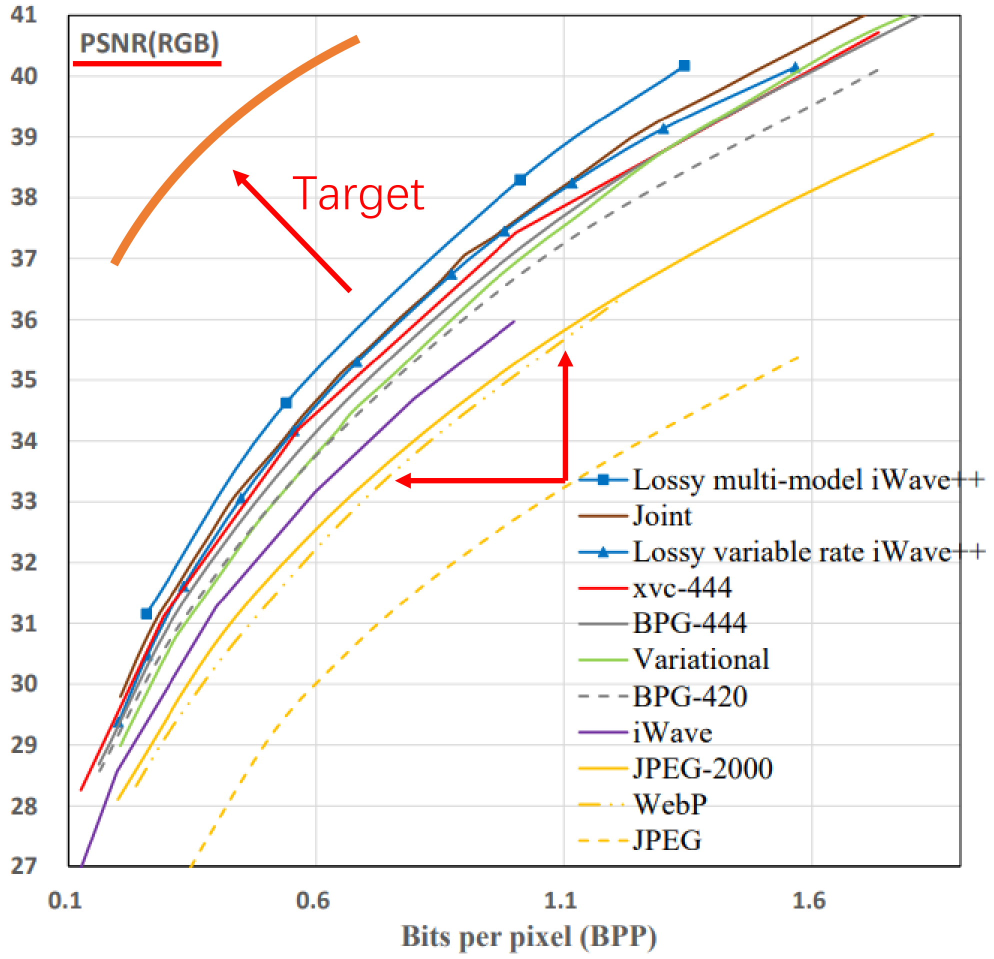
Lossy compression (.jpg): 9 MB

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)



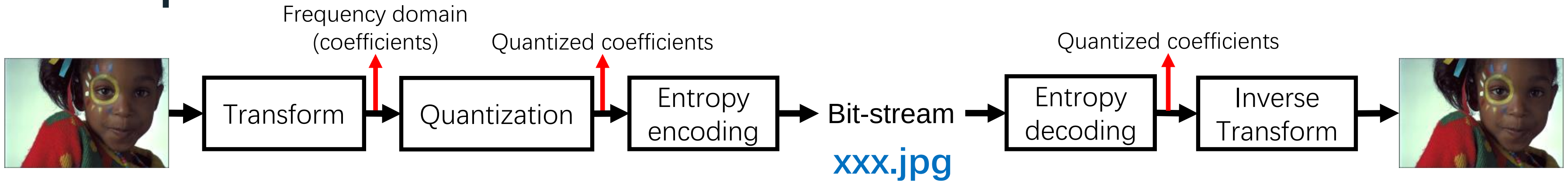
Better quality at the same bit-rate

Lower bit-rate at the same quality

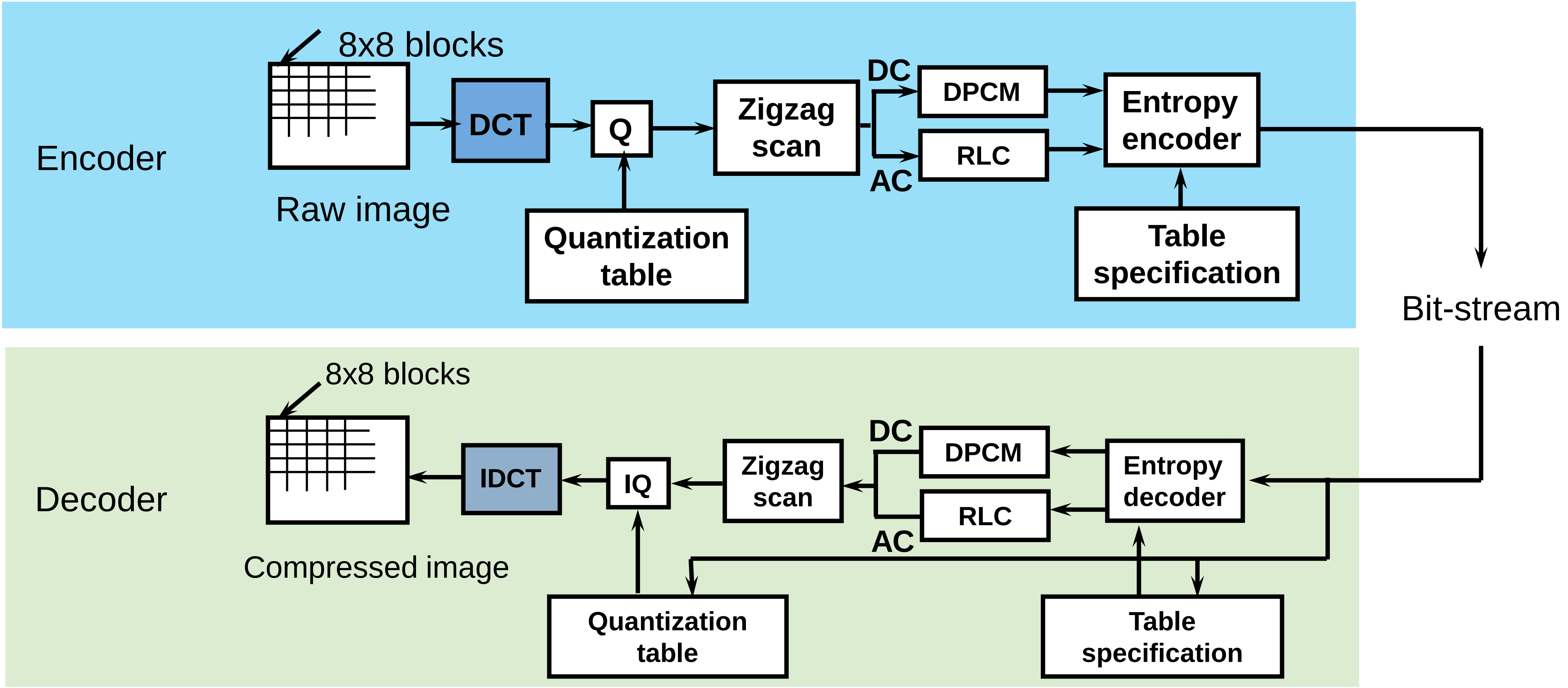
Higher bit-rate (file size)

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:

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

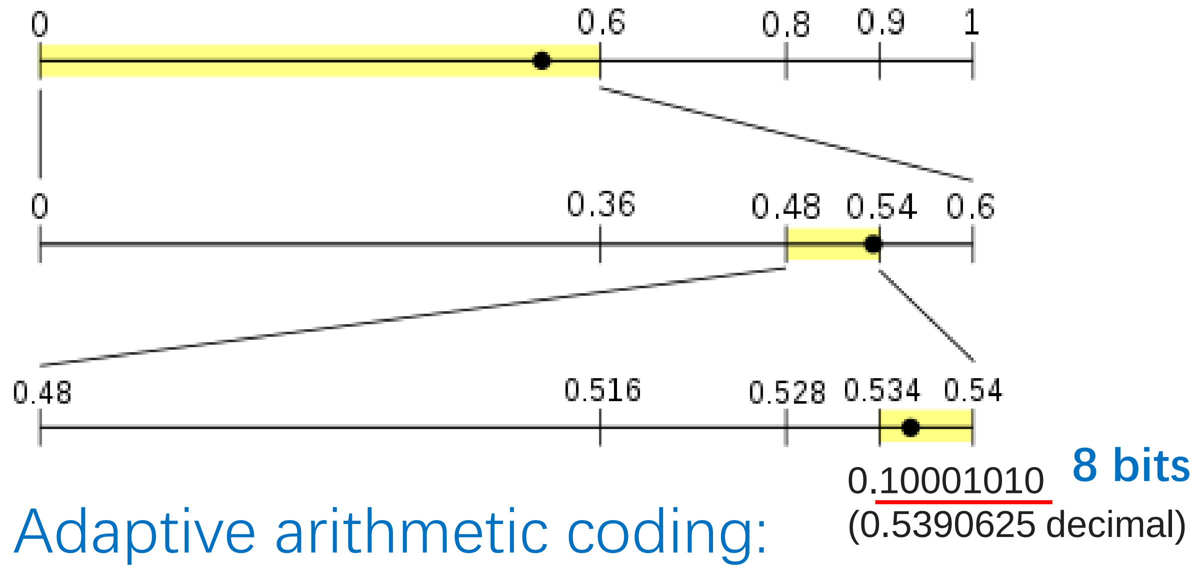
(Adaptive) arithmetic coding is theoretically able to losslessly compress data at

- 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

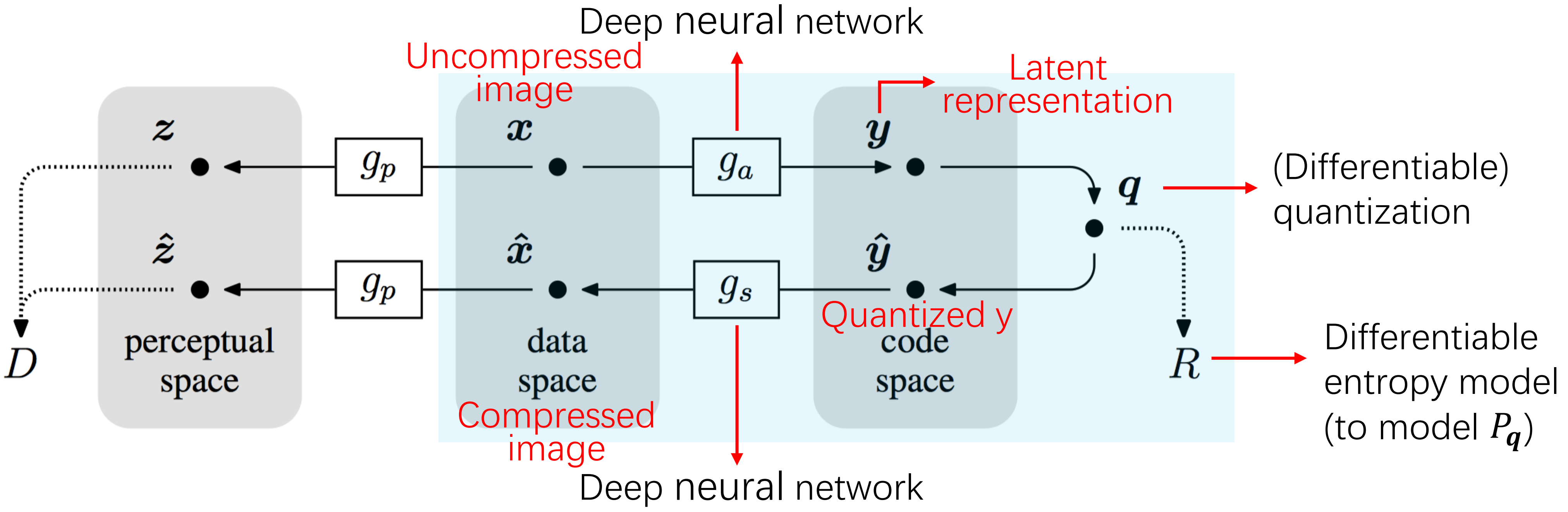


Adaptive arithmetic coding:

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

Learned Image Compression

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

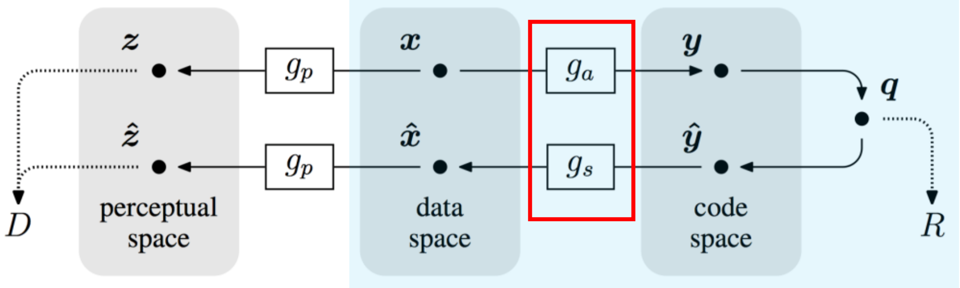


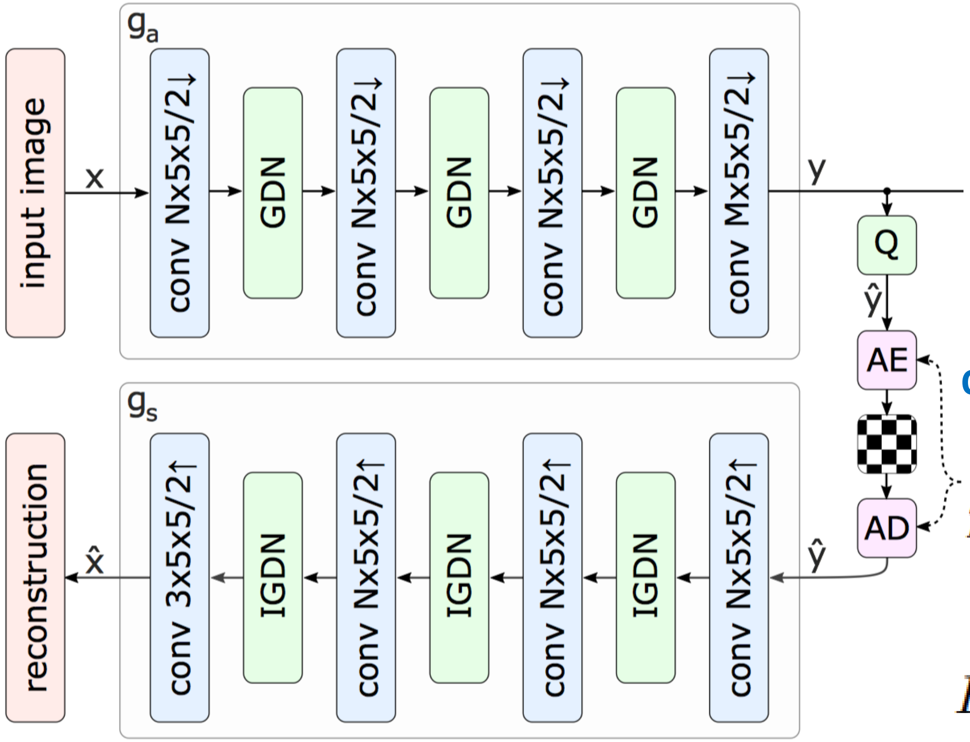
$$L[g_a, g_s, P_{\boldsymbol{q}}] = \underbrace{-\mathbb{E}\big[\log_2 P_{\boldsymbol{q}}\big]}_{R} + \lambda\, \mathbb{E}\big[d(\boldsymbol{x}, \hat{\boldsymbol{x}})\big]$$

[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 $\hat{\boldsymbol{y}}$

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

$$\boxed{\begin{aligned}\boldsymbol{H}^{(k)} &= \mathrm{softplus}(\hat{\boldsymbol{H}}^{(k)}) \\ \boldsymbol{a}^{(k)} &= \tanh(\hat{\boldsymbol{a}}^{(k)})\end{aligned}}$$

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

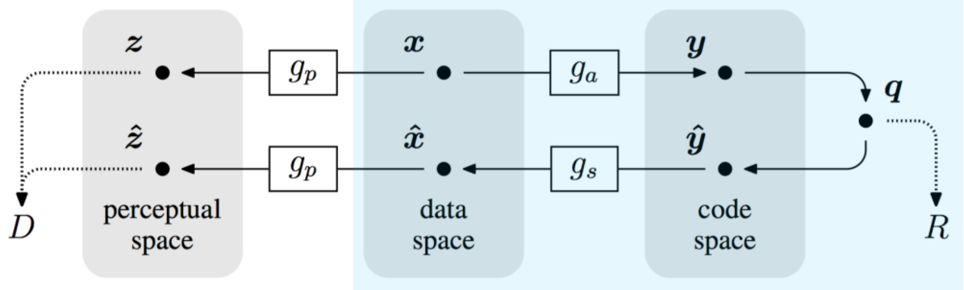
$$L(\boldsymbol{\theta},\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{x},\Delta\boldsymbol{y}}\left[\underbrace{-\sum_i \log_2 p_{\tilde{y}_i}(g_a(\boldsymbol{x};\boldsymbol{\phi}) + \Delta\boldsymbol{y};\boldsymbol{\psi}^{(i)})}_{\text{bit-rate}} + \underbrace{\boxed{\lambda}}_{\text{trade-off}}\underbrace{d\left(g_p(g_s(g_a(\boldsymbol{x};\boldsymbol{\phi})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

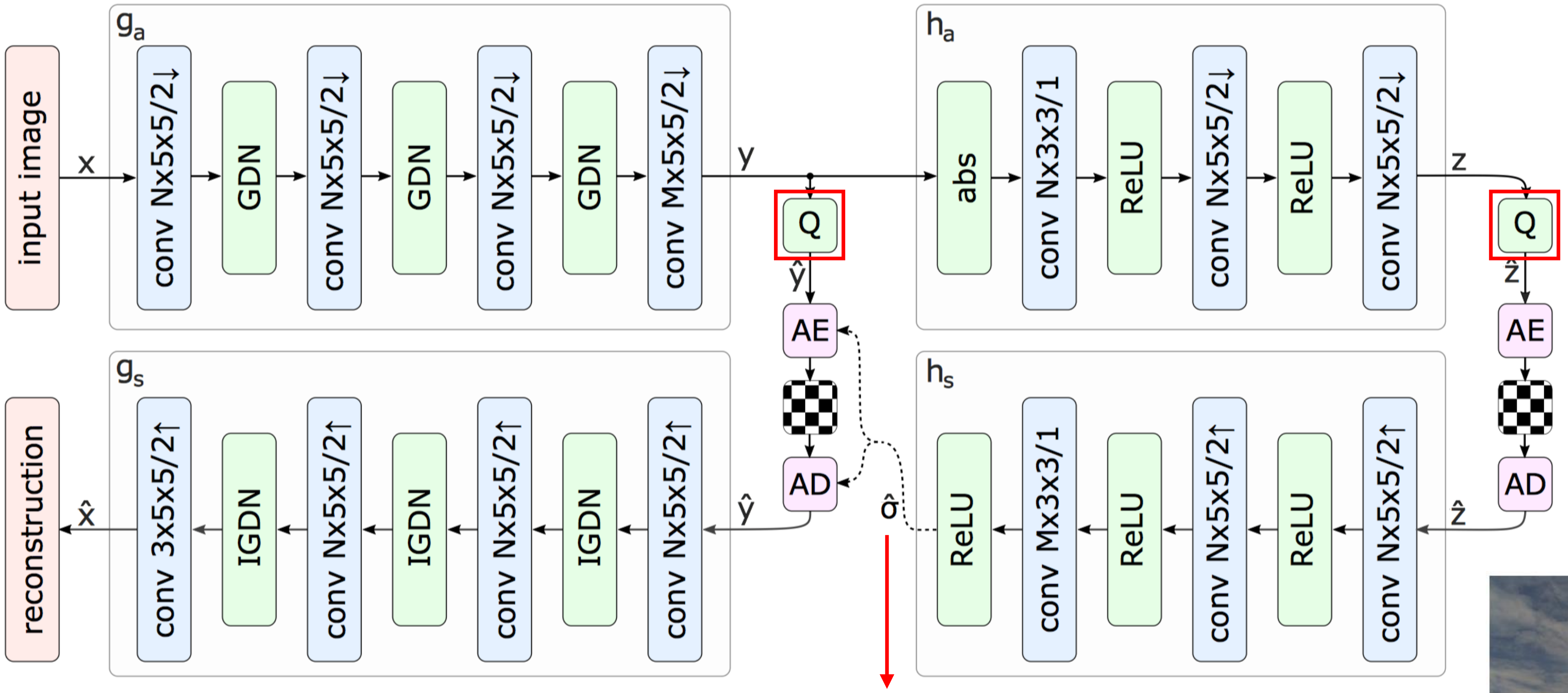
Optimized in an end-to-end manner

[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

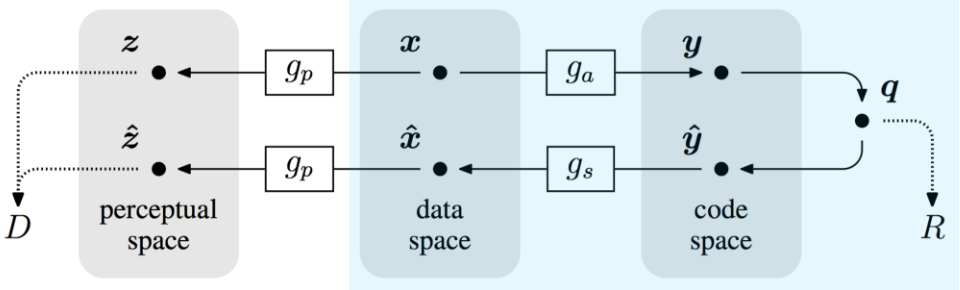
less dependency (less redundancy)

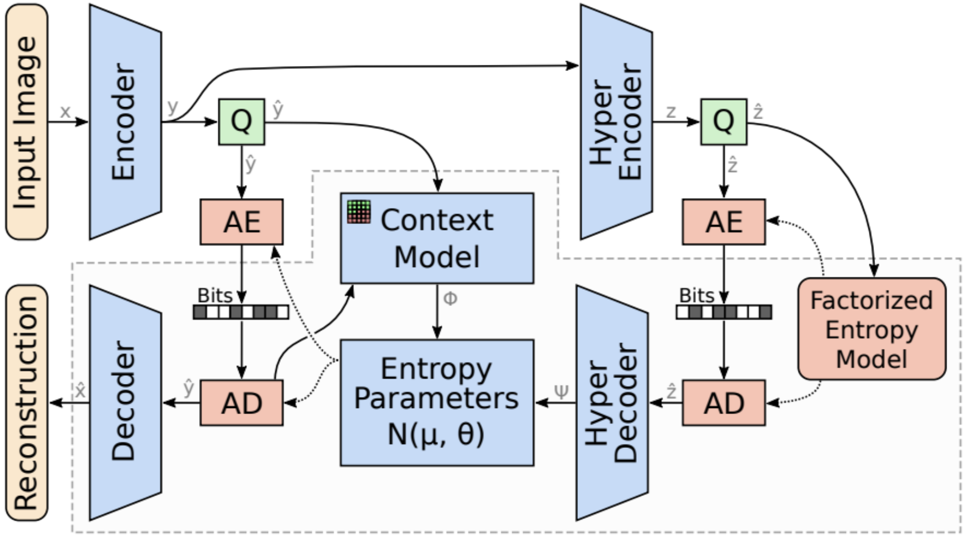
$\boldsymbol{y}$ (factorized) $\boldsymbol{y}/\boldsymbol{\sigma}$ (hyperprior)

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

Learned Image Compression

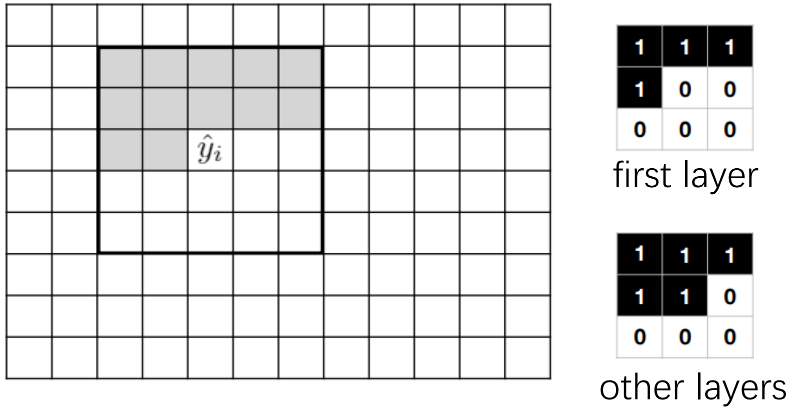
- CNN transformer + **autoregressive** entropy model [3]





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 $d \in \{1, \ldots, f_D\}$ **do**
  for $h \in \{1, \ldots, f_H\}$ **do**
    for $w \in \{1, \ldots, f_W\}$ **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: $p(\boldsymbol{y}) = p(y_1)\cdot p(y_2|y_1)\cdot p(y_3|y_2,y_1)\ldots p(y_N|y_{<N})$

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

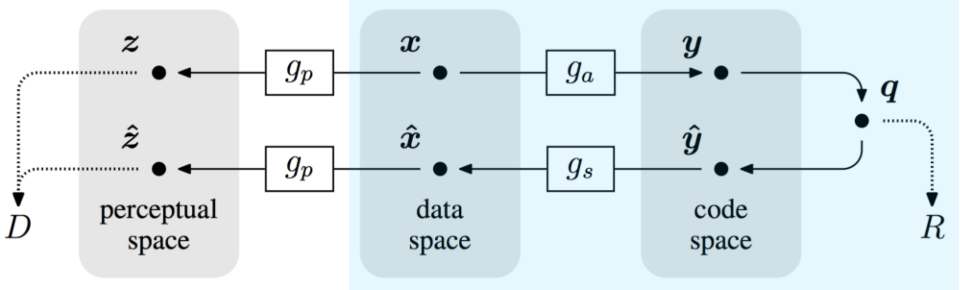
$$p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{z}}, \boldsymbol{\theta}_{hd}, \boldsymbol{\theta}_{cm}, \boldsymbol{\theta}_{ep}) = \prod_i \left(\mathcal{N}(\mu_i, \sigma_i^2) * \mathcal{U}(-\tfrac{1}{2}, \tfrac{1}{2})\right)(\hat{y}_i)$$

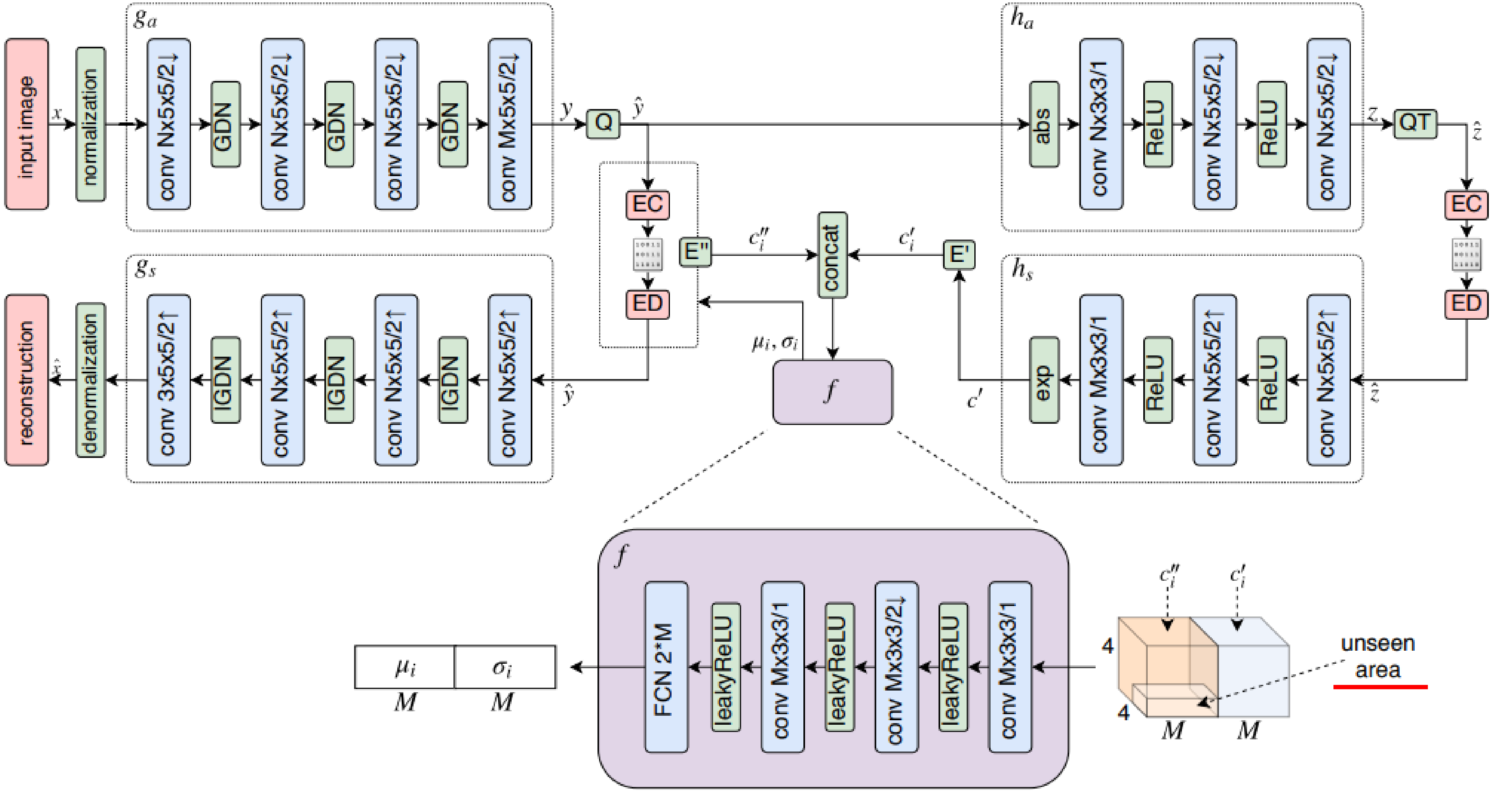
with $\mu_i, \sigma_i = g_{ep}(\boldsymbol{\psi}, \boldsymbol{\phi}_i; \boldsymbol{\theta}_{ep}), \boldsymbol{\psi} = g_h(\hat{\boldsymbol{z}}; \boldsymbol{\theta}_{hd})$, and $\boldsymbol{\phi}_i = g_{cm}(\hat{\boldsymbol{y}}_{<i}; \boldsymbol{\theta}_{cm})$

[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

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

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.

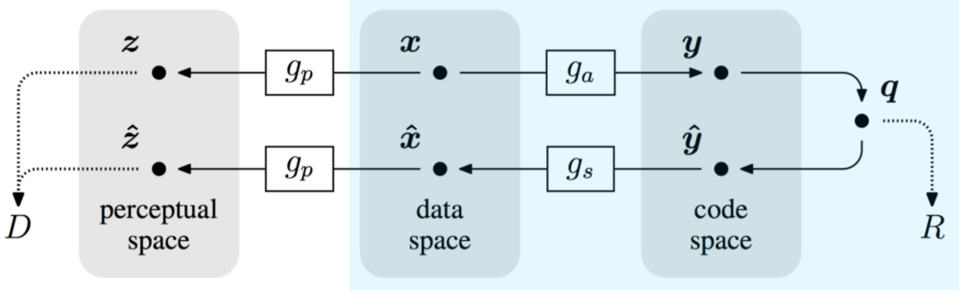
$\boldsymbol{z} = [10℃, 15℃, 30℃]$

$p(\boldsymbol{y} =$ [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

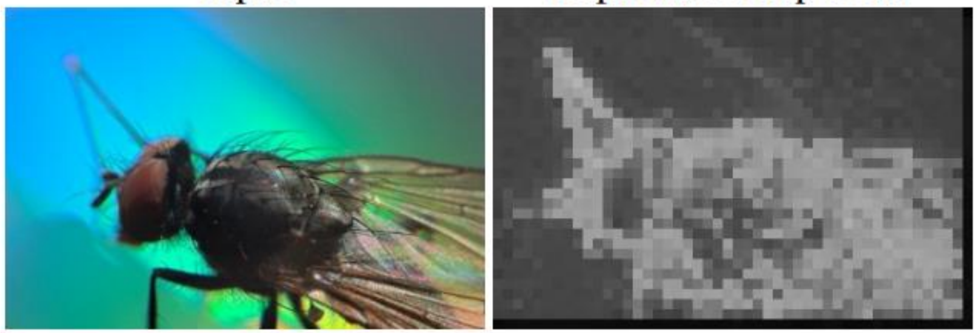
$$\tilde{z}_i = \sum_{j=1}^{L} \frac{\exp(-\sigma\|z_i - c_j\|)}{\sum_{l=1}^{L} \exp(-\sigma\|z_i - c_l\|)} c_j$$

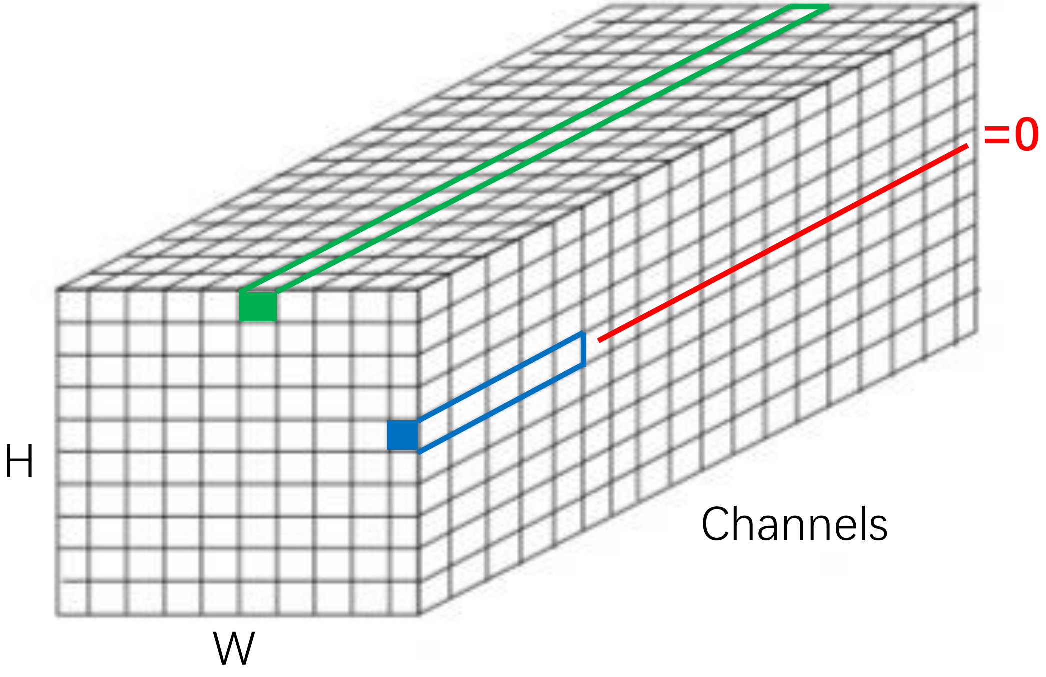
Training: differentiable

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

- Importance map [4]

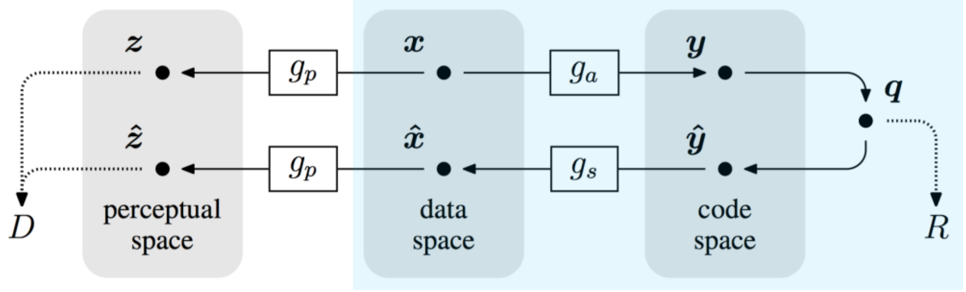
Input | Importance map of M





[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

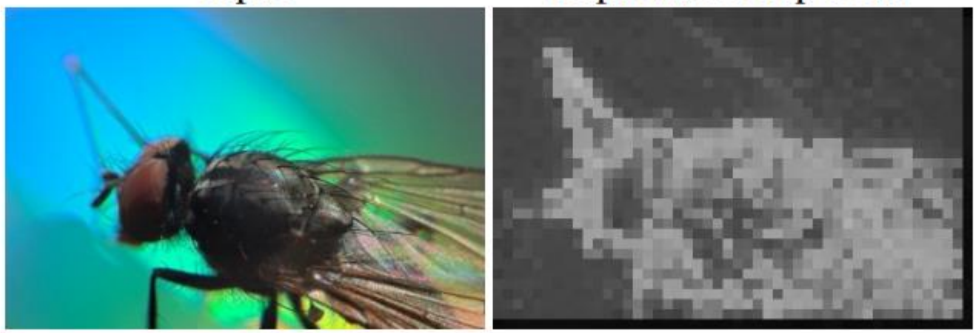
$$\tilde{z}_i = \sum_{j=1}^{L} \frac{\exp(-\sigma\|z_i - c_j\|)}{\sum_{l=1}^{L} \exp(-\sigma\|z_i - c_l\|)} c_j$$

Training: differentiable

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

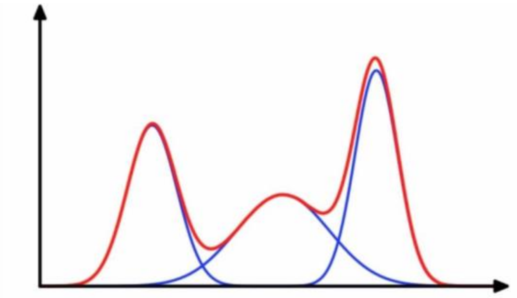
- Importance map [4]

Input | Importance map of M



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

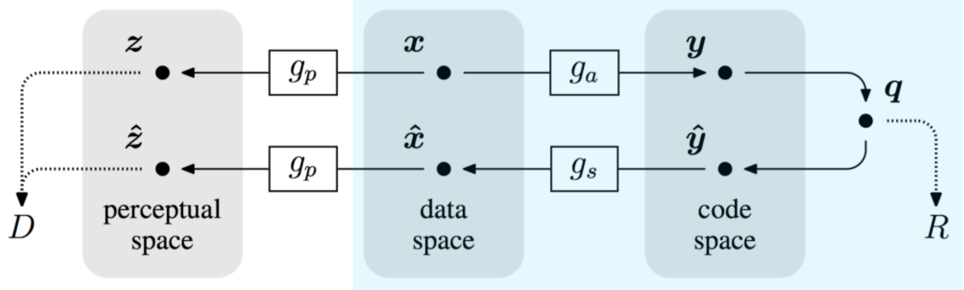
$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) \sim \sum_{k=1}^{K} \boldsymbol{w}^{(k)} \mathcal{N}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{2(k)})$$

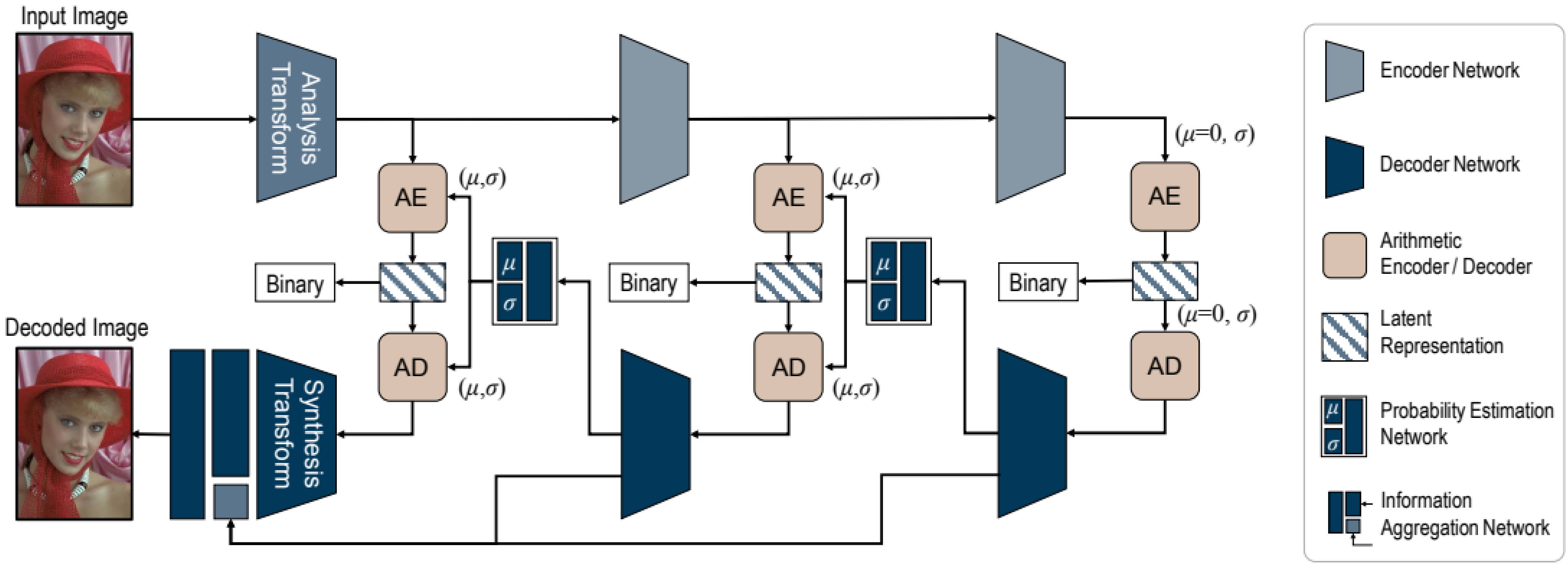


[4] Mentzer, Fabian, et al. “Conditional Probability Models for Deep Image Compression”, in CVPR, 2018.
[6] Cheng et al. ”Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules”, in CVPR. 2020.

Learned Image Compression

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

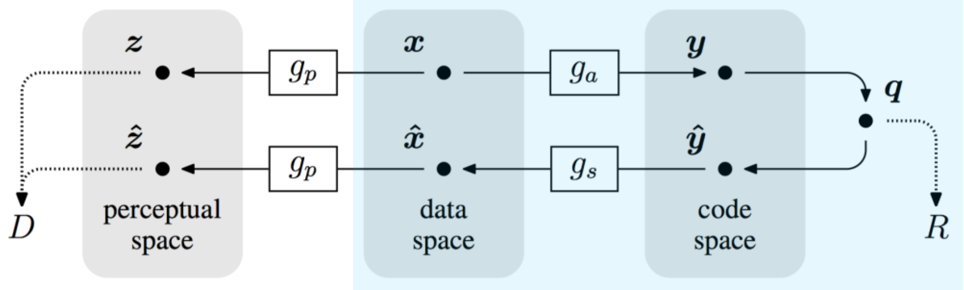


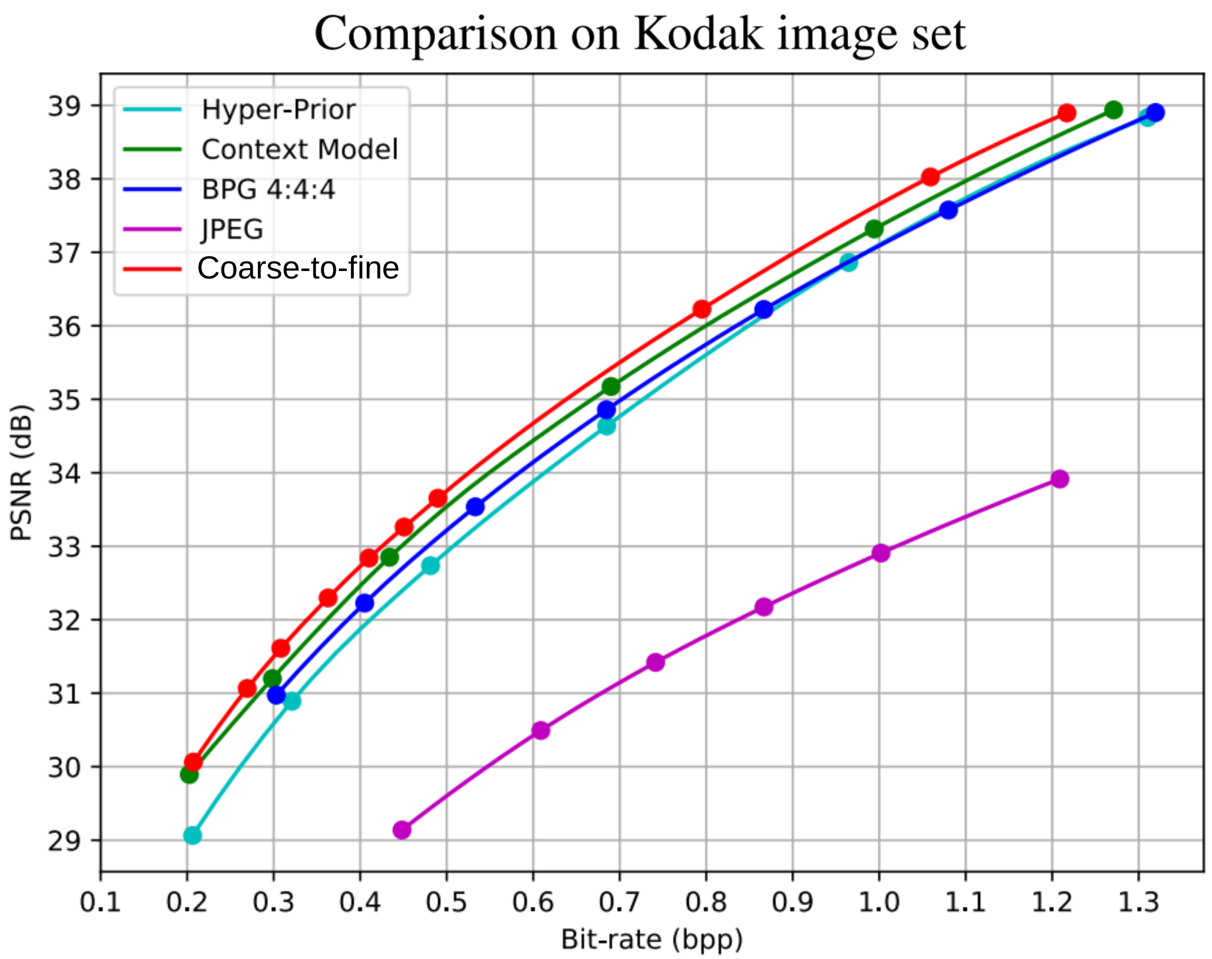


[7] Hu, Yueyu, et al. "Coarse-to-Fine Hyper-Prior Modeling for Learned Image Compression." in AAAI. 2020.

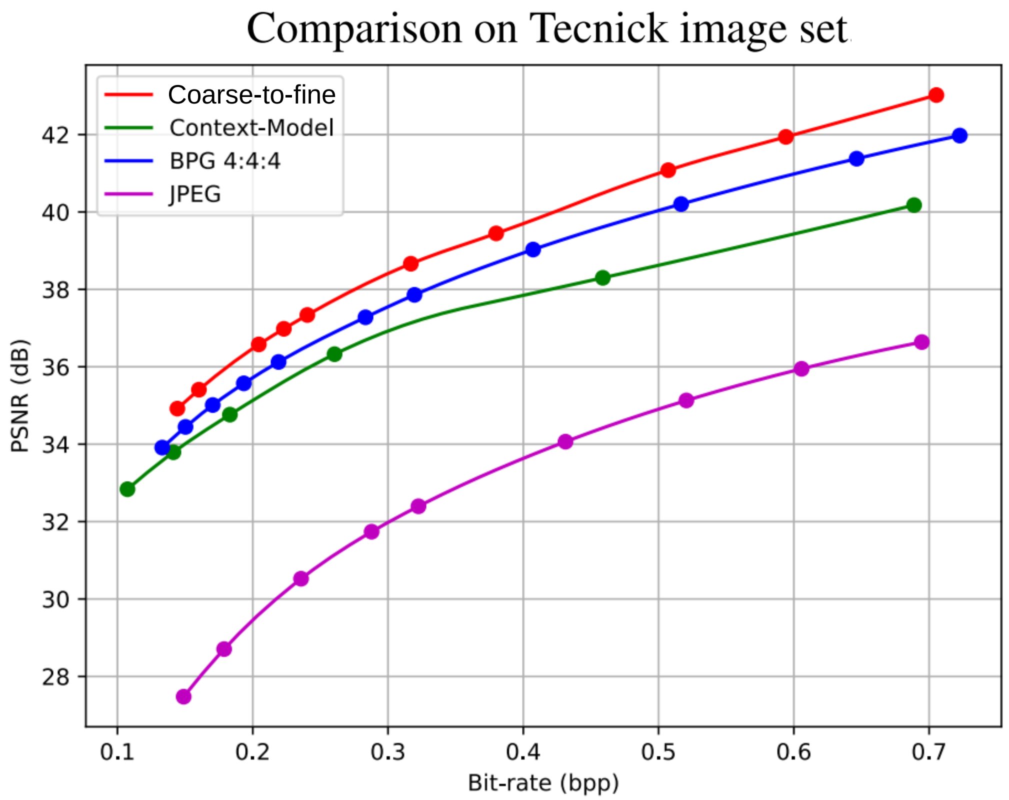
Learned Image Compression

- Performance





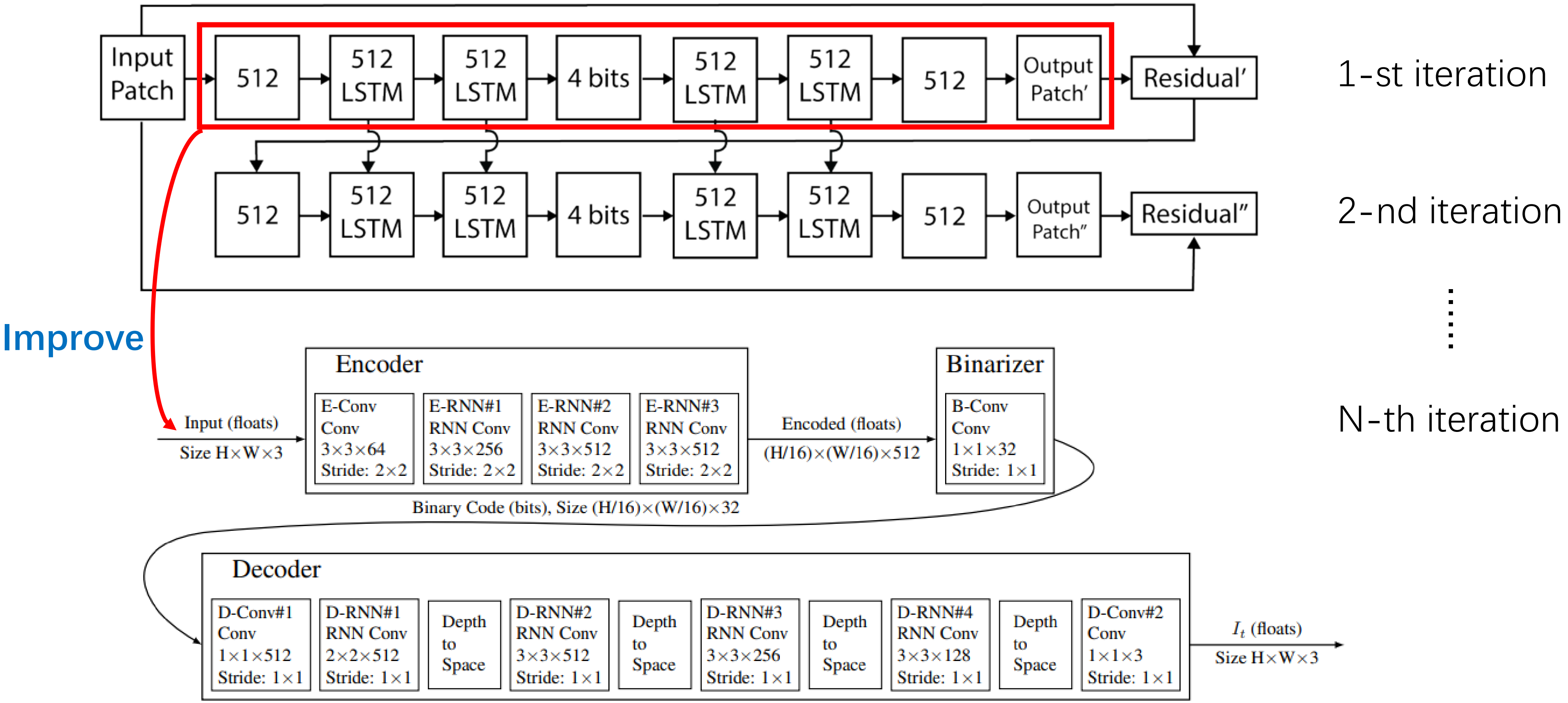
The context (autoregressive) and coarse-to-fine models outperform BPG 4:4:4 (latest traditional standard)



The rank may vary on different datasets

Learned Image Compression

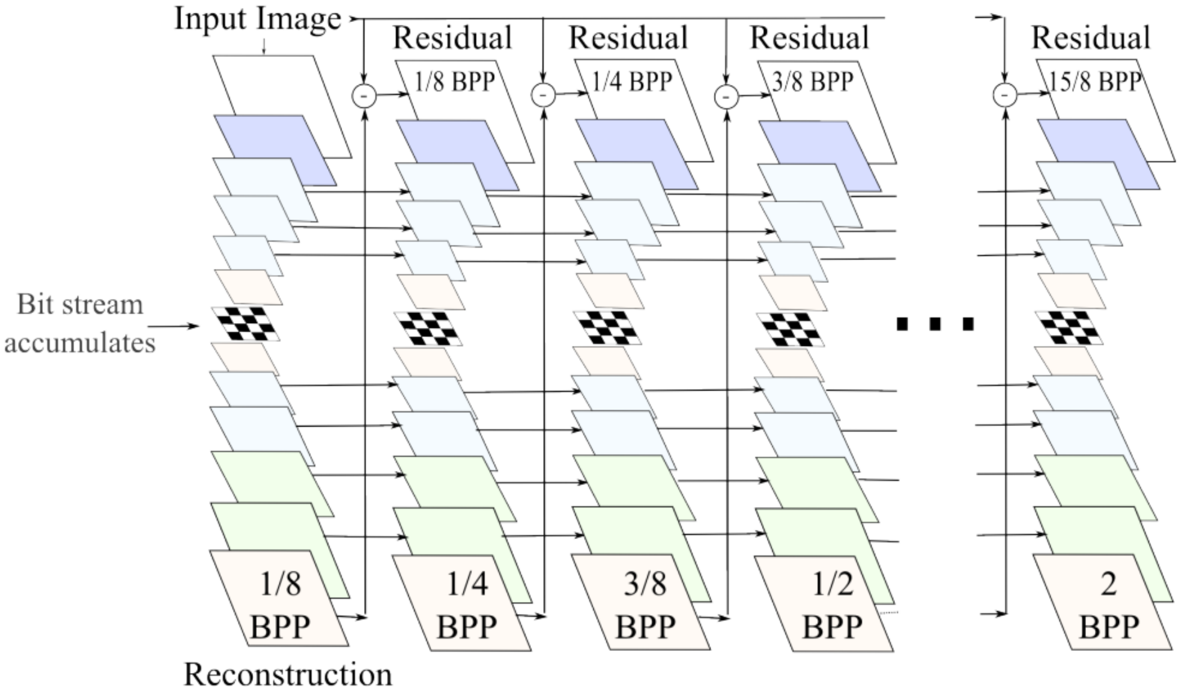
- Variable rate image compression: RNN-based methods [8, 9]



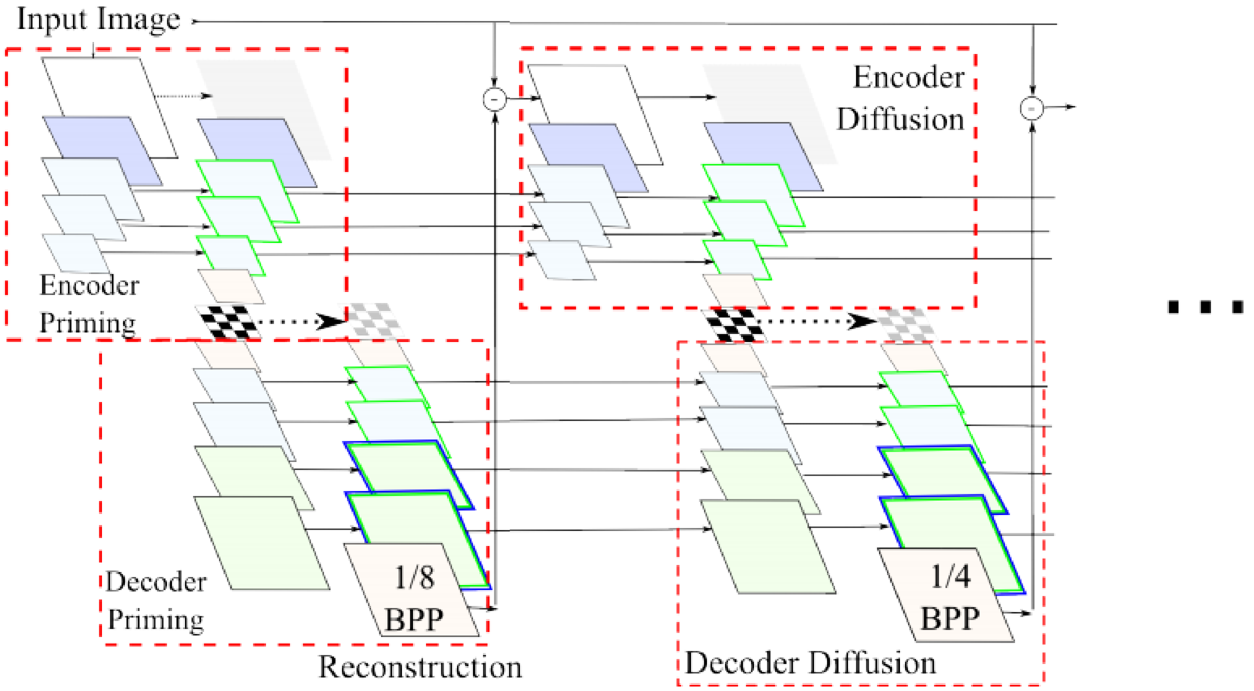
[8] Toderici, George, et al. "Variable Rate Image Compression with Recurrent Neural Networks." in ICLR. 2016.
[9] Toderici, George, et al. "Full Resolution Image Compression with Recurrent Neural Networks." in CVPR, 2017.

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.

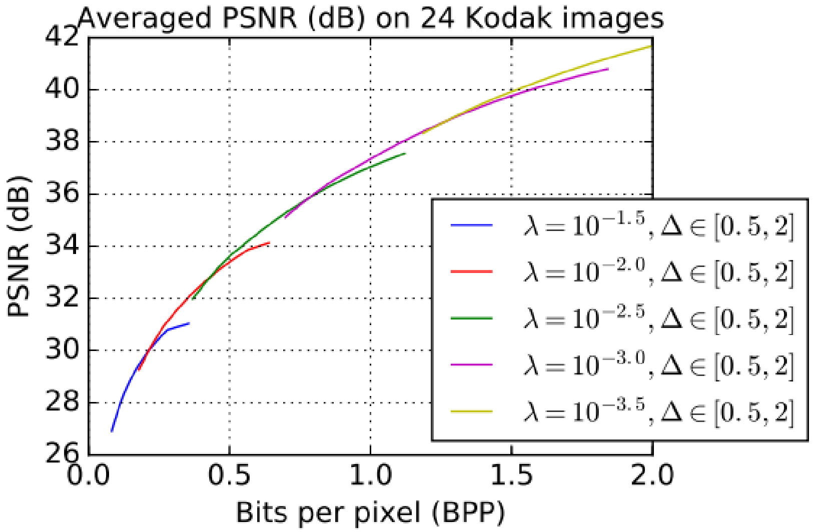
Learned Image Compression

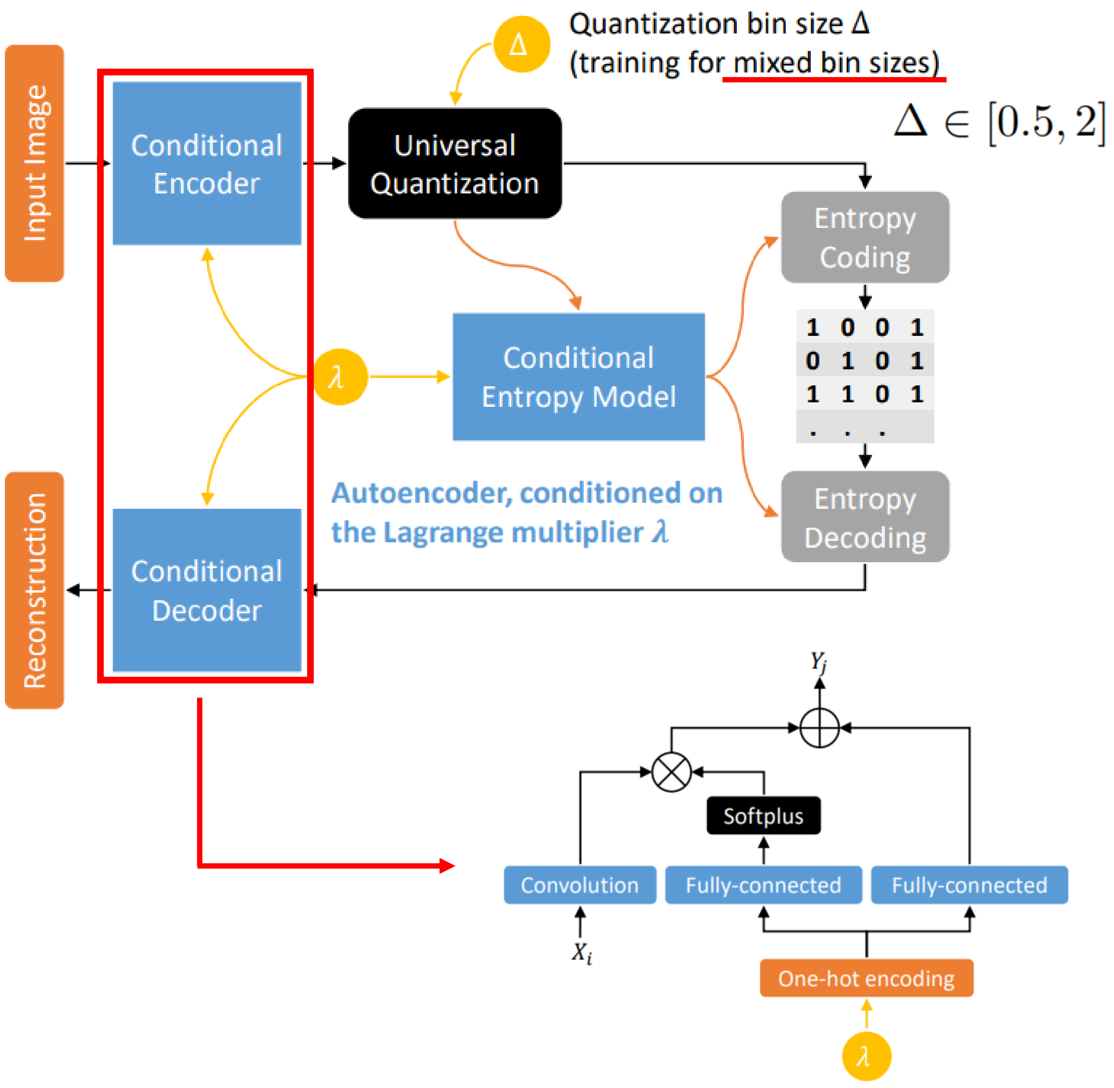
- Variable rate image compression: Conditional autoencoder [11]

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

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

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

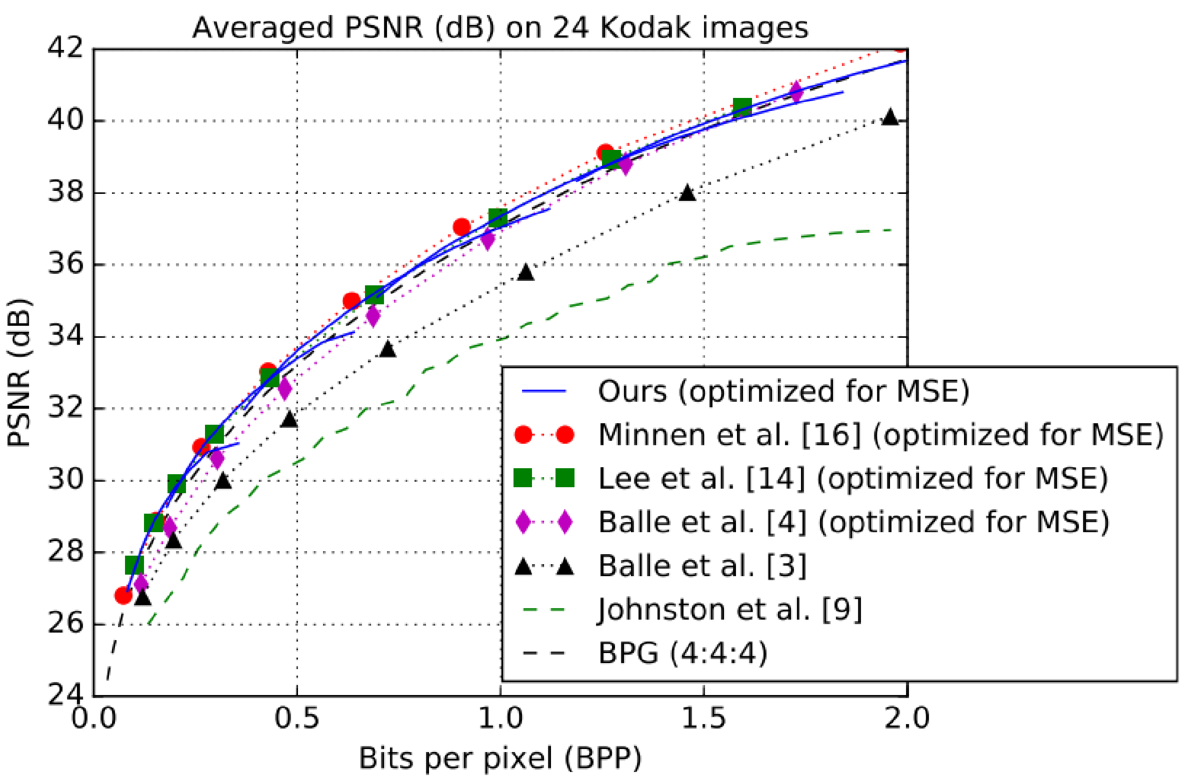


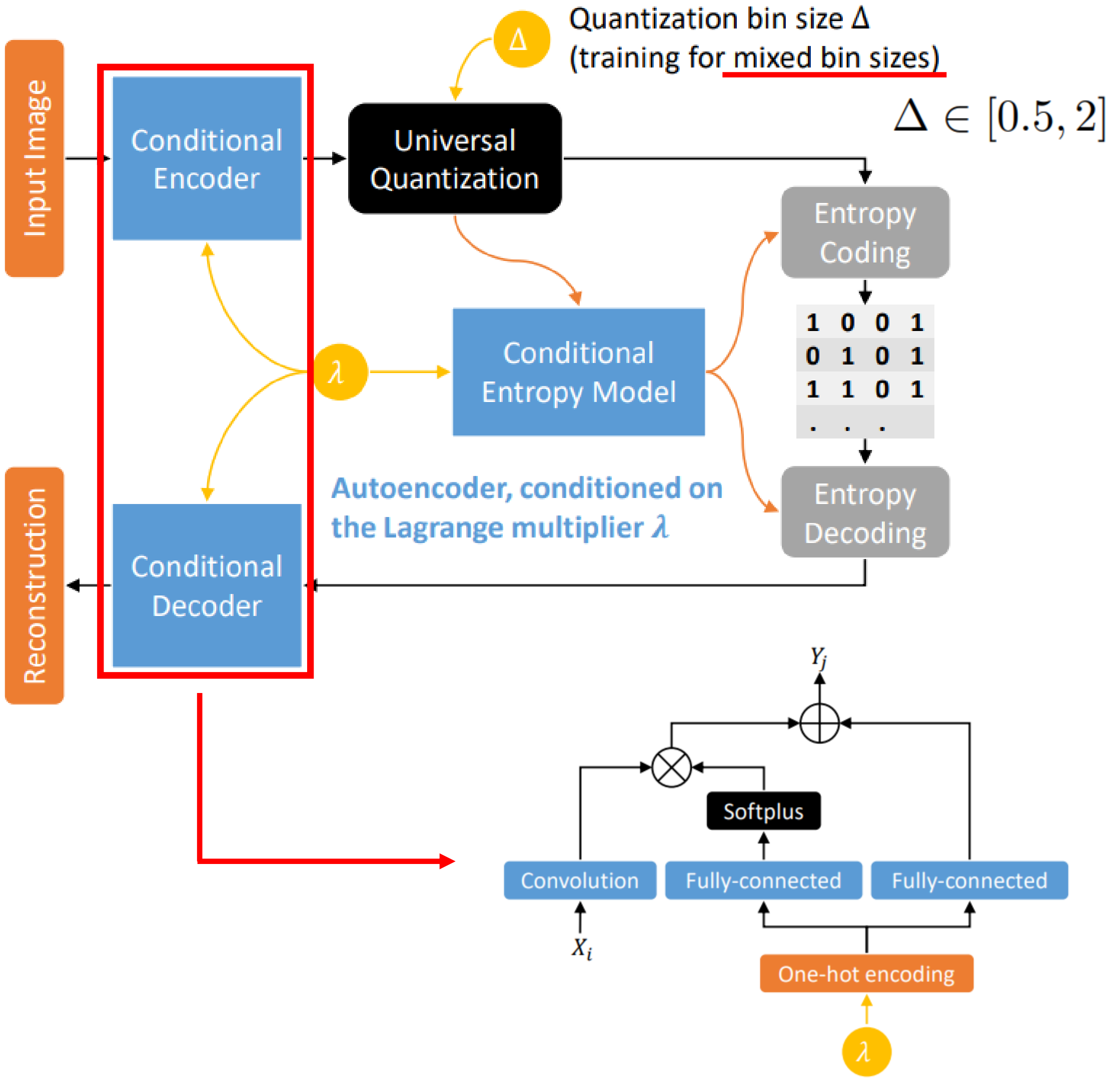


[11] Choi, Yoojin, et al. "Variable Rate Deep Image Compression With a Conditional Autoencoder." in ICCV. 2019.

Learned Image Compression

- Variable rate image compression: Conditional autoencoder [11]

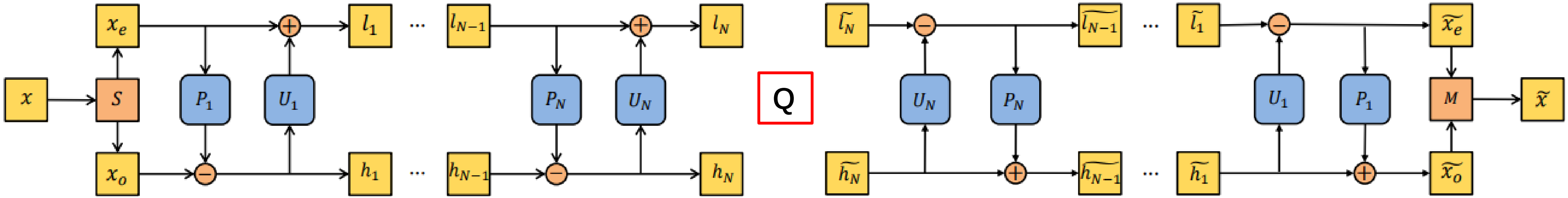




[11] Choi, Yoojin, et al. "Variable Rate Deep Image Compression With a Conditional Autoencoder." in ICCV. 2019.

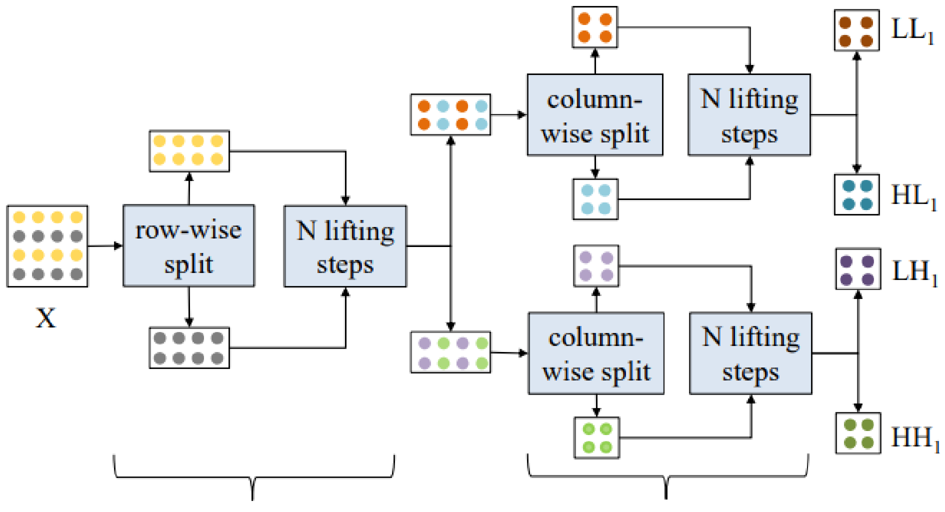
Learned Image Compression

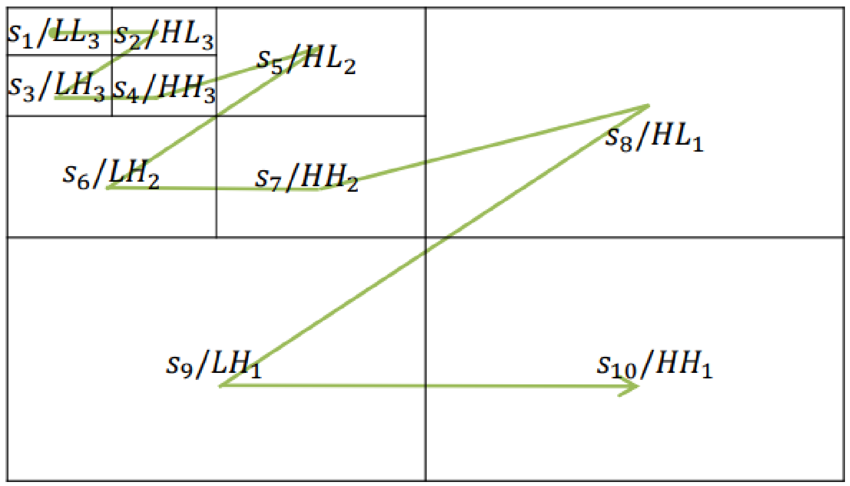
- Variable rate image compression: Wavelet-like transformer [12]



(a) Forward transform (b) Inverse transform

Invertible: achieving lossy and lossless compression by the same framework

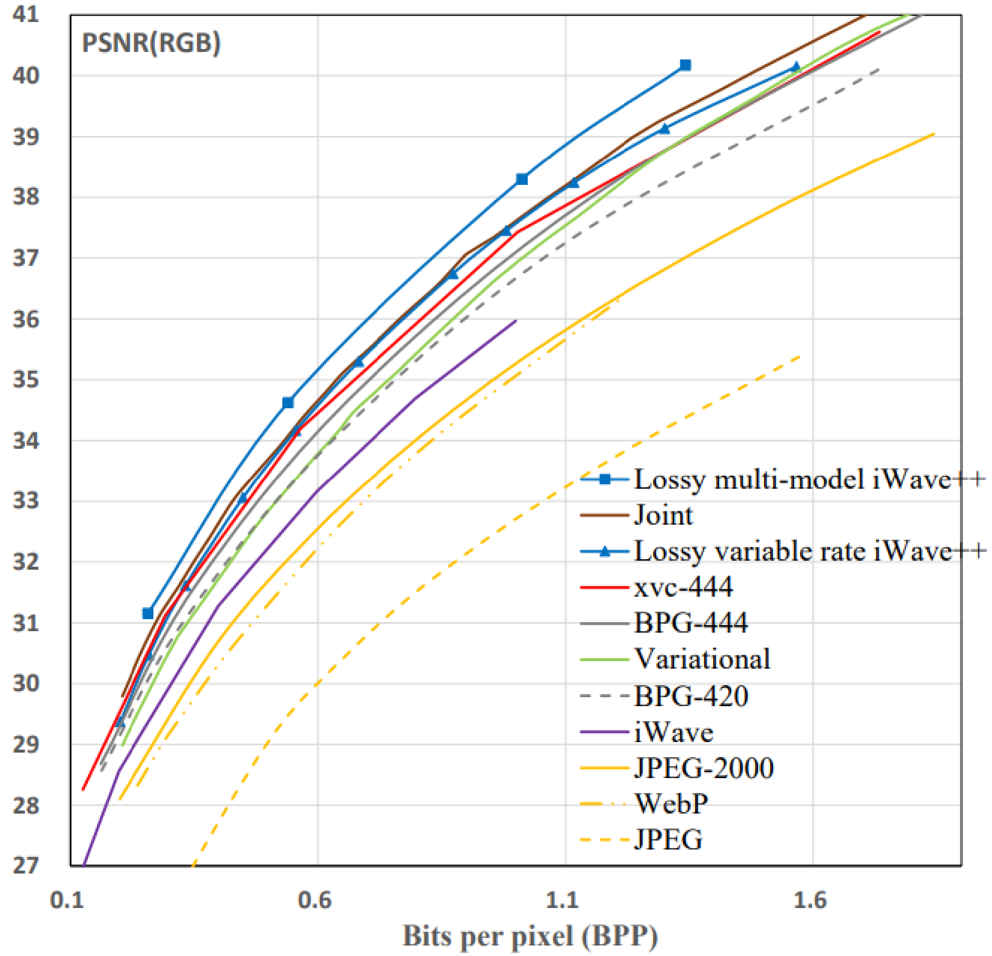




[12] Ma, Haichuan, et al. "End-to-End Optimized Versatile Image Compression With Wavelet-Like Transform." in IEEE T-PAMI. 2020.

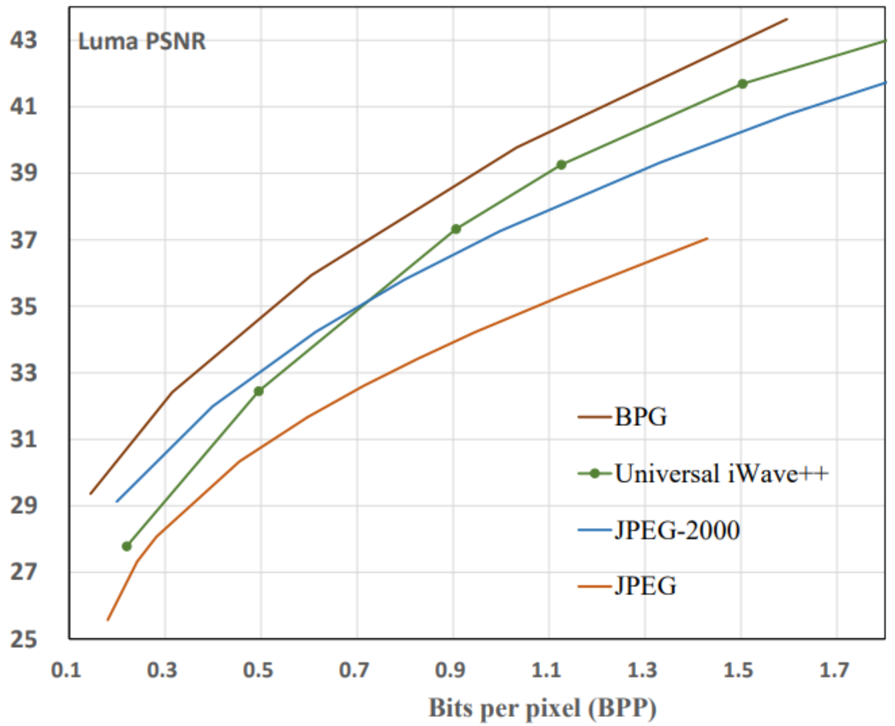
Learned Image Compression

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



(one model for each point)

(one model for all points)

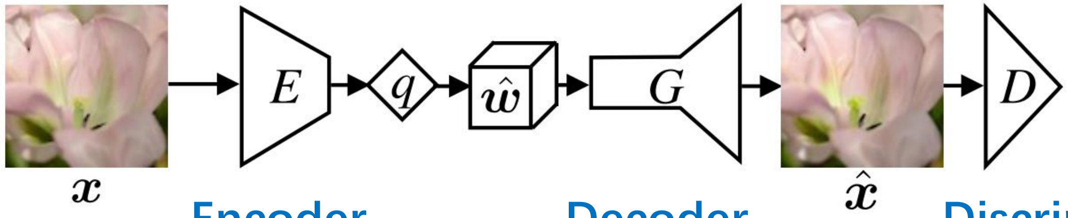


one model for both lossy and lossless compression

[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]



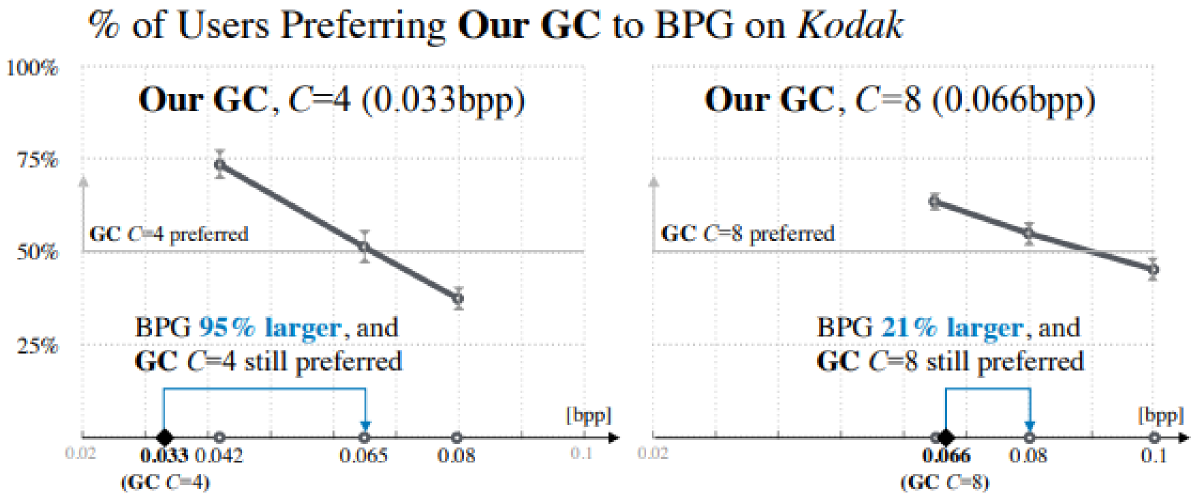
Encoder

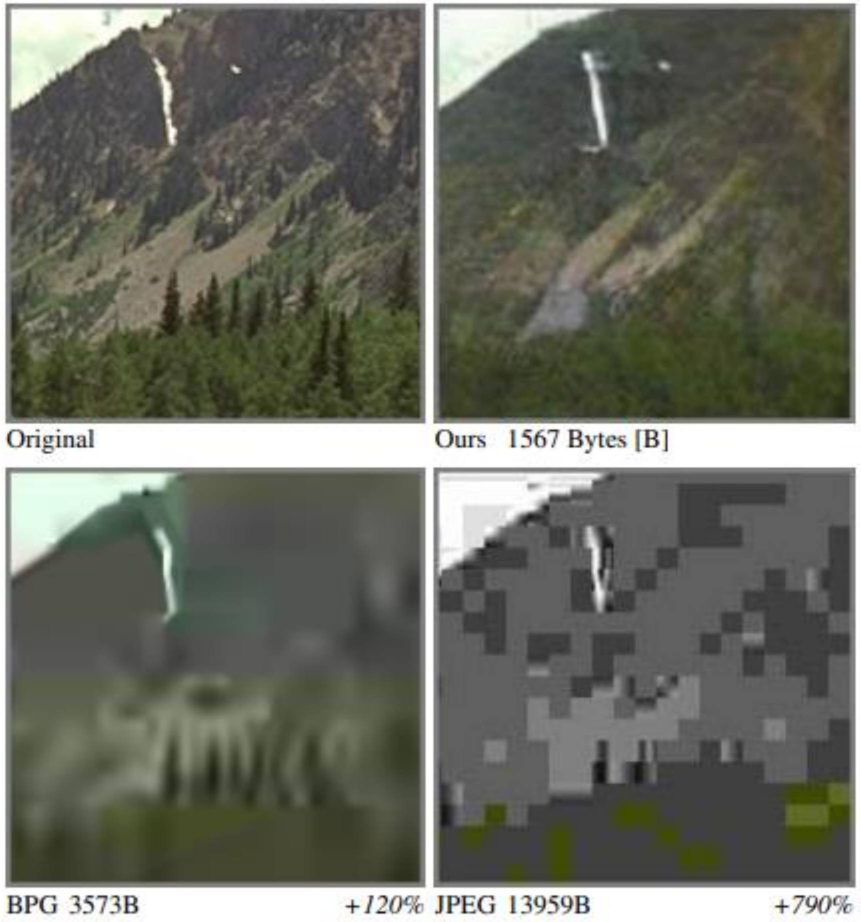
Decoder (Generator)

Discriminator

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

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

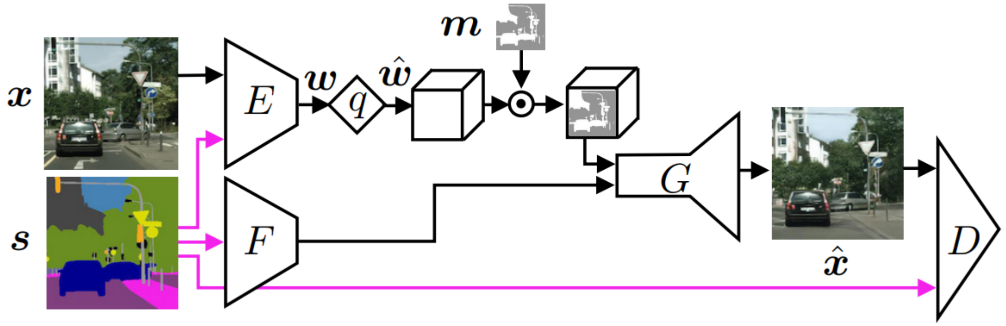




[13] Agustsson, Eirikur, et al., "Generative adversarial networks for extreme learned image compression." in ICCV. 2019.

Learned Image Compression

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



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

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

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

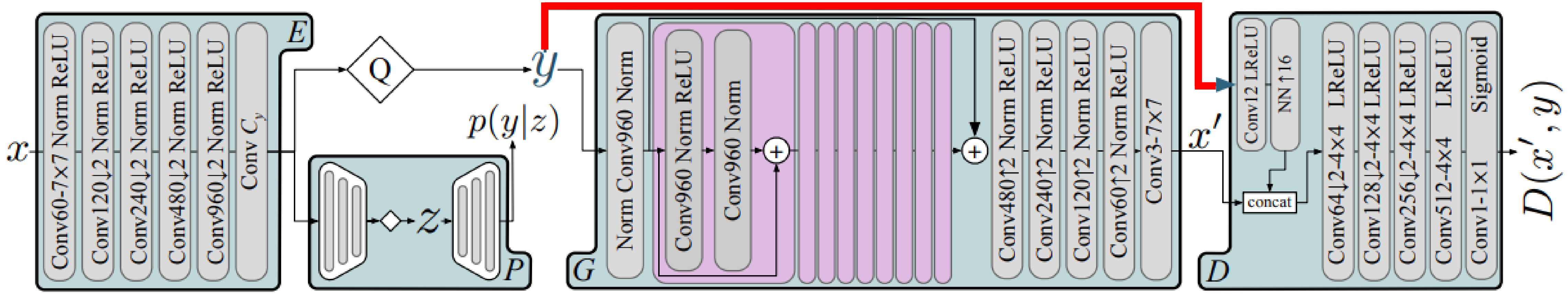
people (0.219bpp, -33%) | *building* (0.199bpp, -39%) | no synth. (0.326bpp, -0%)

[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:

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

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

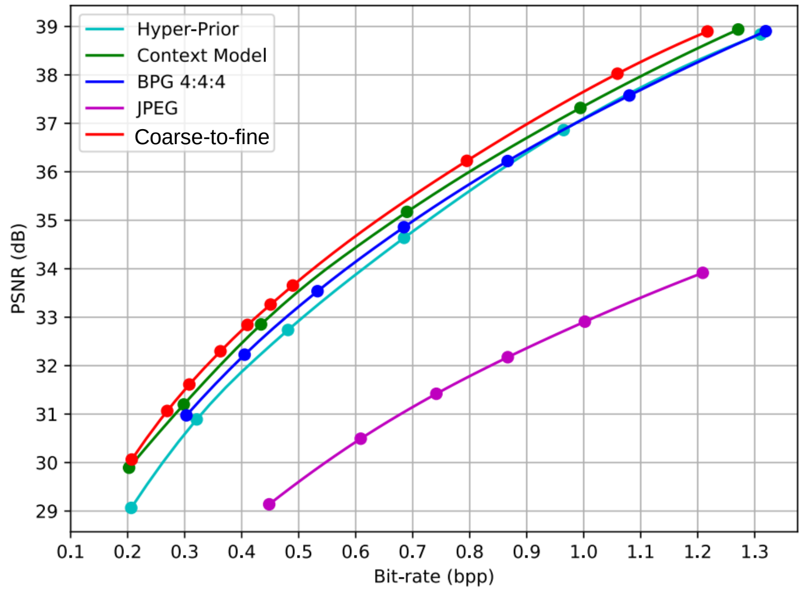
[14] Mentzer, Fabian, et al. "High-Fidelity Generative Image Compression." in NeurIPS. 2020.

Learned Image Compression

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

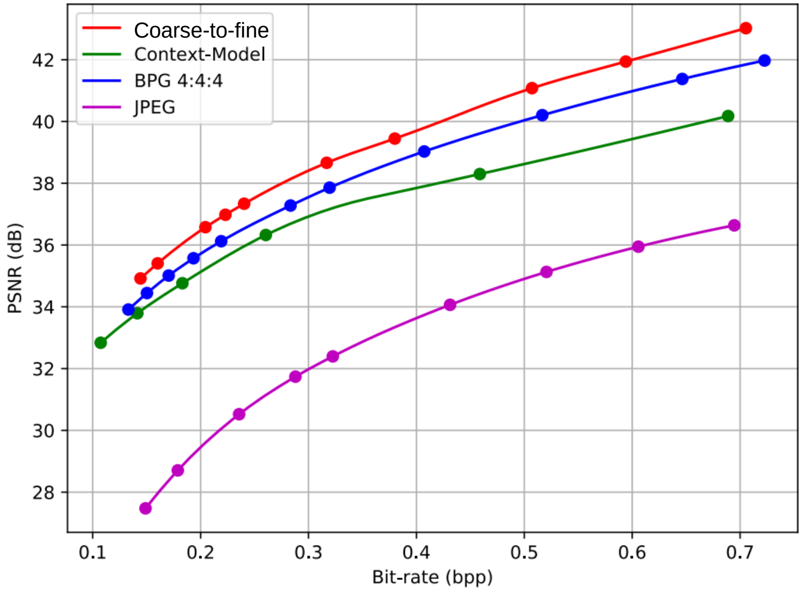
[14] Mentzer, Fabian, et al., "High-Fidelity Generative Image Compression." in NeurIPS. 2020.

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.

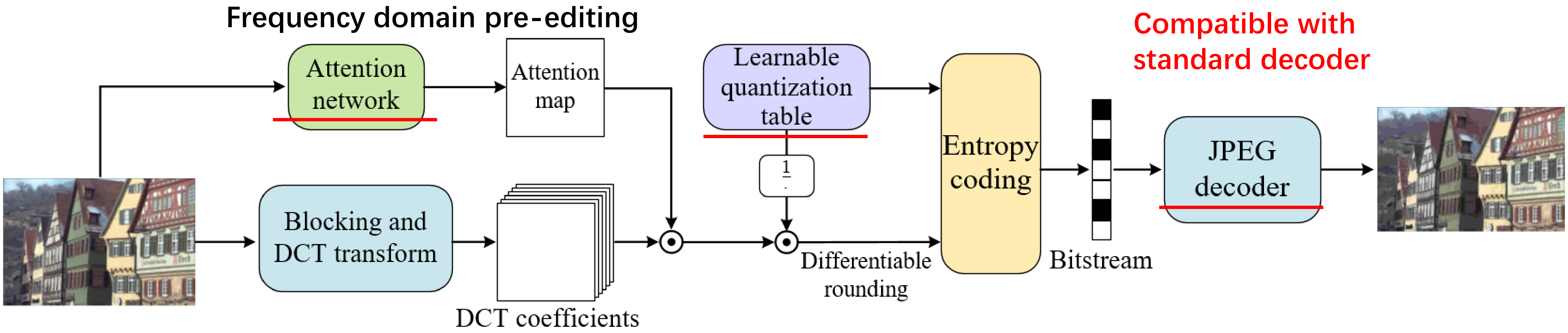
JPEG AI

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]

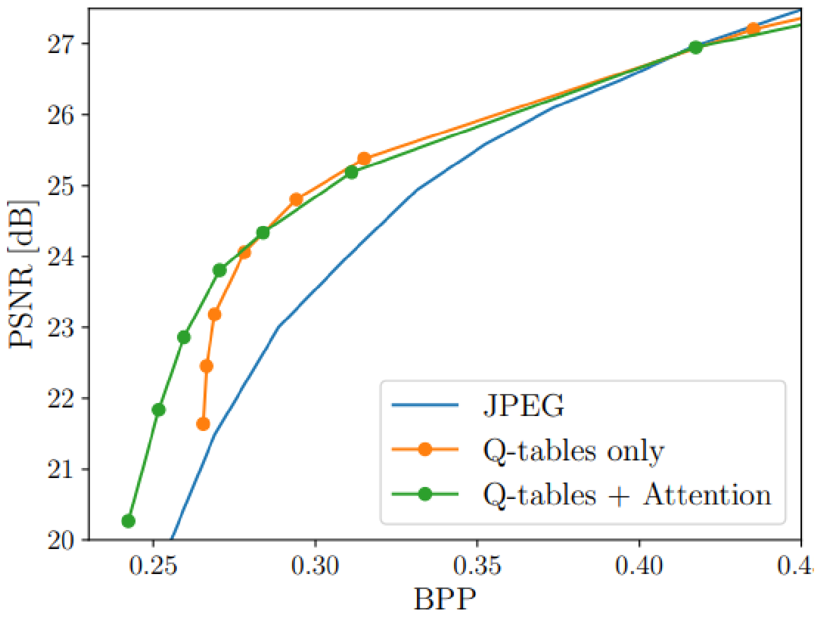


[15] Strümpler, Yannick, et al. "Learning to Improve Image Compression without Changing the Standard Decoder." in ECCVW. 2020.

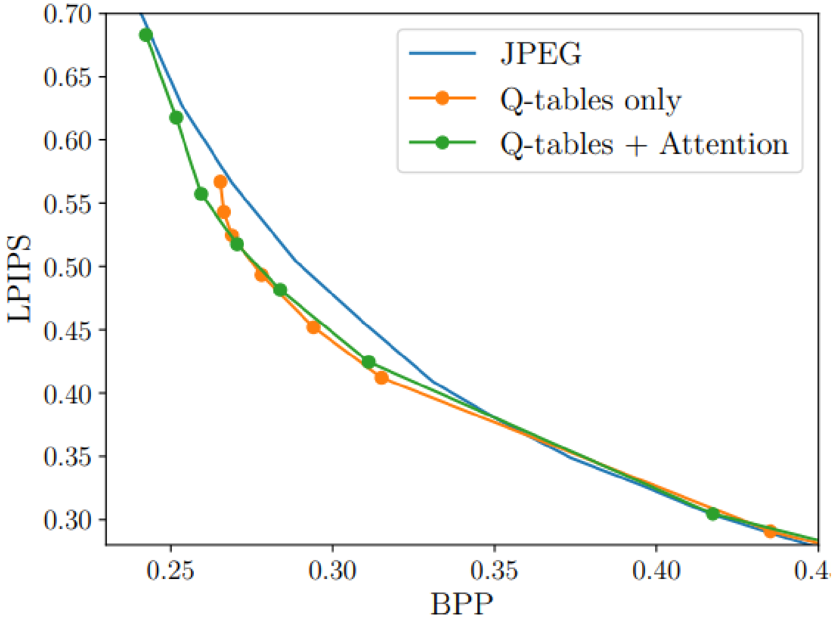
Learned Image Compression

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

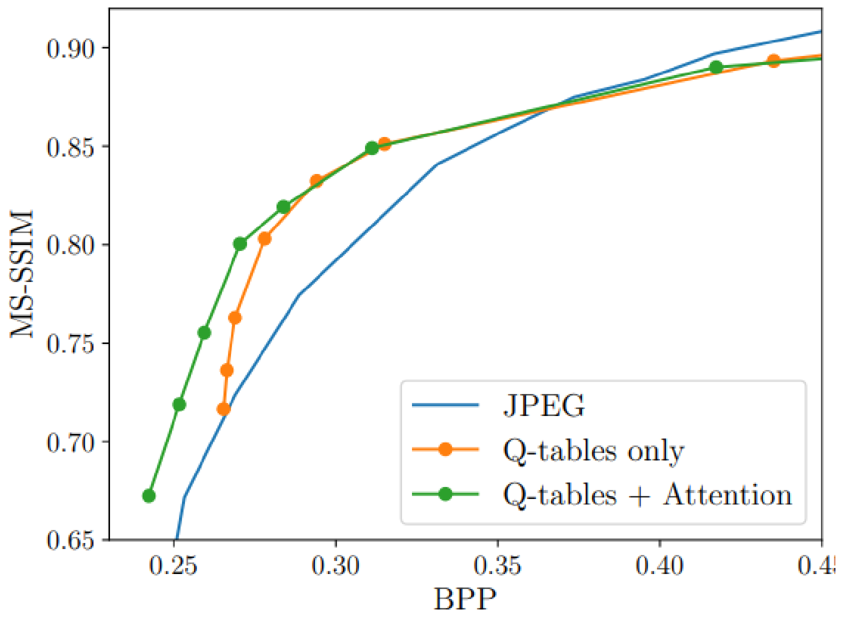
We made an attempt: [15]



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.

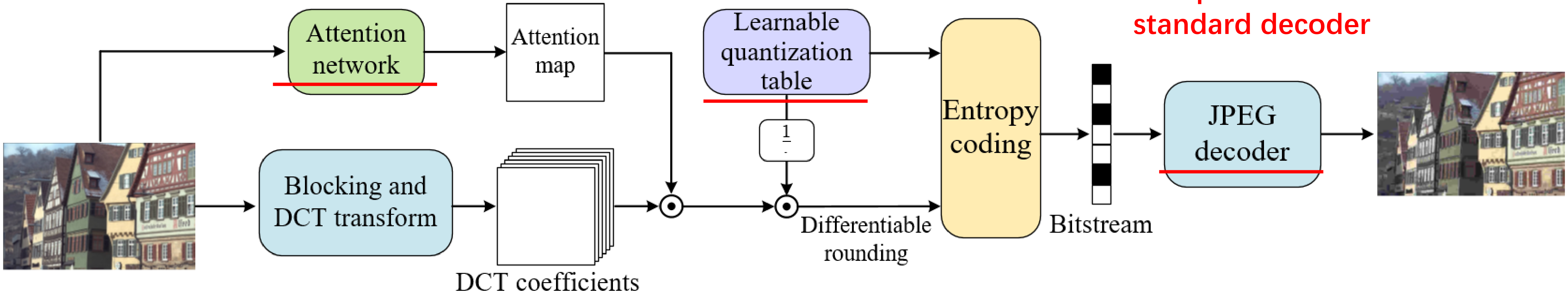
Learned Image Compression

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

We made an attempt: [15]

Frequency domain pre-editing

Compatible with standard decoder



- 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

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

VCIP, December 1-4, 2020

IEEE VCIP 2020

Learned Image and Video Compression with Deep Neural Networks

PART 2: Learned Video Compression

Guo Lu
Assistant Professor
Beijing Institute of Technology, China

Dong Xu
Professor
University of Sydney, Australia

VCIP, December 1-4, 2020

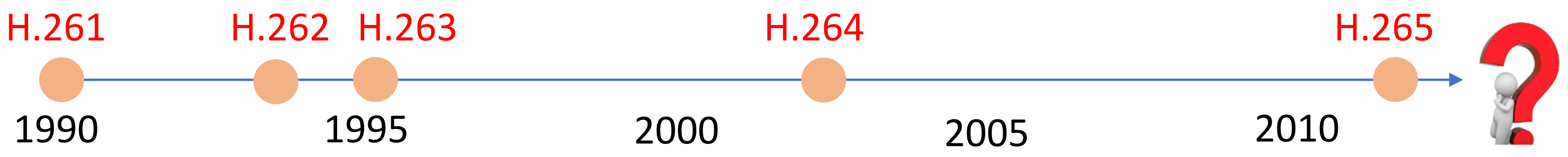
IEEE VCIP 2020

Outline

- Background for Video Compression
- End-to-end Learned P-frame Compression
- End-to-end Learned B-frame Compression
- Learned Autoencoder based Video Compression
- Discussion

Background for Video Compression

Traditional codecs rely on classical prediction-transform architecture and hand-crafted techniques.

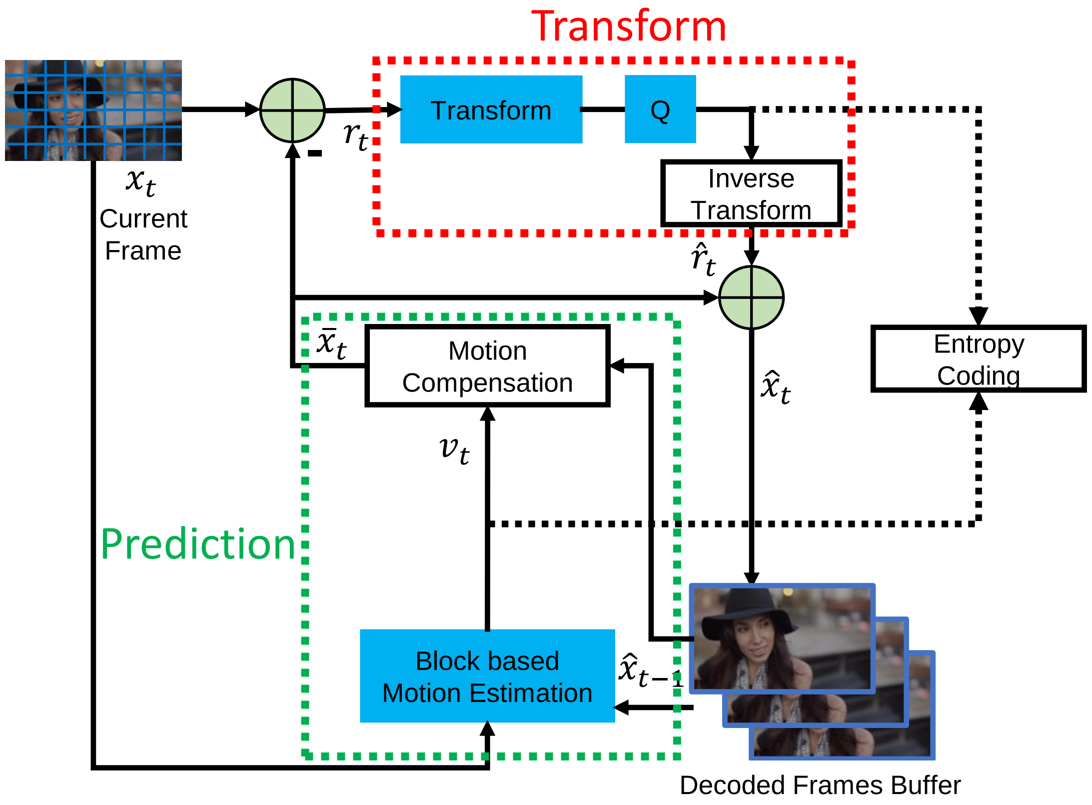


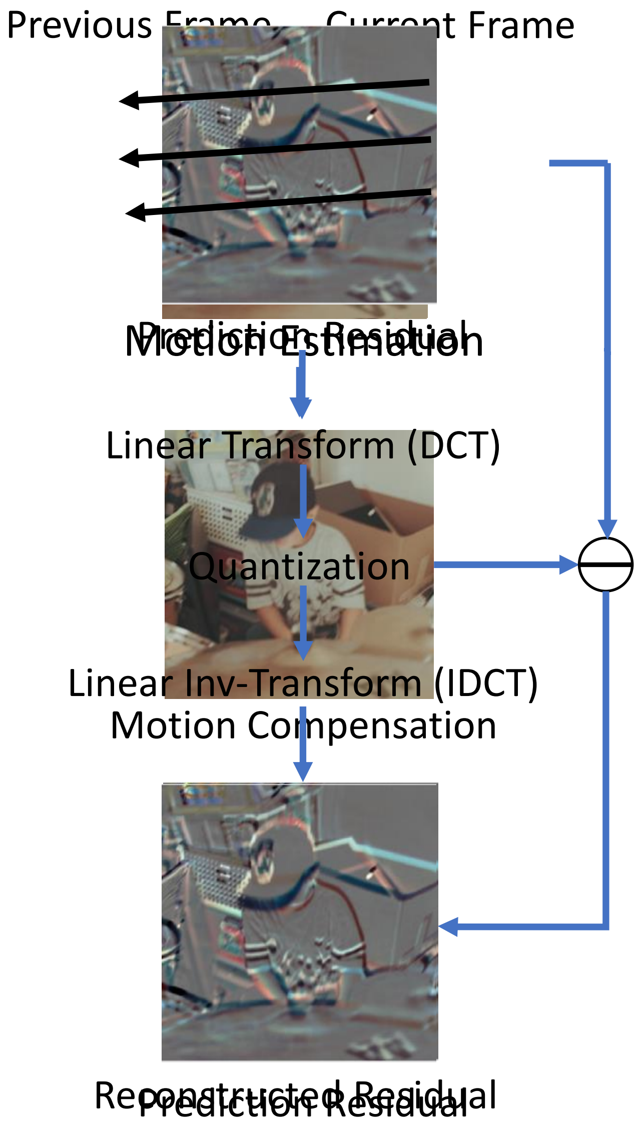
Deep learning has been widely used for a lot of vision tasks for its powerful representation ability.

What happens when video compression meets deep learning?

Background for Video Compression

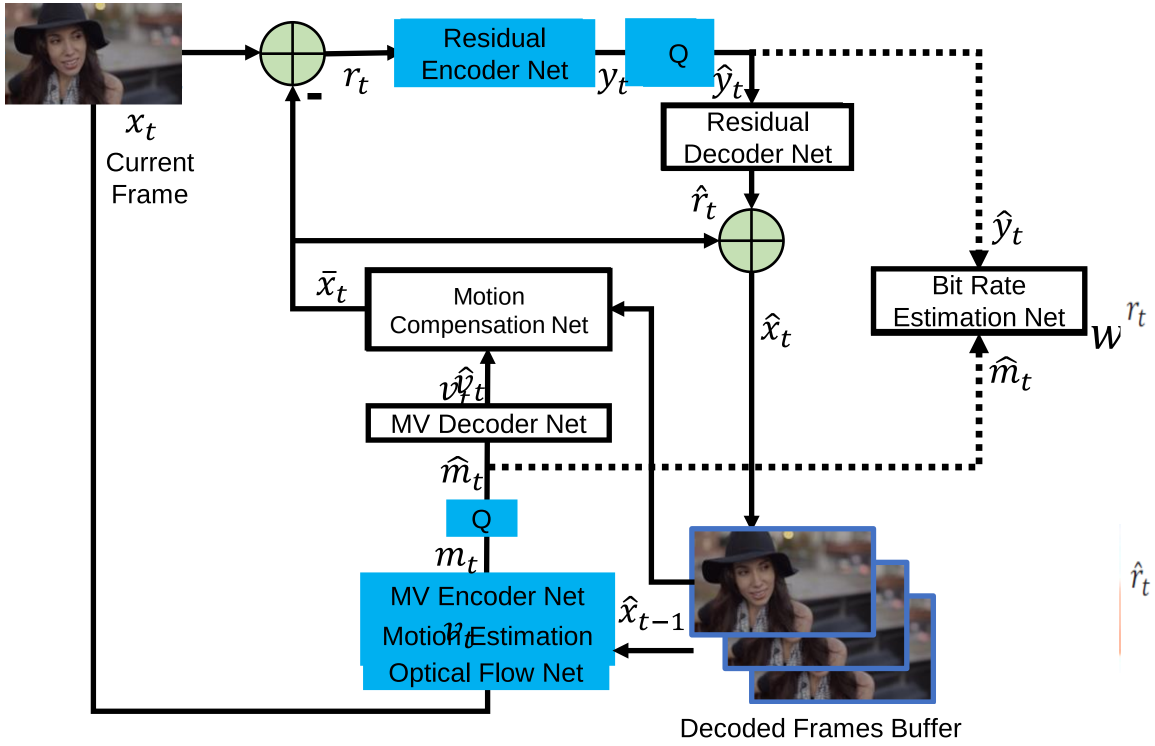
- Traditional Video Compression

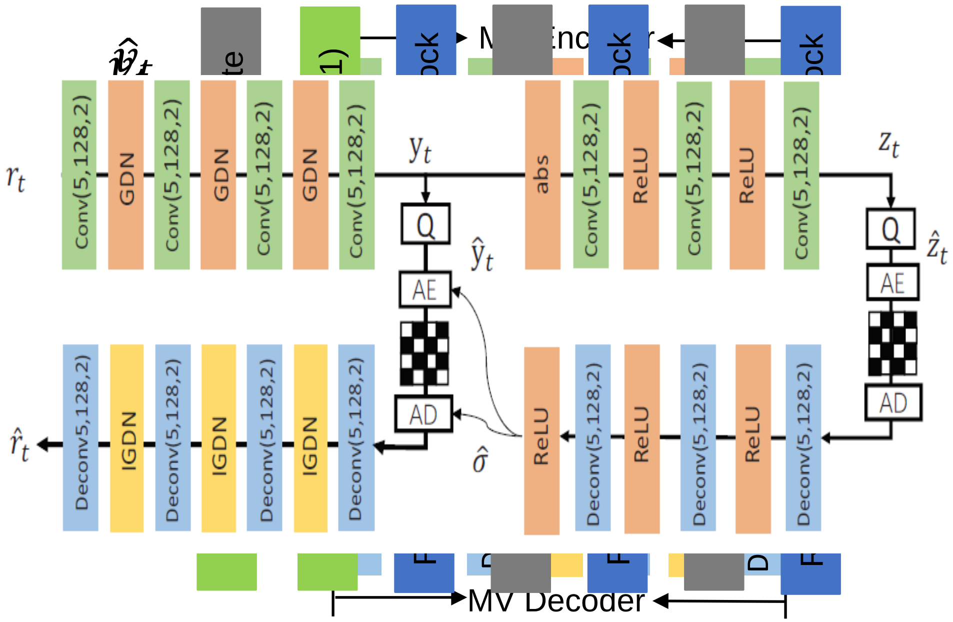




End-to-End Learned P-Frame Video Compression

- The first end-to-end optimized video compression system[1]



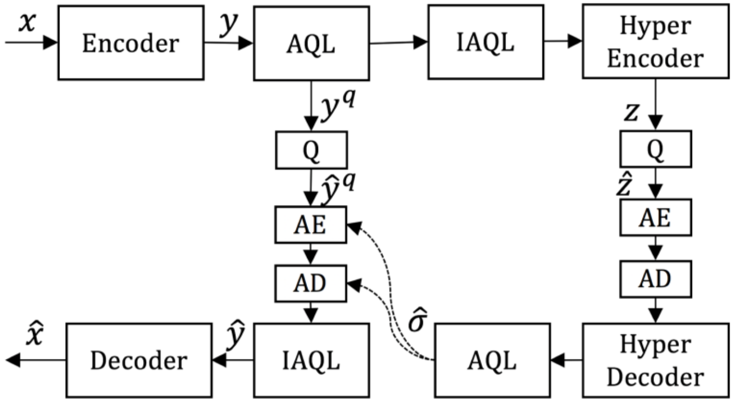


$$\min \lambda D + R$$

[1]Guo Lu, et al. “DVC: An end-to-end deep video compression framework”, in CVPR(Oral), 2019.

End-to-End Learned P-Frame Video Compression

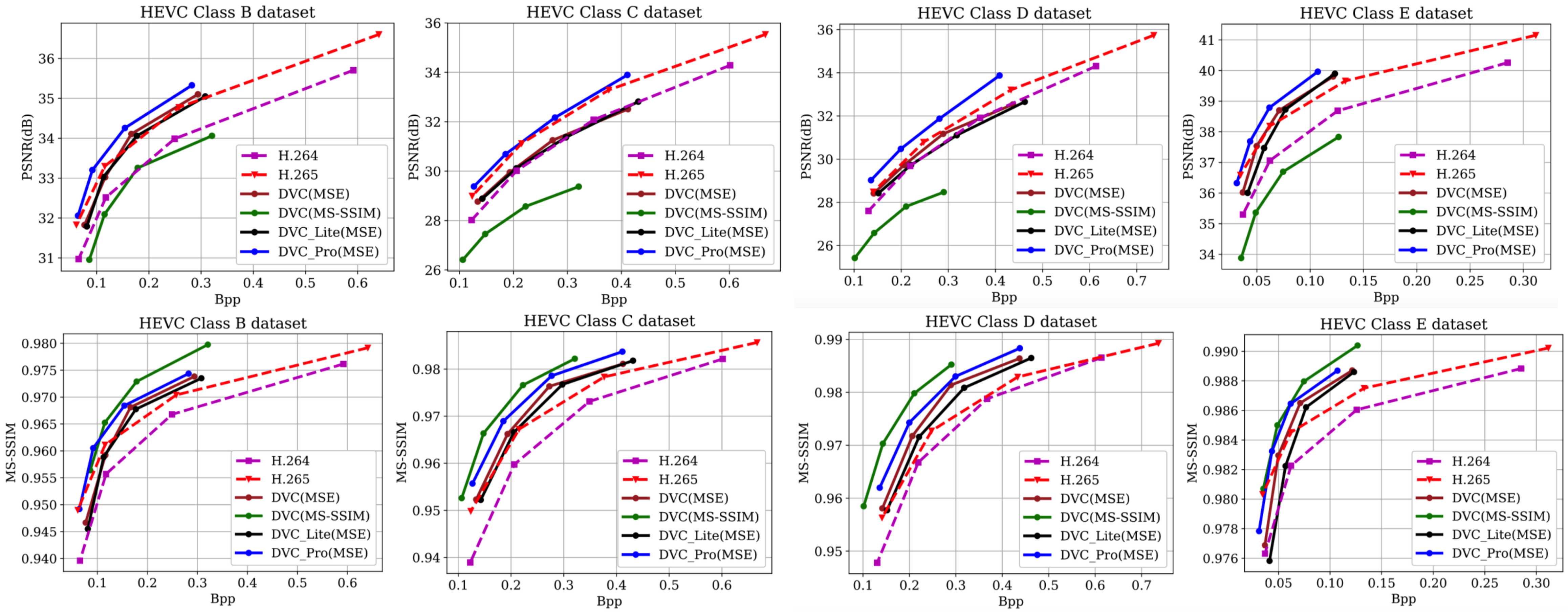
- Flexible Framework[1,2]
  - Advanced entropy model for motion/residual compression with auto-regressive entropy -> reduce bitrate
  - Motion and Residual Refinement -> reduce distortion
- Variable Bitrate Solution
  - Adaptive quantization layer -> feature transformation between different rates



[1]Guo Lu, et al. “DVC: An end-to-end deep video compression framework”, in CVPR(Oral), 2019.
[2]Guo Lu, et al. “An End-to-End Learning Framework for Video Compression,” in T-PAMI. 2020.

End-to-End Learned P-Frame Video Compression

- PSNR results on JCT-VC



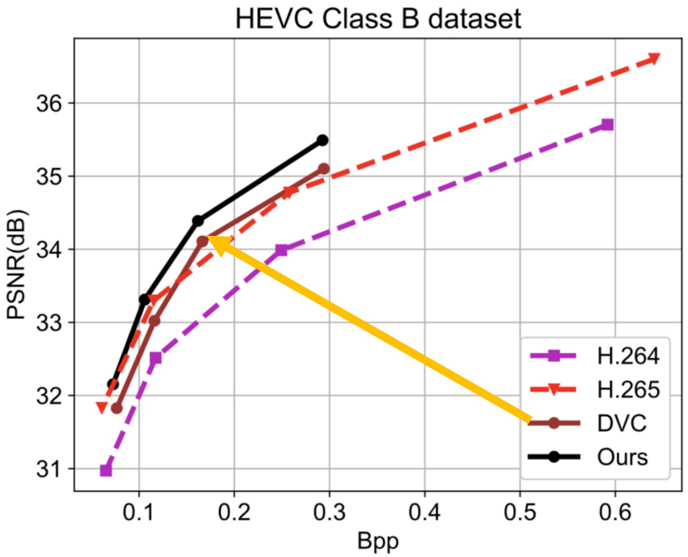
[1]Guo Lu, et al. “DVC: An end-to-end deep video compression framework”, in CVPR(Oral), 2019.
[2]Guo Lu, et al. “An End-to-End Learning Framework for Video Compression,” in T-PAMI. 2020.

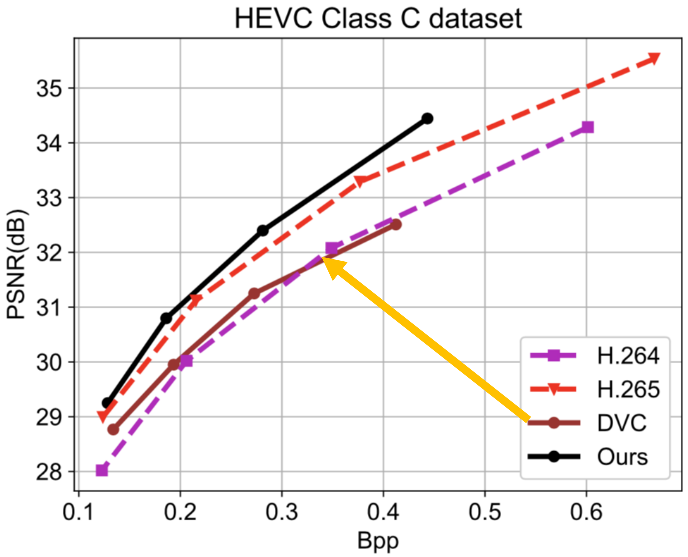
End-to-End Learned P-Frame Video Compression

- RDO is the fundamental technique for video compression
  - Choose the optimal mode in the coding stage, such as motion vector, block size, etc.
- Learning based video compression
  - RDO is ignored in the inference stage
  - the “modes” are fixed after the training stage
- How to apply the RDO technique in learned video compression
  - Directly optimize the encoder
  - Introduce more “modes” and select the optimal

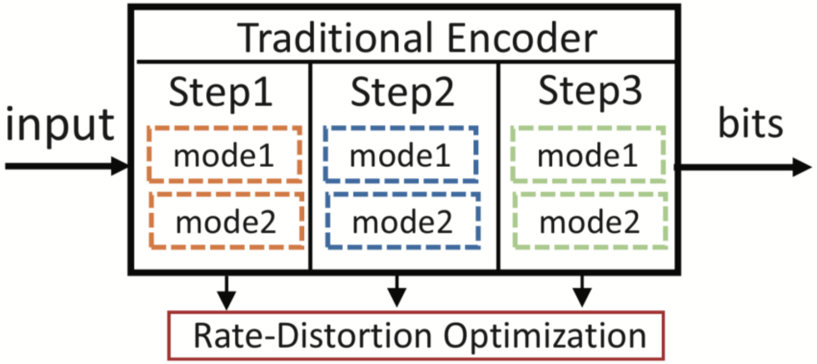
End-to-End Learned P-Frame Video Compression

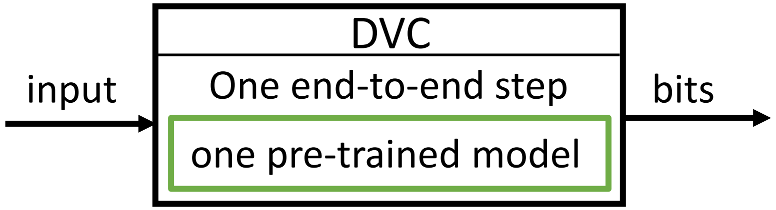
1. Lack of adaptiveness to different video content[3]





DVC, Not as adaptive to different content as traditional methods



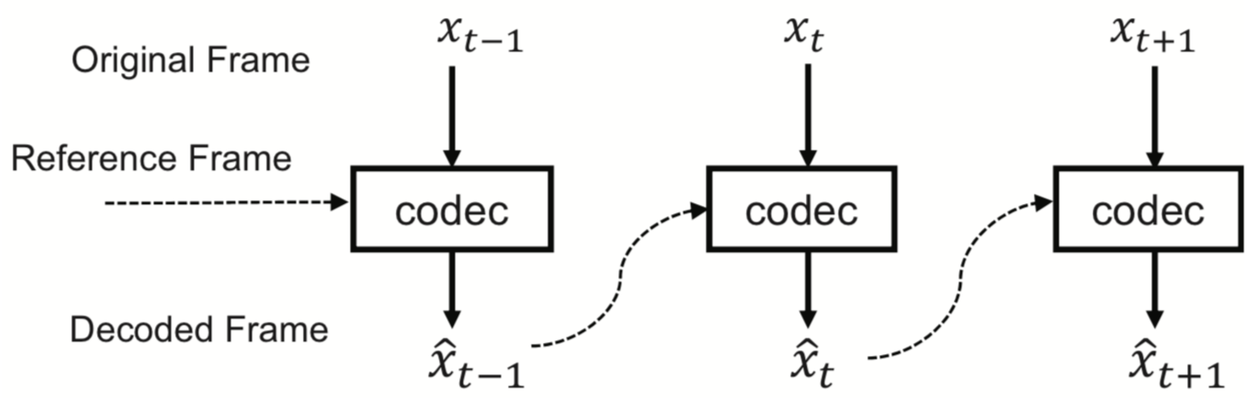


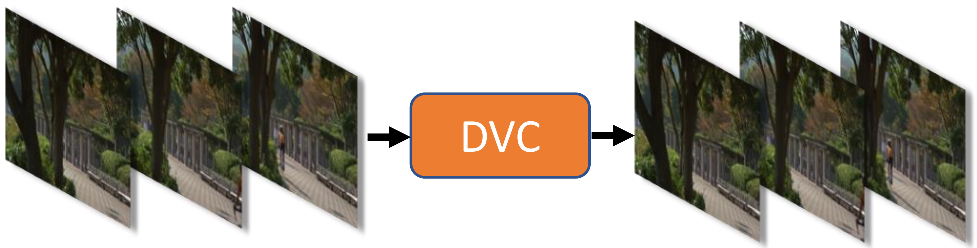
Once the training procedure is finished, **the parameters in the learning based encoder are fixed**

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

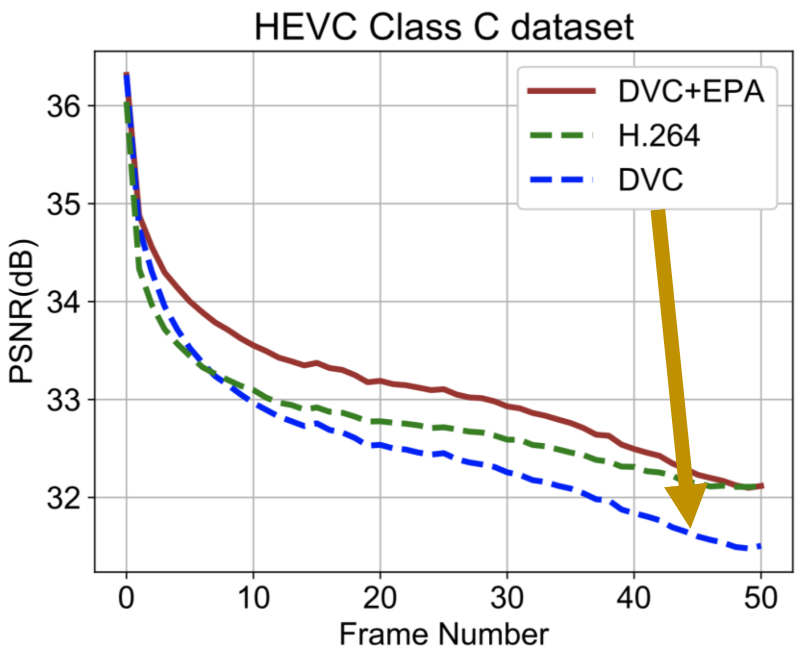
End-to-End Learned P-Frame Video Compression

2. Error propagation in inter predictive coding





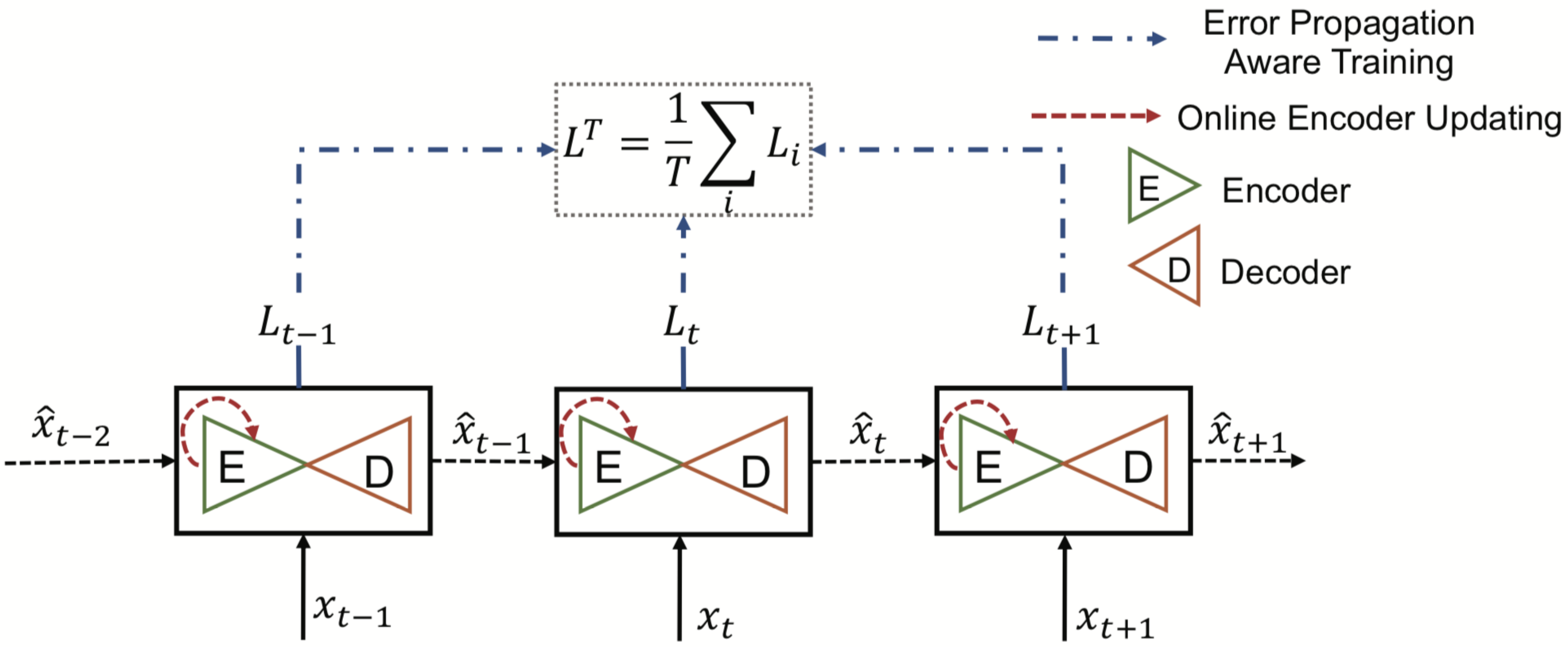
Frame quality keeps decreasing



Quality keeps decreasing because **the error propagation is not considered in the training procedure**

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

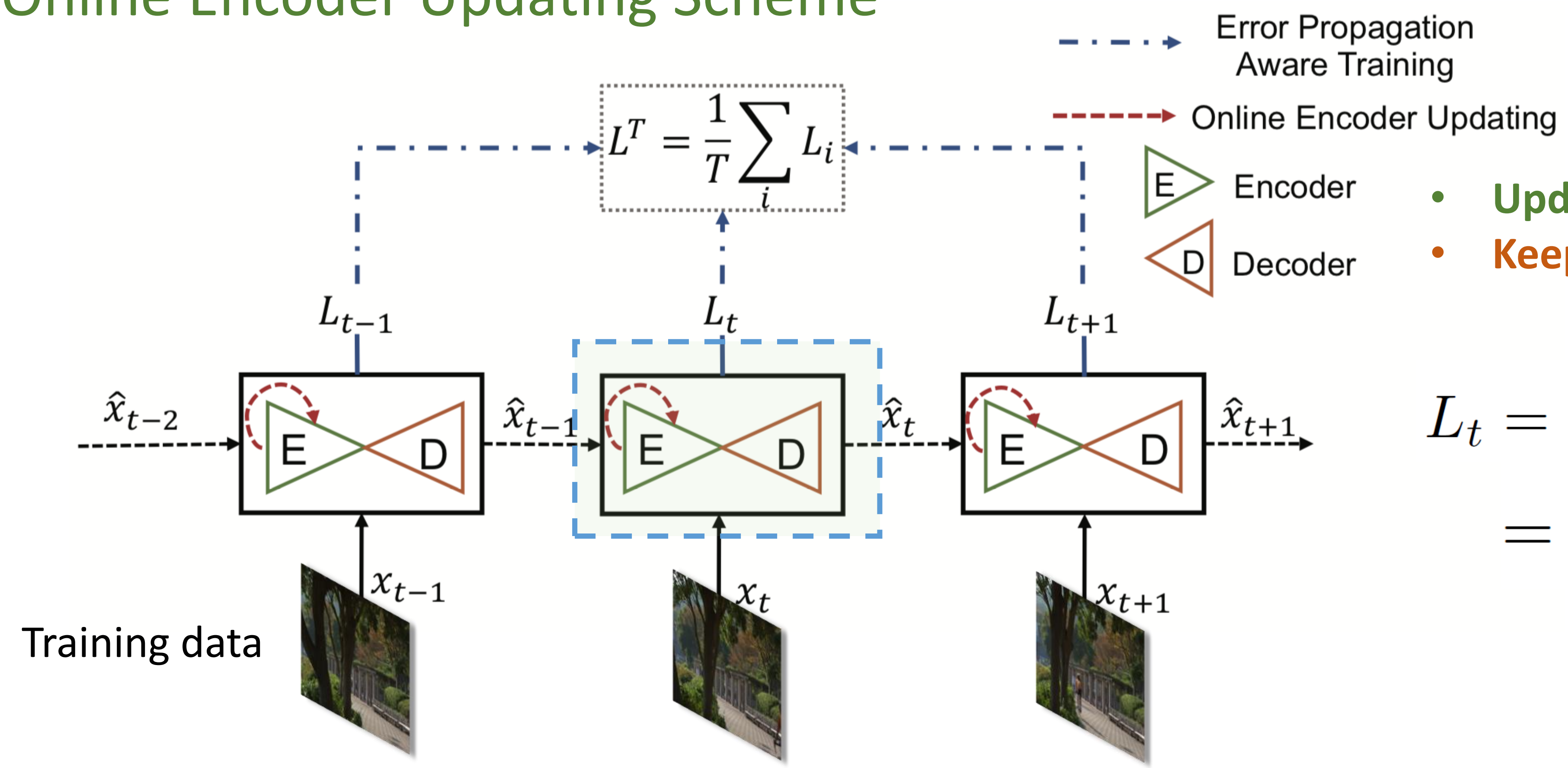


1. Online Encoder Updating Scheme
2. Error Propagation Aware Training Strategy

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

1. Online Encoder Updating Scheme



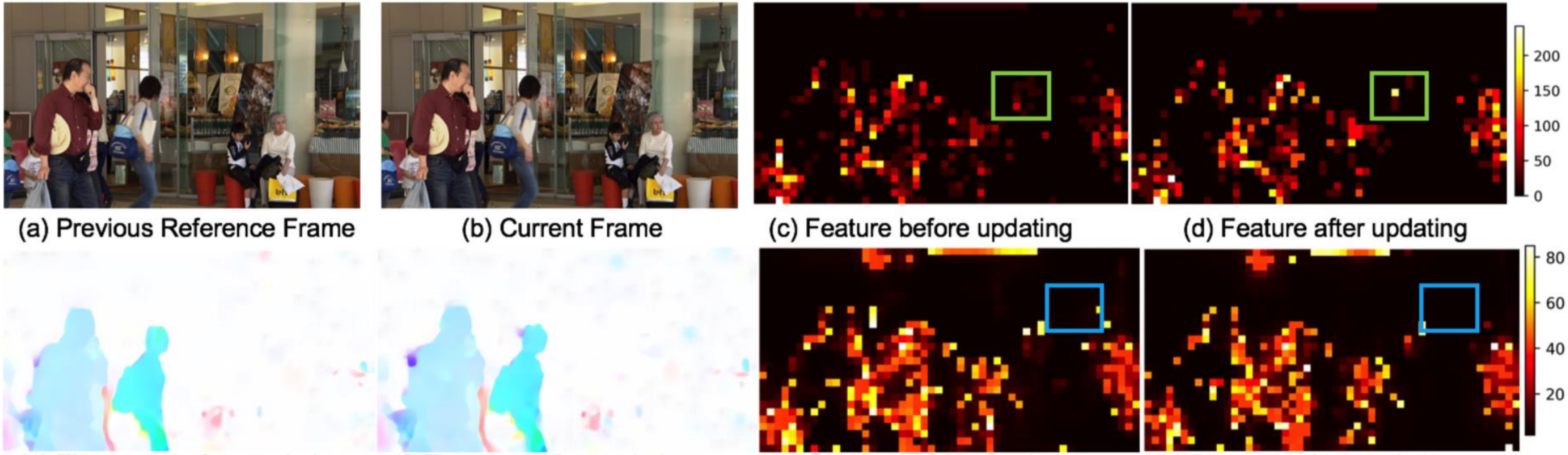
- **Update encoder online**
- **Keep the decoder unchanged**

$$
\begin{aligned}
L_t &= \lambda D_t + R_t \\
&= \lambda d(x_t, \hat{x}_t) \\
&\quad + [H(\hat{y}_t) + H(\hat{m}_t)]
\end{aligned}
$$

[3]Guo Lu, Chunlei Cai, et al. “Content Adaptive and Error Propagation Aware Deep Video Compression”, in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

1. Online Encoder Updating Scheme



(a) Previous Reference Frame (b) Current Frame (c) Feature before updating (d) Feature after updating

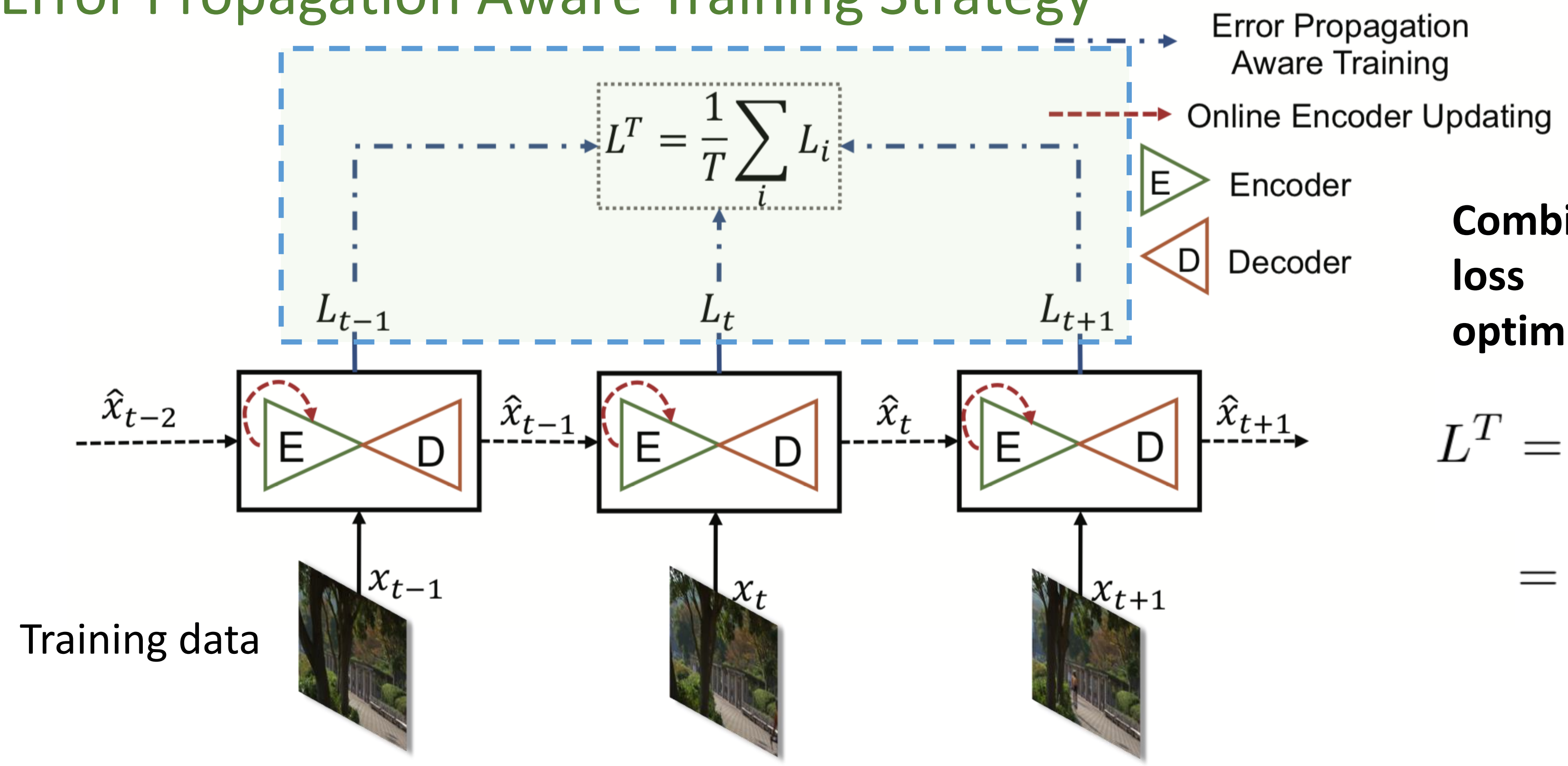
(e) Flow map before updating (f) Flow map after updating (g) Bits map before updating (h) Bits map after updating

Improve adaptiveness with **Same decoding time**

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

2. Error Propagation Aware Training Strategy



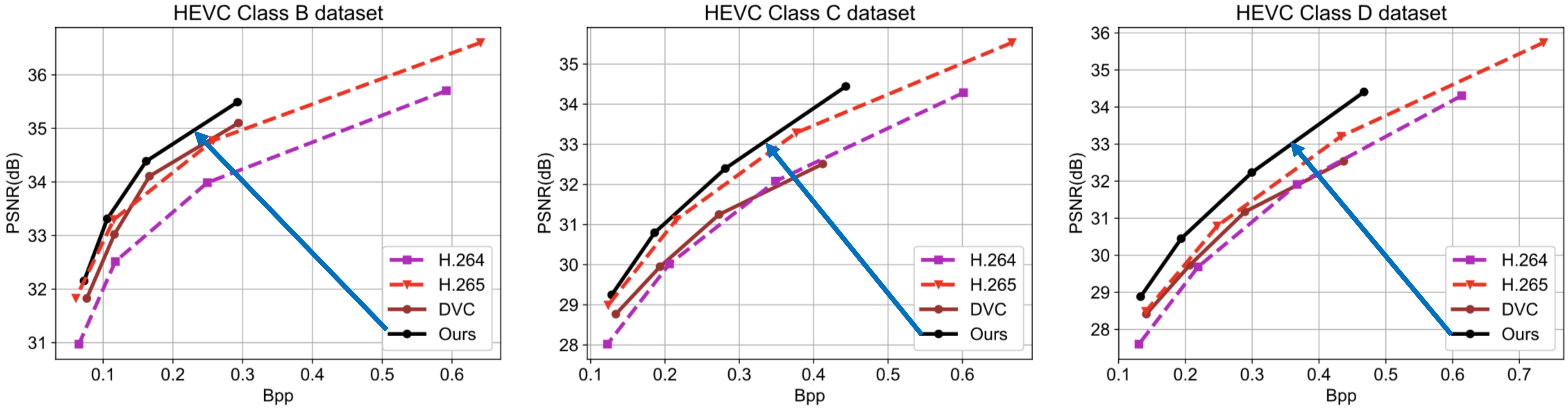
Combine the rate-distortion loss of several frames to optimize the learned codec

$$L^T = \frac{1}{T}\sum_t L_t = \frac{1}{T}\sum_t \{\lambda d(x_t, \hat{x}_t) + [H(\hat{y}_t) + H(\hat{m}_t)]\}$$

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

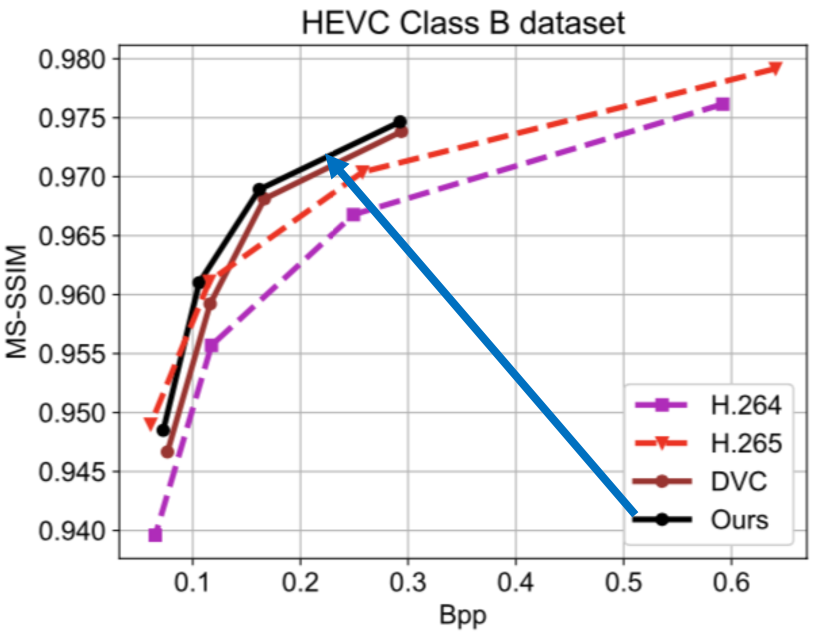
All methods use same GoP settings

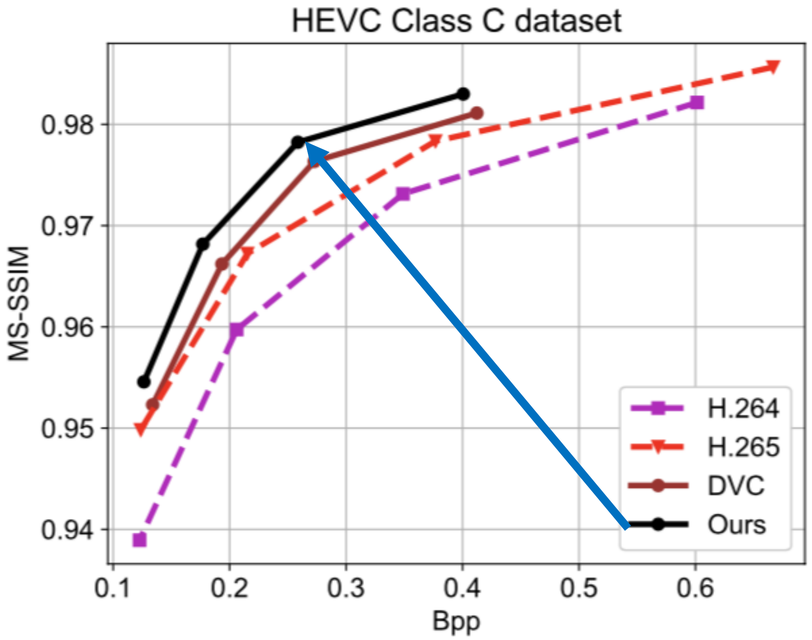


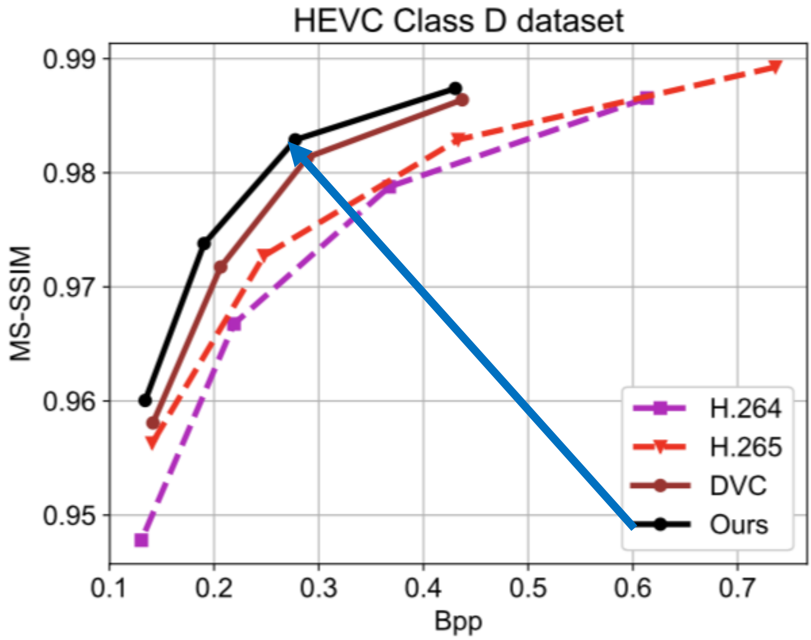
[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

All methods use same GoP settings







[3]Guo Lu, Chunlei Cai, et al. “Content Adaptive and Error Propagation Aware Deep Video Compression”, in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

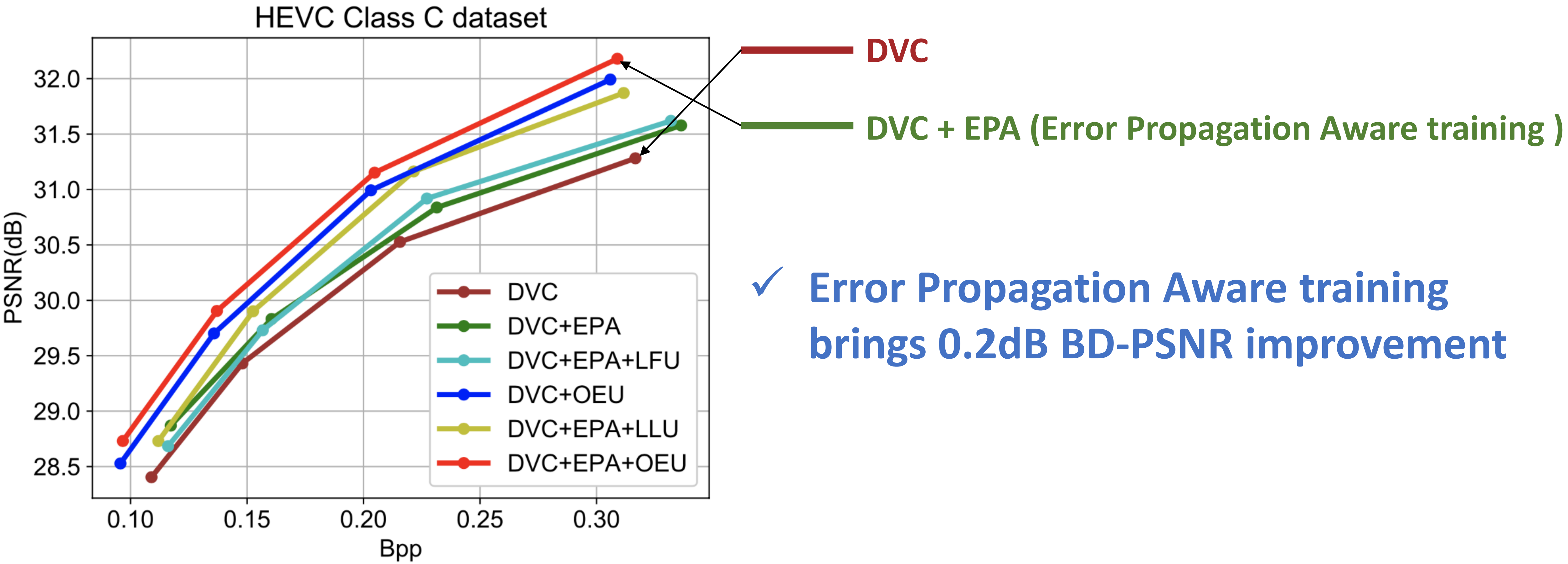
All methods use same GoP settings

Table 1: The BDBR and BD-PSNR results of different algorithms when compared with H.264. Negative values in BDBR represent the bitrate saving.

	BDBR(%)			BD-PSNR(dB)		
Dataset	H.265	DVC	Ours	H.265	DVC	Ours
Class B	-32.0	-27.9	-41.7	0.78	0.71	1.12
Class C	-20.8	-3.5	-25.9	0.91	0.13	1.18
Class D	-12.3	-6.2	-25.1	0.57	0.26	1.25

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

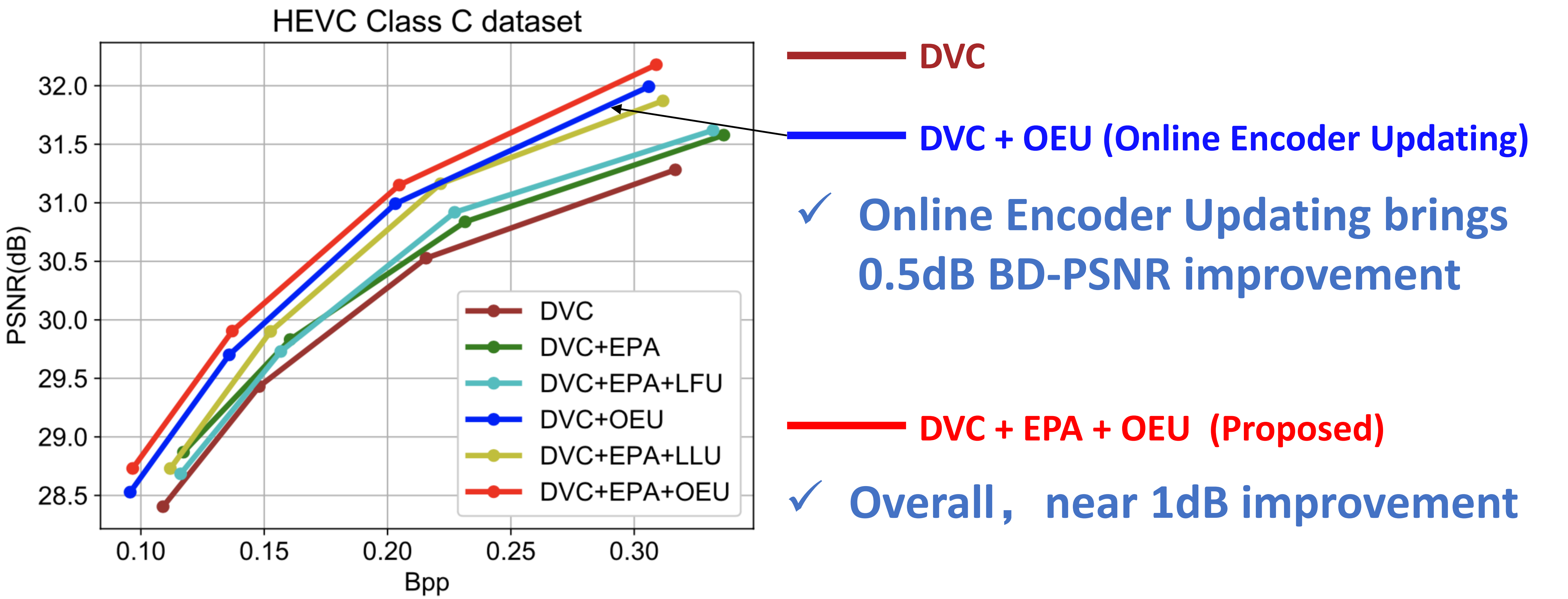
End-to-End Learned P-Frame Video Compression



✓ **Error Propagation Aware training brings 0.2dB BD-PSNR improvement**

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression



✓ **Online Encoder Updating brings 0.5dB BD-PSNR improvement**

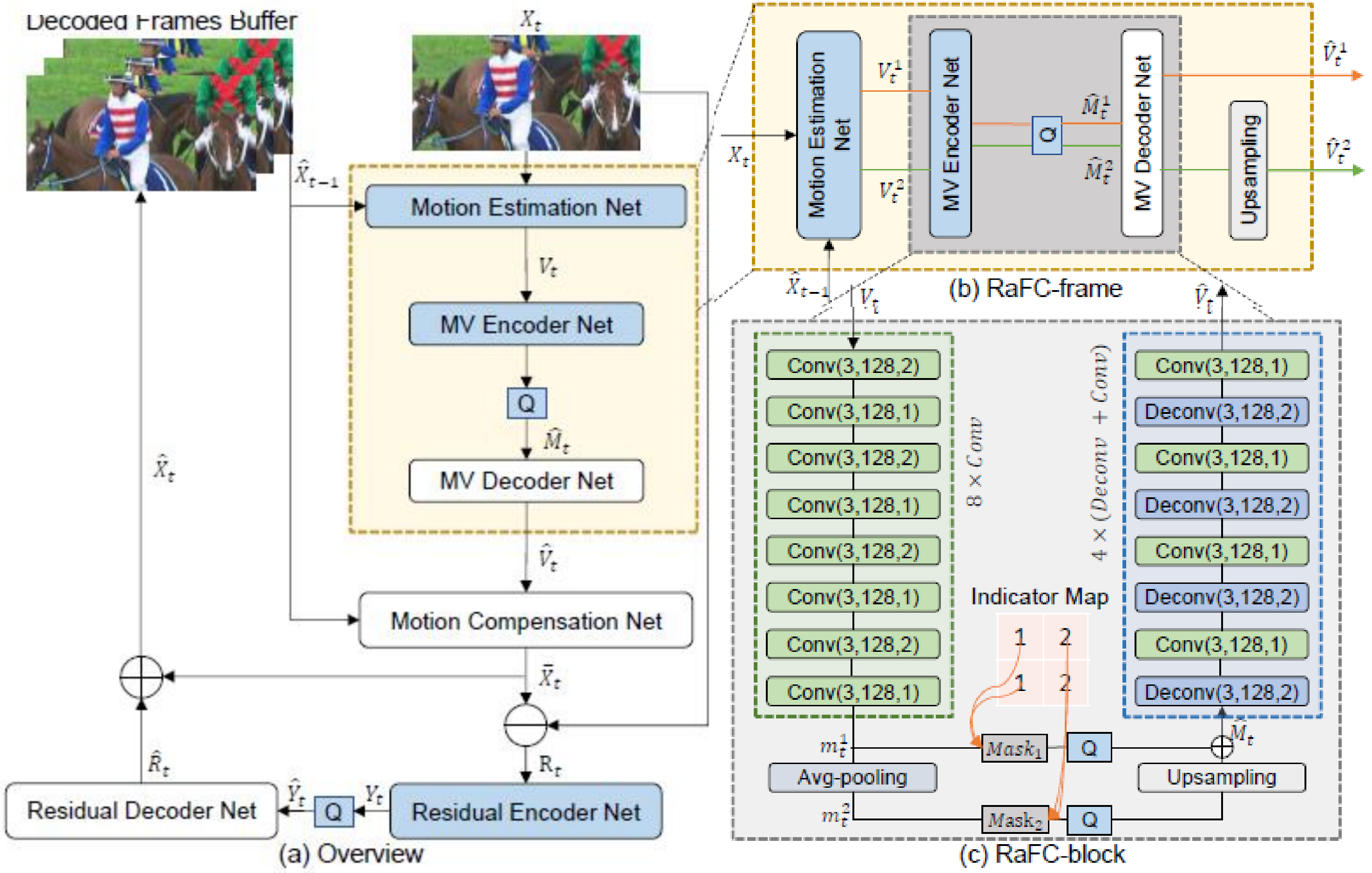
✓ **Overall， near 1dB improvement**

[3]Guo Lu, Chunlei Cai, et al. "Content Adaptive and Error Propagation Aware Deep Video Compression", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

- Various block sizes are adopted in traditional video compression
  - Large block size for smooth region
  - Small block size for complex region
- Fixed optical flow resolution and motion representation resolutions are used in existing work, like DVC.
  - Generates more mode -> flow resolutions or representation resolutions
  - Choose the optimal mode using RDO

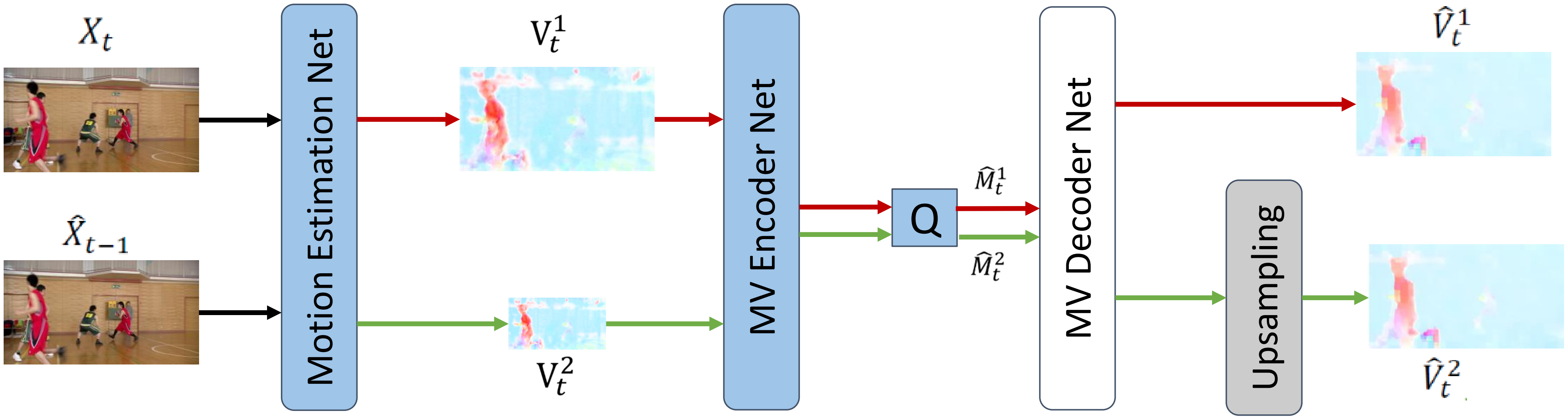
End-to-End Learned P-Frame Video Compression



- (a) Overview of the proposed video compression system.
- (b) RaFC-Frame: decides the **Global Optimal Flow Map** resolution for each video frame.
- (c) RaFC-Block: select the optimal resolution for each **Local Block** of motion feature

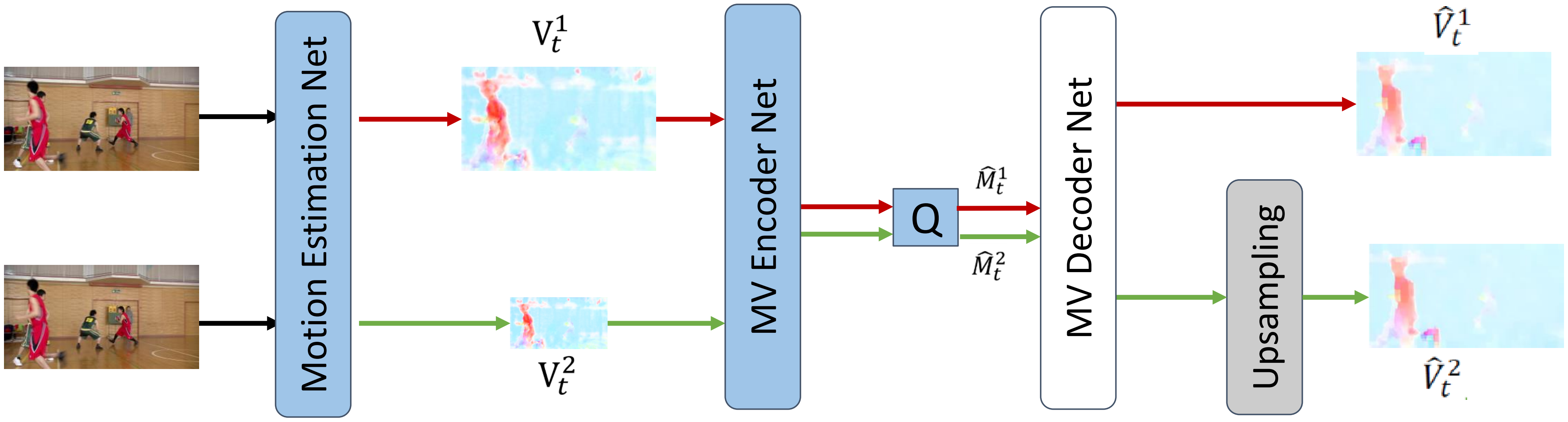
[4] Zhihao Hu, et al. “Improving Deep Video Compression by Resolution-adaptive Flow Coding”, in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression



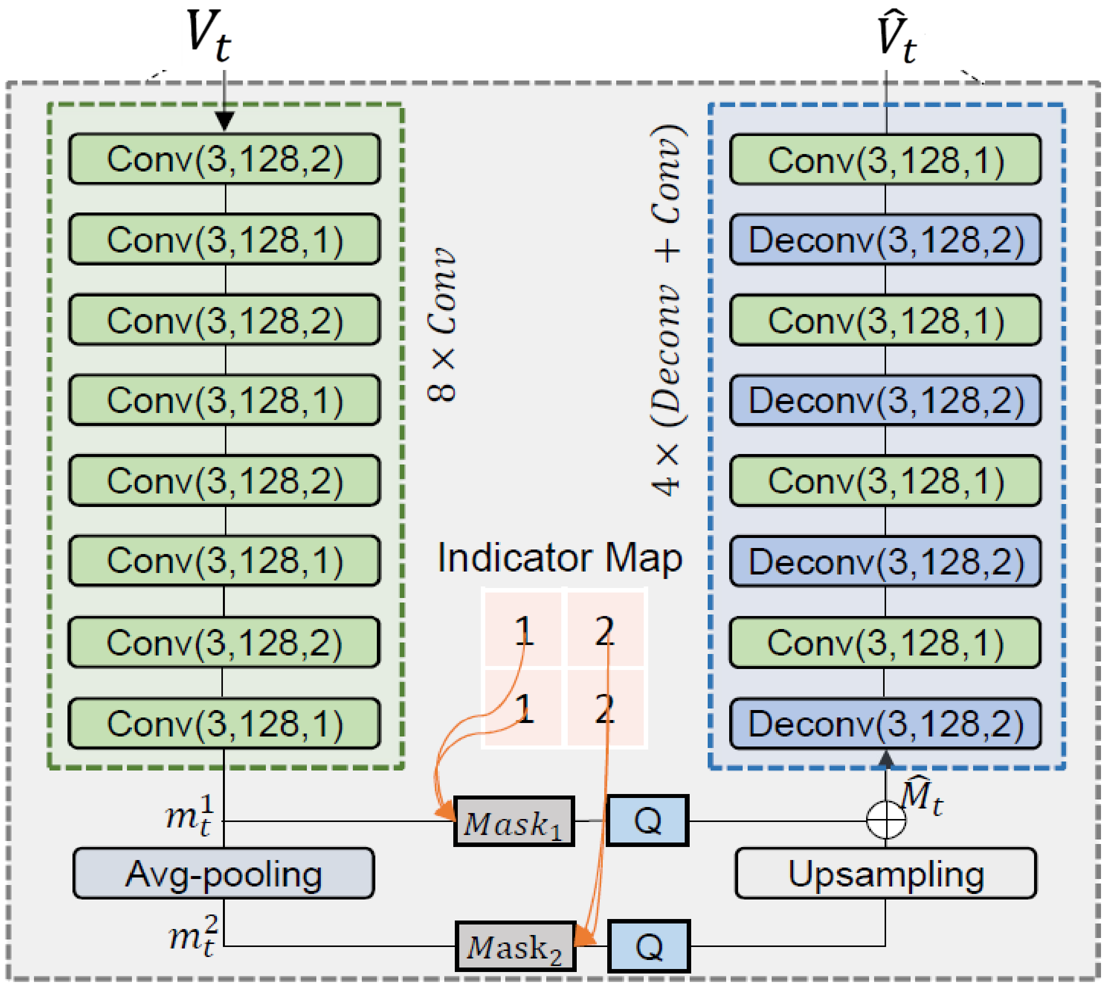
[4] Zhihao Hu, et al. "Improving Deep Video Compression by Resolution-adaptive Flow Coding", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression



[4] Zhihao Hu, et al. “Improving Deep Video Compression by Resolution-adaptive Flow Coding”, in ECCV(Oral), 2020.

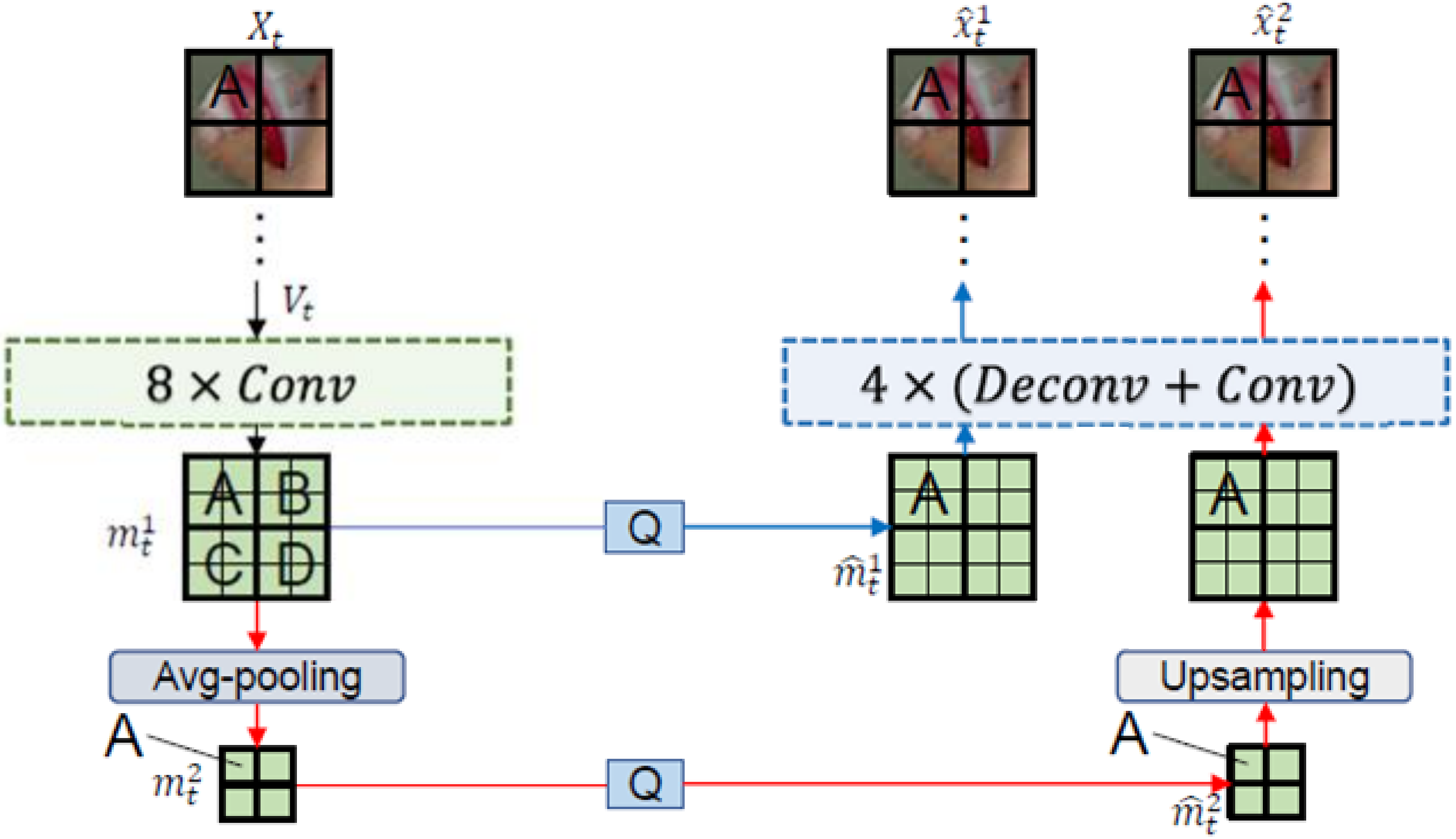
End-to-End Learned P-Frame Video Compression



- Generate multi-scale motion features
- Select the optimal resolution of the motion features for each block

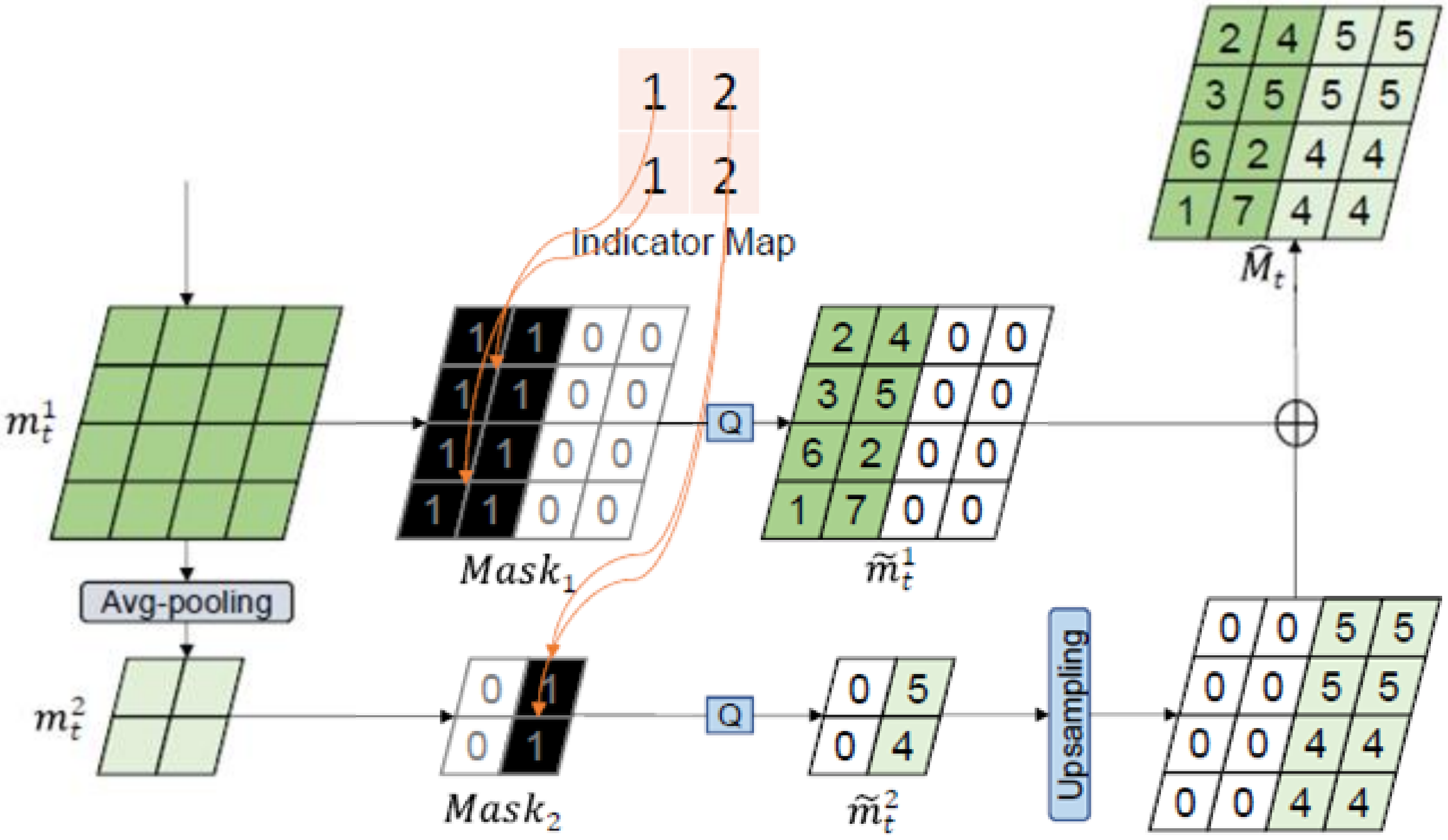
[4] Zhihao Hu, et al. "Improving Deep Video Compression by Resolution-adaptive Flow Coding", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression



[4] Zhihao Hu, et al. "Improving Deep Video Compression by Resolution-adaptive Flow Coding", in ECCV(Oral), 2020.

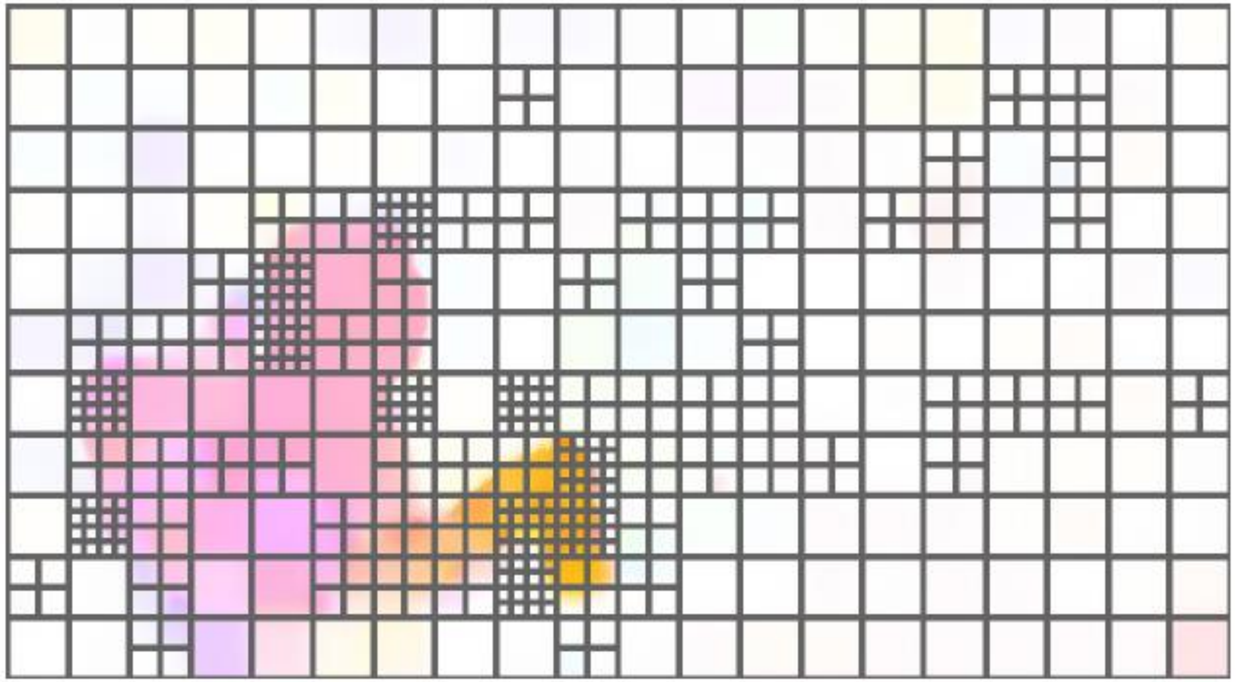
End-to-End Learned P-Frame Video Compression



[4] Zhihao Hu, et al. "Improving Deep Video Compression by Resolution-adaptive Flow Coding", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

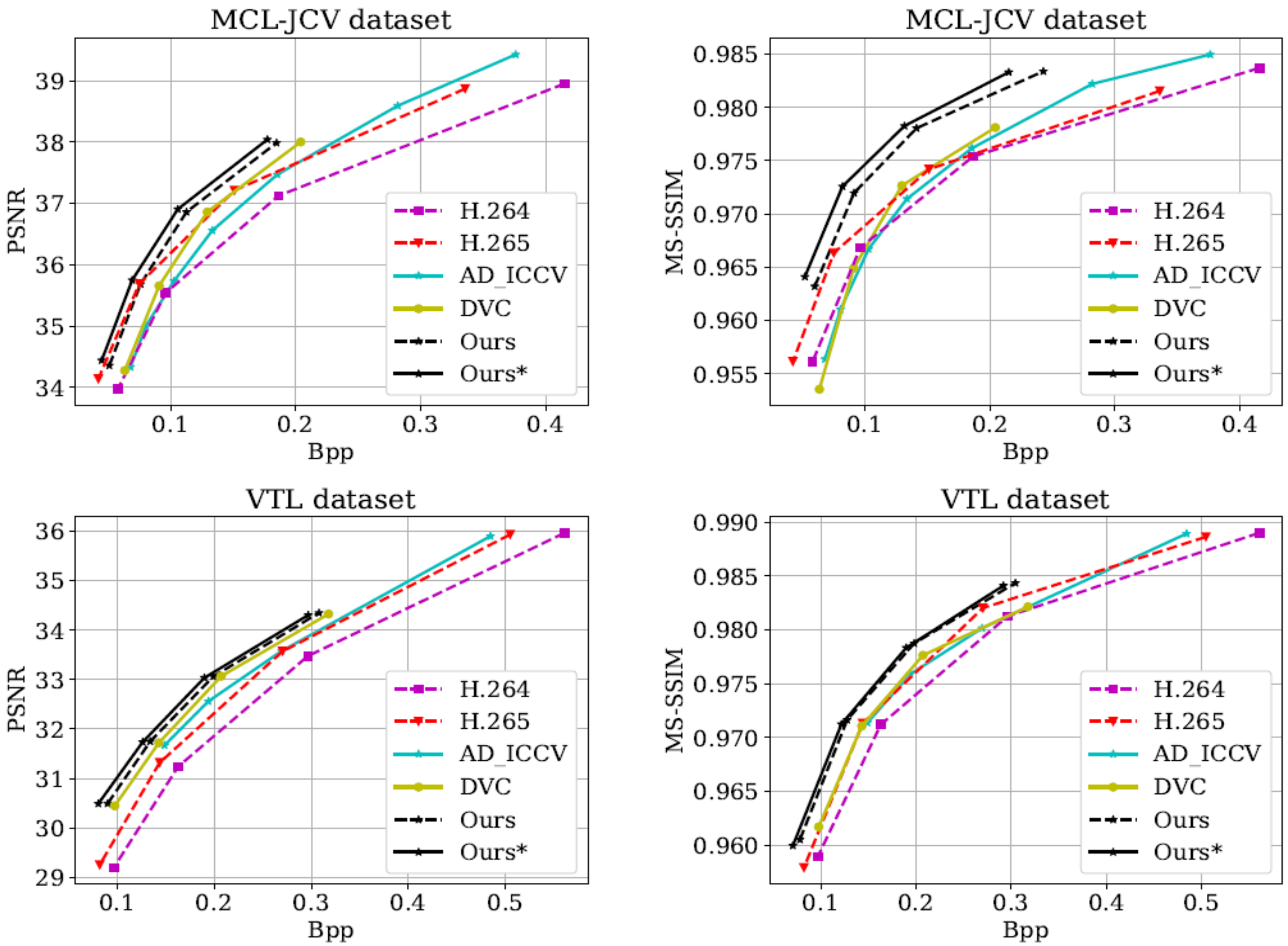
(a)



(b)

[4] Zhihao Hu, et al. "Improving Deep Video Compression by Resolution-adaptive Flow Coding", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

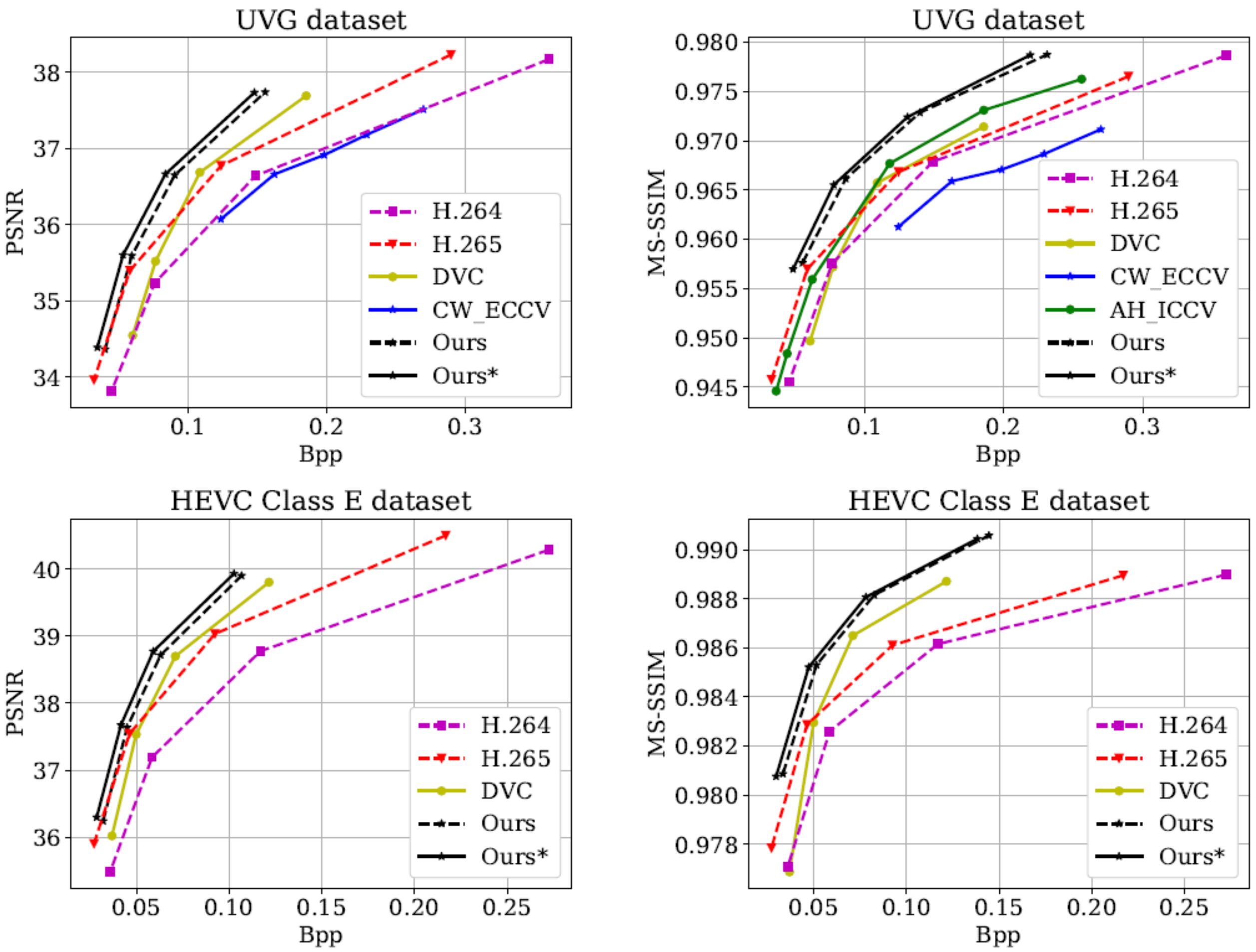


MCL-JCV:
BDBR:
Ours: -35.0% vs. DVC:-14.5%

VTL:
BDBR:
Ours: -30.0% vs. DVC:-21.9%

[4] Zhihao Hu, et al. “Improving Deep Video Compression by Resolution-adaptive Flow Coding”, in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression



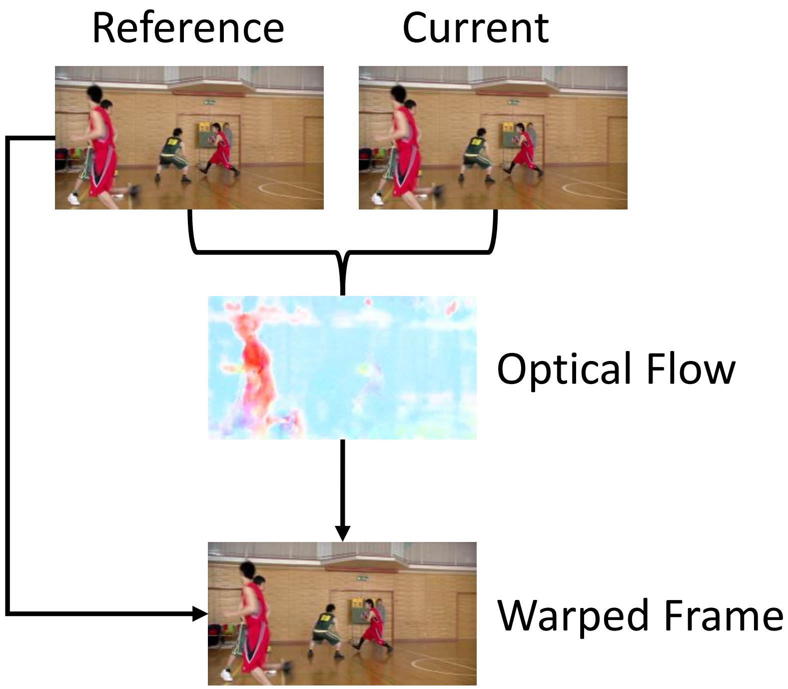
UVG:
BDBR:
Ours: -35.7% vs. DVC:-19.4%

Class E:
BDBR:
Ours: -50.3% vs. DVC:-34.8%

[4] Zhihao Hu, et al. "Improving Deep Video Compression by Resolution-adaptive Flow Coding", in ECCV(Oral), 2020.

End-to-End Learned P-Frame Video Compression

Traditional Motion Compensation Procedure



Formulations

$$\mathbf{x}' := \text{Bilinear-Warp}(\mathbf{x}, \mathbf{f}) \qquad \text{s.t.}$$
$$\mathbf{x}'[x, y] = \mathbf{x}[x + \mathbf{f}_x[x, y], y + \mathbf{f}_y[x, y]]$$

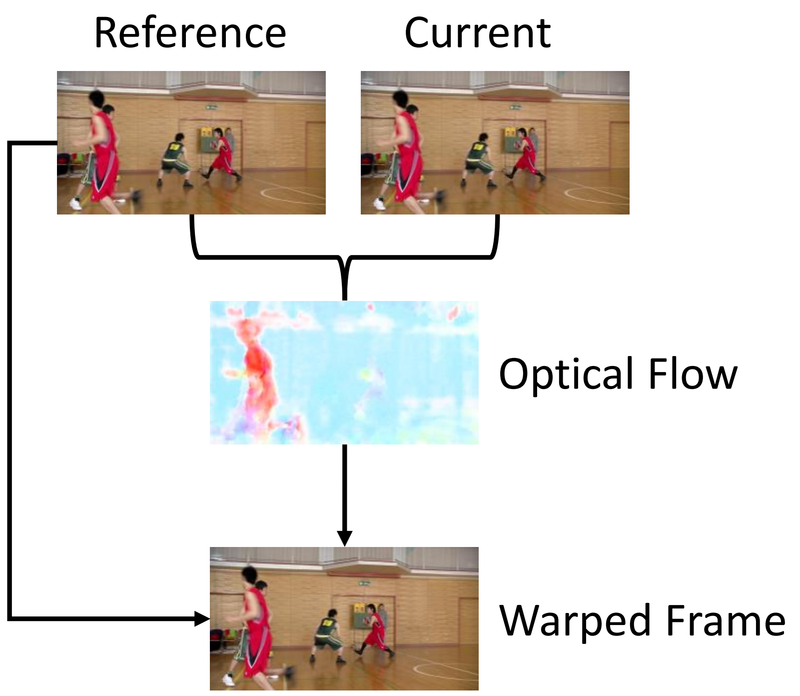
Limitations

1. Rely on existing network architecture
2. May need pretrain
3. Large residual due to inaccurate warp operation

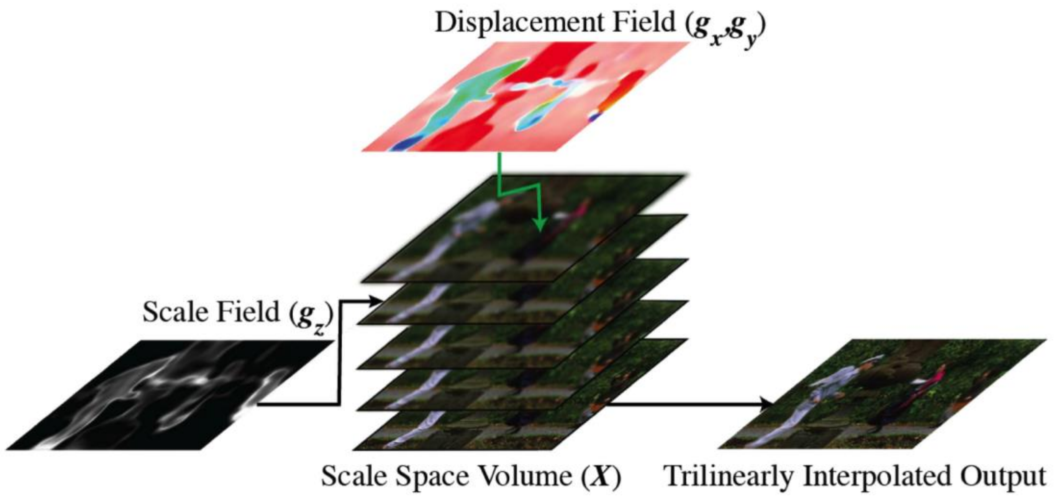
[5] Eirikur Agustsson, et al. “Scale-space Flow for End-to-End Optimized Video Compression”, in CVPR, 2020.

End-to-End Learned P-Frame Video Compression

Traditional Motion Compensation Procedure



Scale-space-warp[5] motion compensation procedure



Formulations

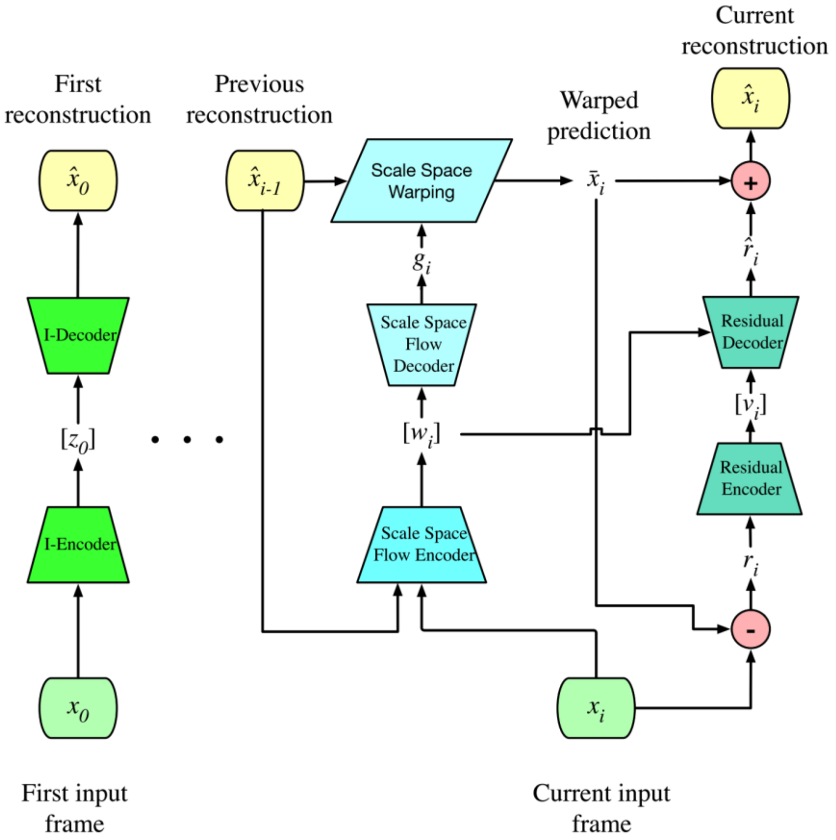
$$\mathbf{x}' := \text{Scale-Space-Warp}(\mathbf{x}, \mathbf{g}) \qquad \text{s.t.}$$

$$\mathbf{x}'[x,y] = \mathbf{X}[x + \mathbf{g}_x[x,y], y + \mathbf{g}_y[x,y], \mathbf{g}_z[x,y]]$$

[5] Eirikur Agustsson, et al. “Scale-space Flow for End-to-End Optimized Video Compression”, in CVPR, 2020.

End-to-End Learned P-Frame Video Compression

- Overall Architecture



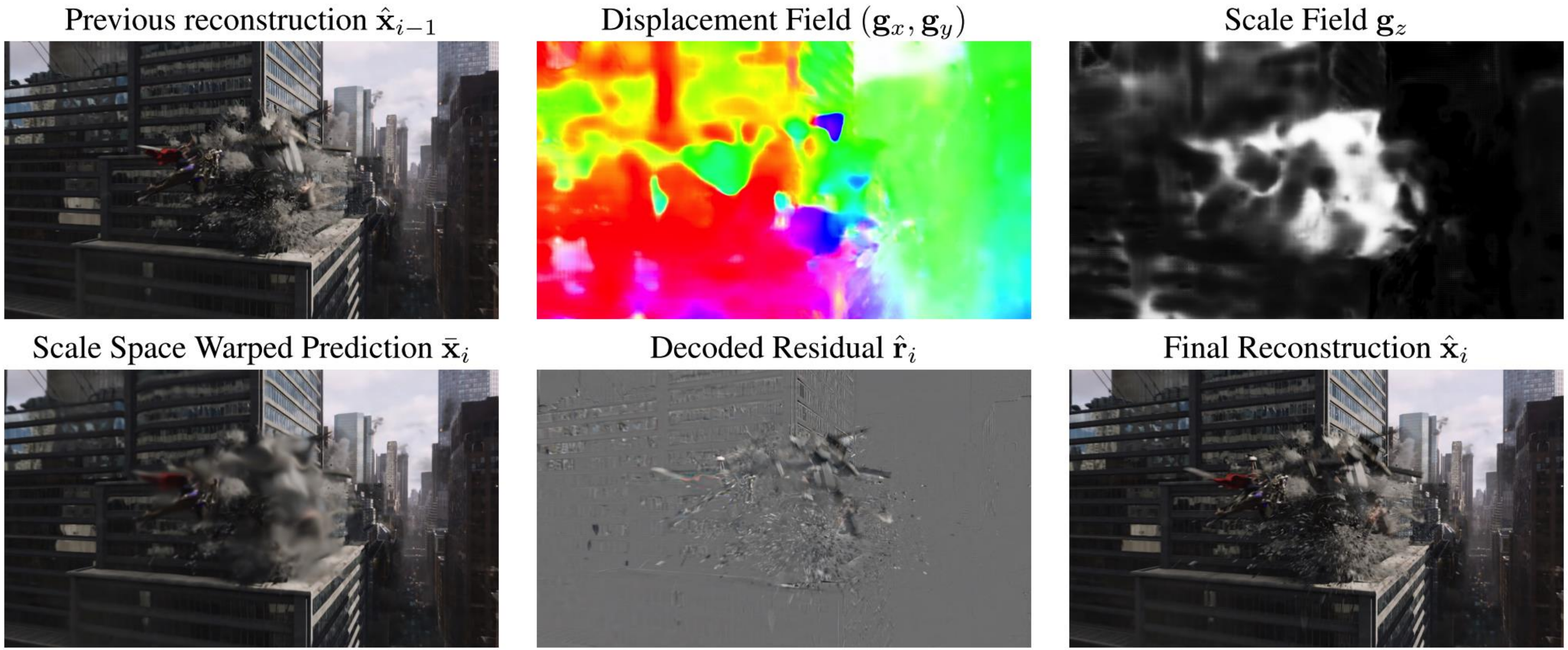
Hybrid Coding Approach:

1. Scale Space Flow Encoder & Decoder
2. Scale Space Warping based Motion Compensation
3. Residual Encoder & Decoder

[5] Eirikur Agustsson, et al. “Scale-space Flow for End-to-End Optimized Video Compression”, in CVPR, 2020.

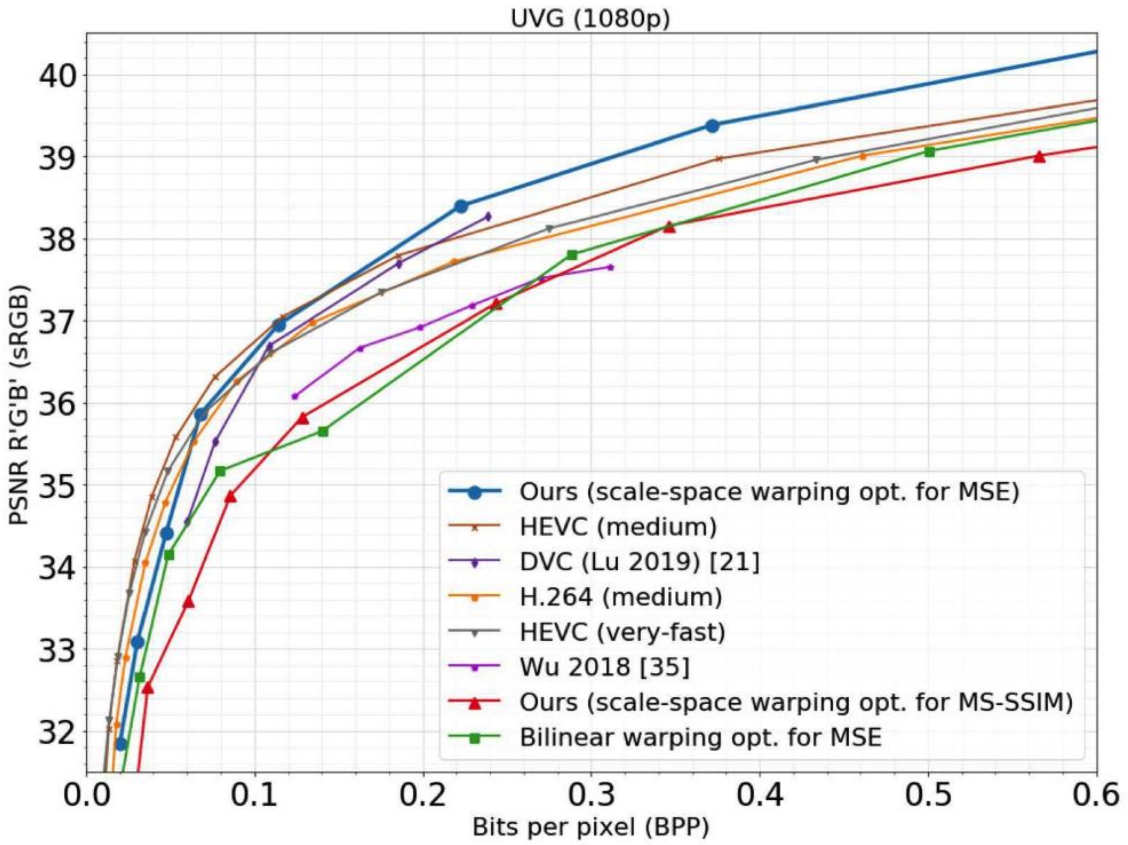
End-to-End Learned P-Frame Video Compression

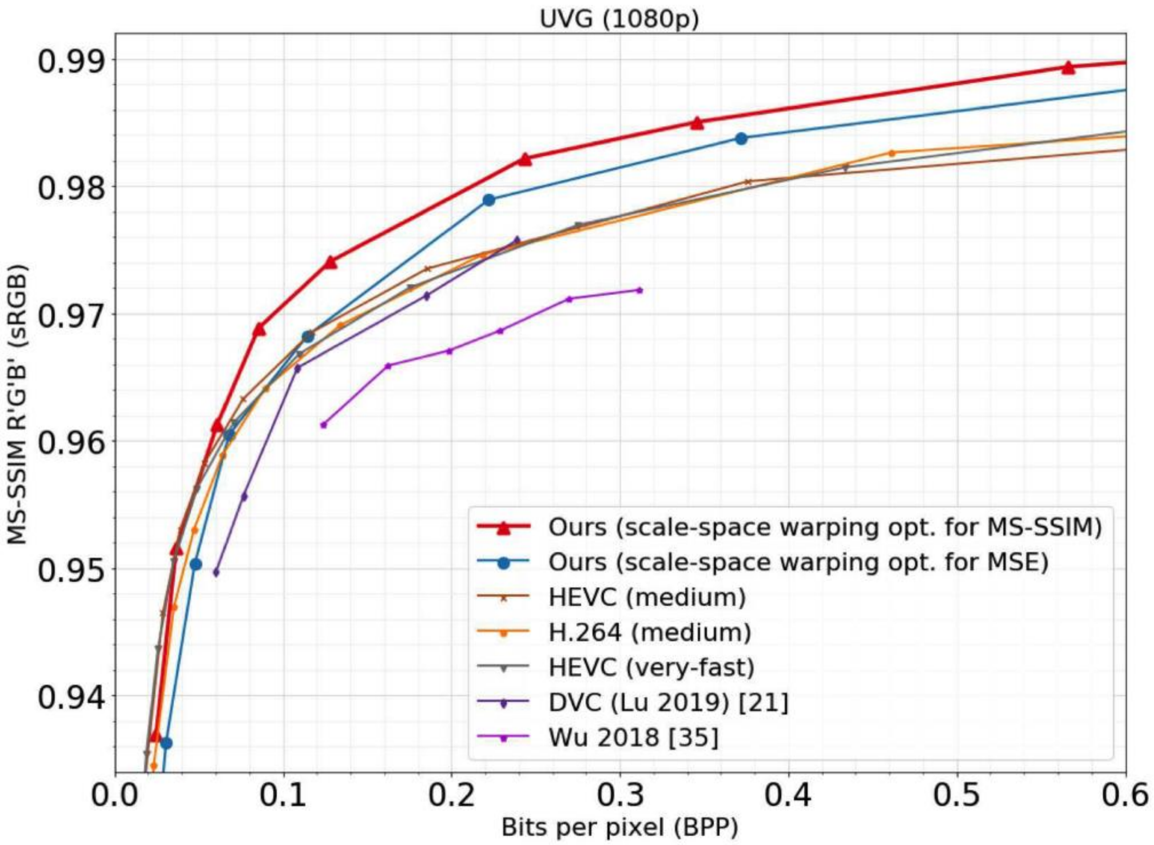
- Scale-space flow visulization



[5] Eirikur Agustsson, et al. "Scale-space Flow for End-to-End Optimized Video Compression", in CVPR, 2020.

End-to-End Learned P-Frame Video Compression





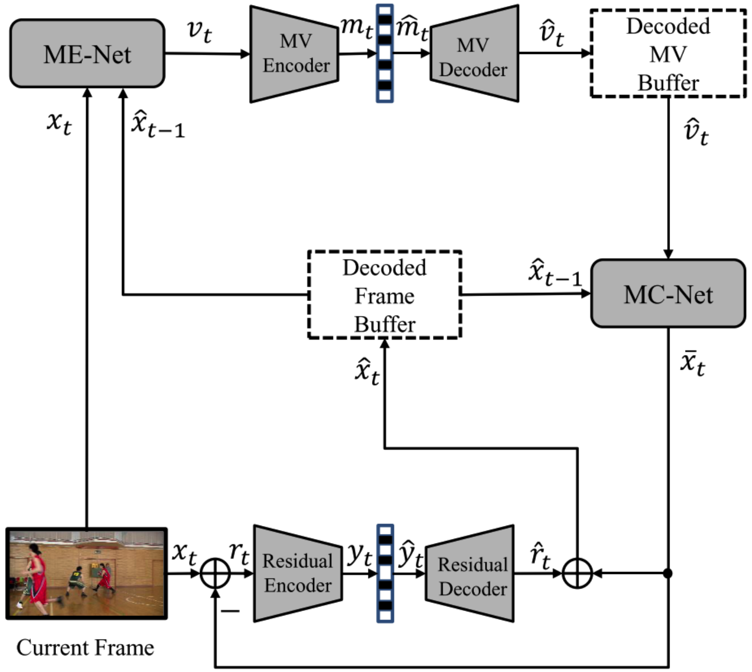
[5] Eirikur Agustsson, et al. “Scale-space Flow for End-to-End Optimized Video Compression”, in CVPR, 2020.

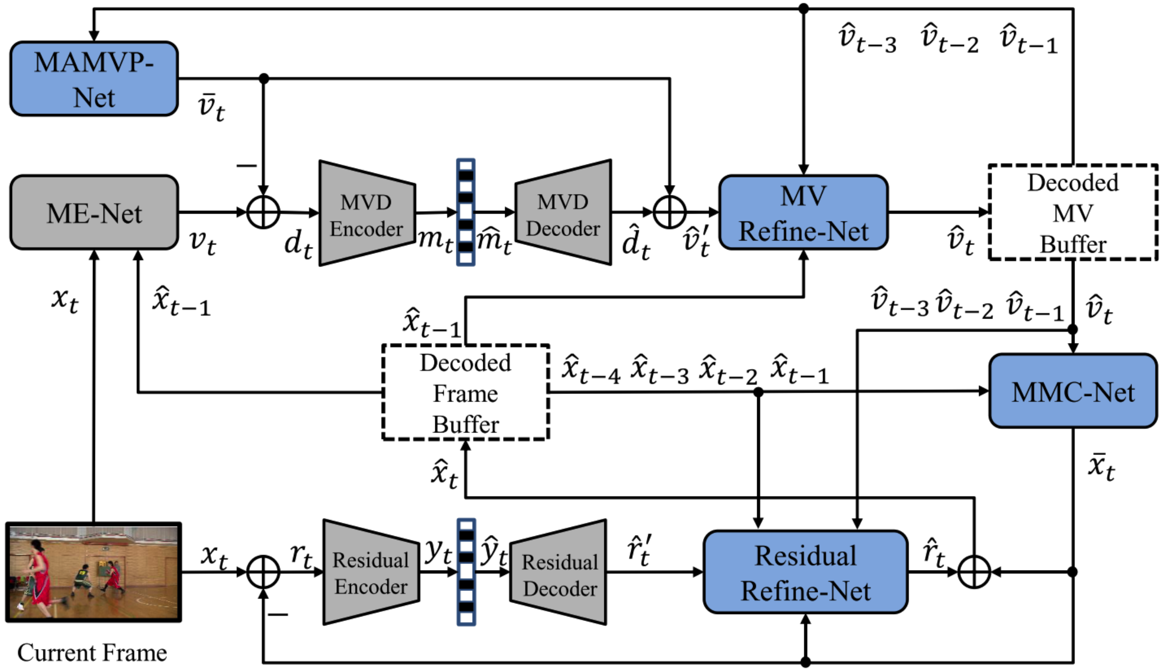
End-to-End Learned P-Frame Video Compression

- Existing methods use one previous reference frame
- Exploiting multiple reference frames for learned video compression
  - Directly use multiple frames for motion estimation or motion compensation.
  - Explore the long-range temporal information in latent space

End-to-End Learned P-Frame Video Compression

- Exploiting multiple reference frames for learned video compression

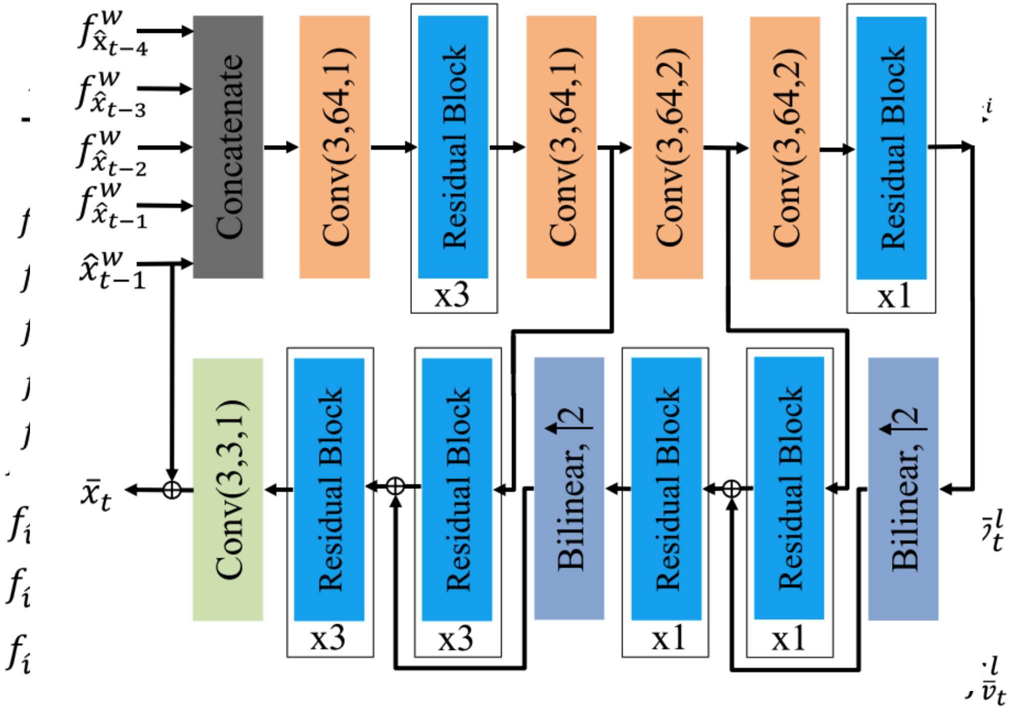




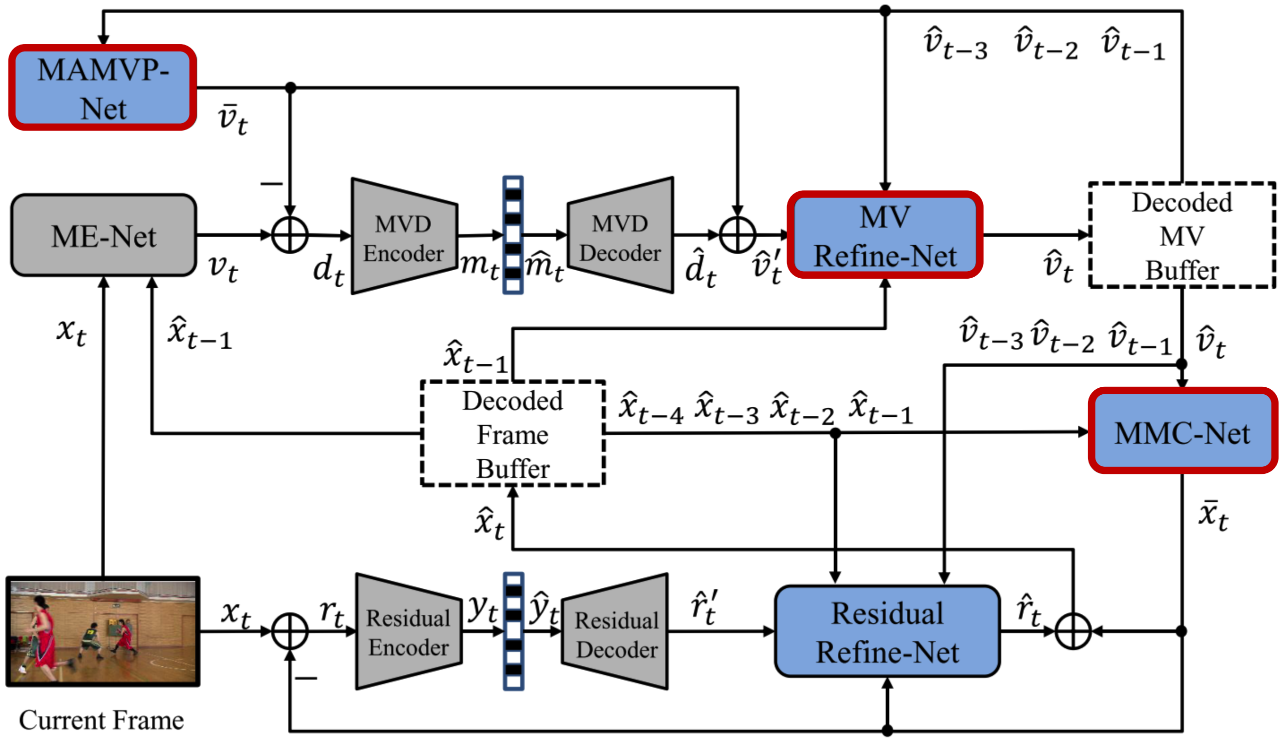
[6] Jianping Lin, et al. "M-LVC: Multiple Frames Prediction for Learned Video Compression", in CVPR, 2020.

End-to-End Learned P-Frame Video Compression

- M-LVC: Multiple Frames Prediction for Learned Video Compression



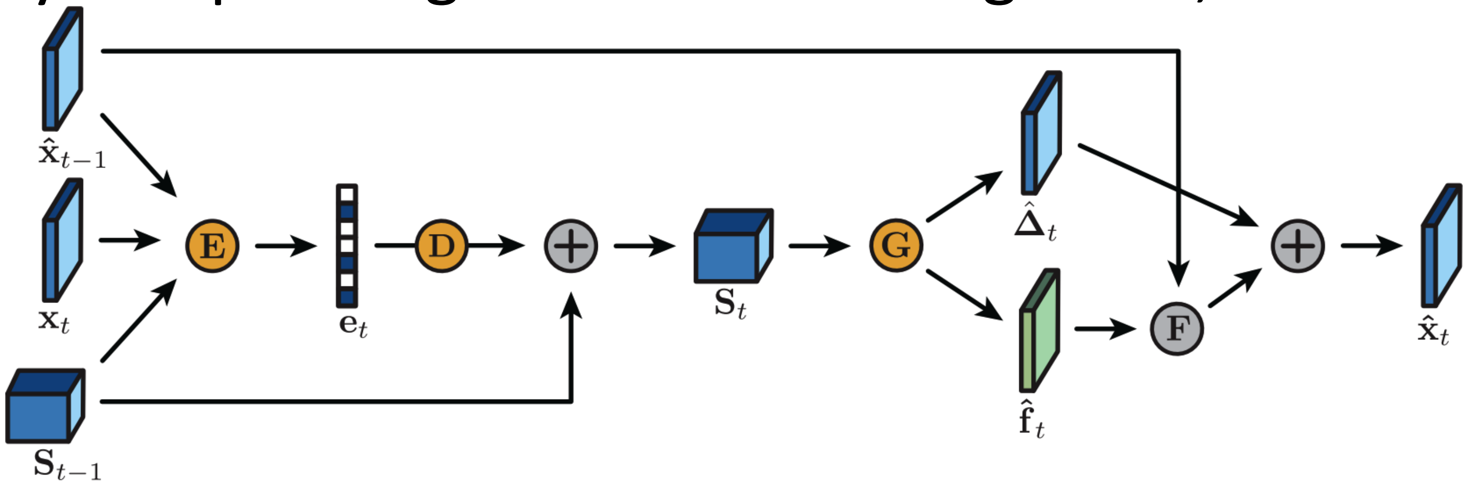
(b)



[6] Jianping Lin, et al. "M-LVC: Multiple Frames Prediction for Learned Video Compression", in CVPR, 2020.

End-to-End Learned P-Frame Video Compression

- Maintains a state of arbitrary information learned by the model and jointly compressing all transmitted signals[7];



S_{t-1} represents the state from previous time steps and includes the information from both residual and motion.

[7] Oren Rippel, et al. "Learned Video Compression", in ICCV, 2019.

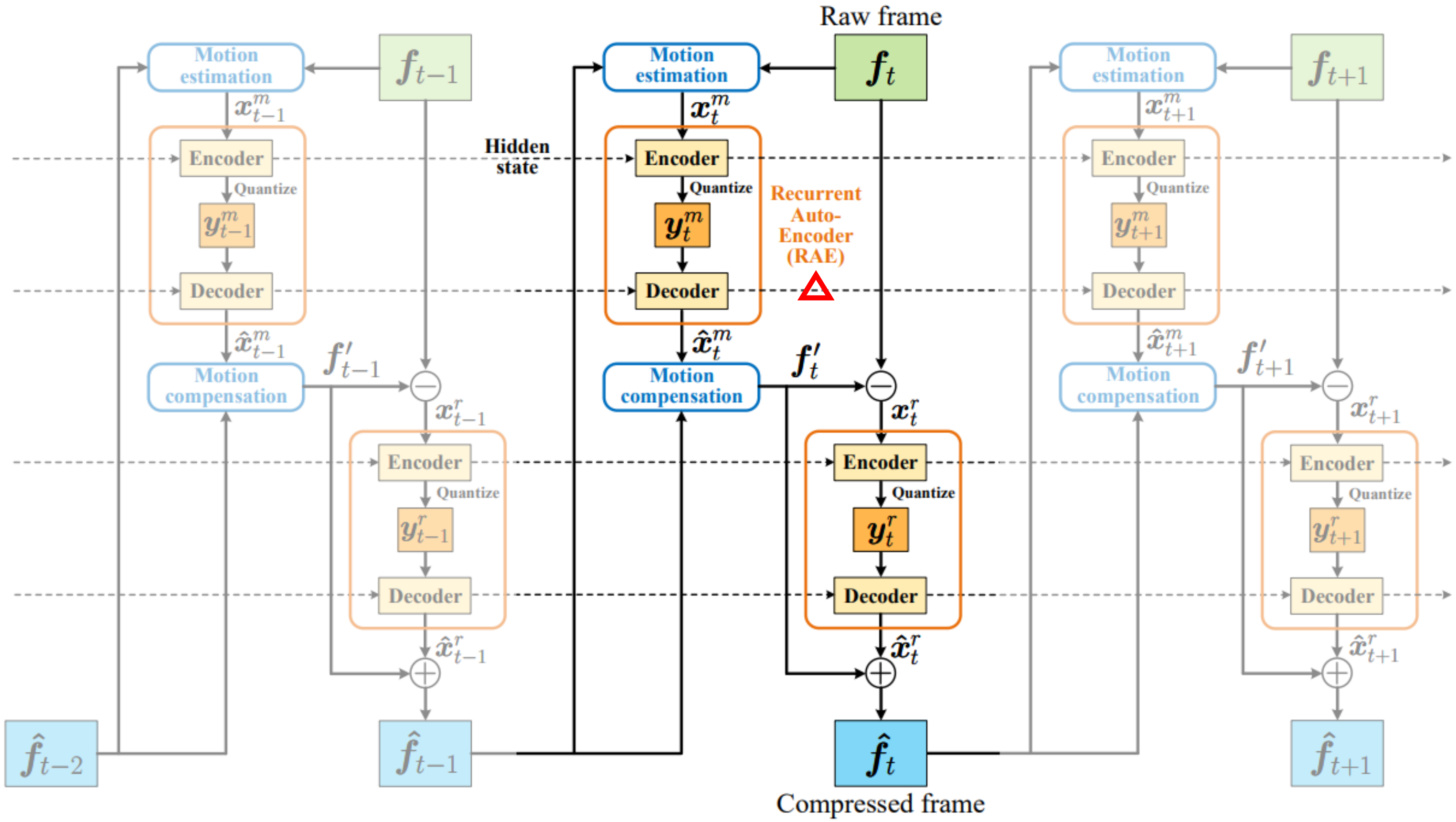
End-to-End Learned P-Frame Video Compression

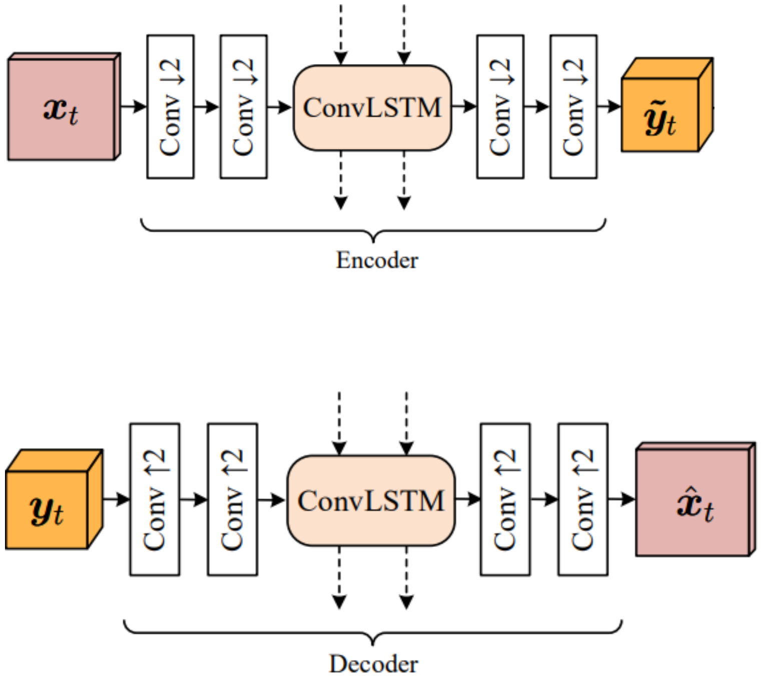
- The latent representations are generated based on limited reference frames;
- Existing work focus on the *independent* context information only;
  - motion compression and residual compression

-> exploiting the temporal redundancy to generate latent representations and more accurate context information

End-to-End Learned P-Frame Video Compression

- Implicitly explore temporal information in multiple frames



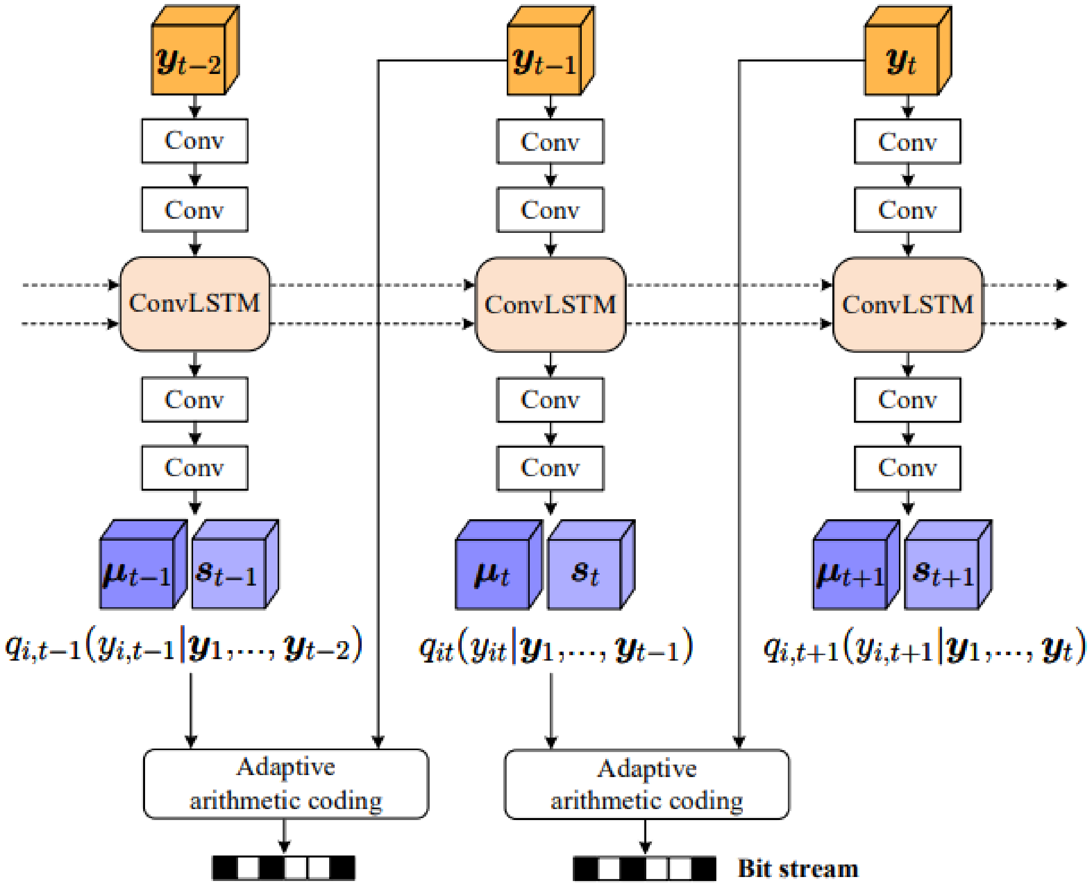


[8] Ren Yang, et al. "Learning for Video Compression with Recurrent Auto-Encoder and Recurrent Probability Model." in submission.

This slide is provided by Ren Yang.

End-to-End Learned P-Frame Video Compression

- Implicitly explore temporal information in multiple frames



Recurrent Probability Model (RPM)

$$H(p_t, q_t) = \mathbb{E}_{y_t \sim p_t}[-\log_2 q_t(y_t \mid y_1, \dots, y_{t-1})]$$

$$q_t(y_t \mid y_1, \dots, y_{t-1}) = \prod_{i=1}^{N} q_{it}(y_{it} \mid y_1, \dots, y_{t-1})$$

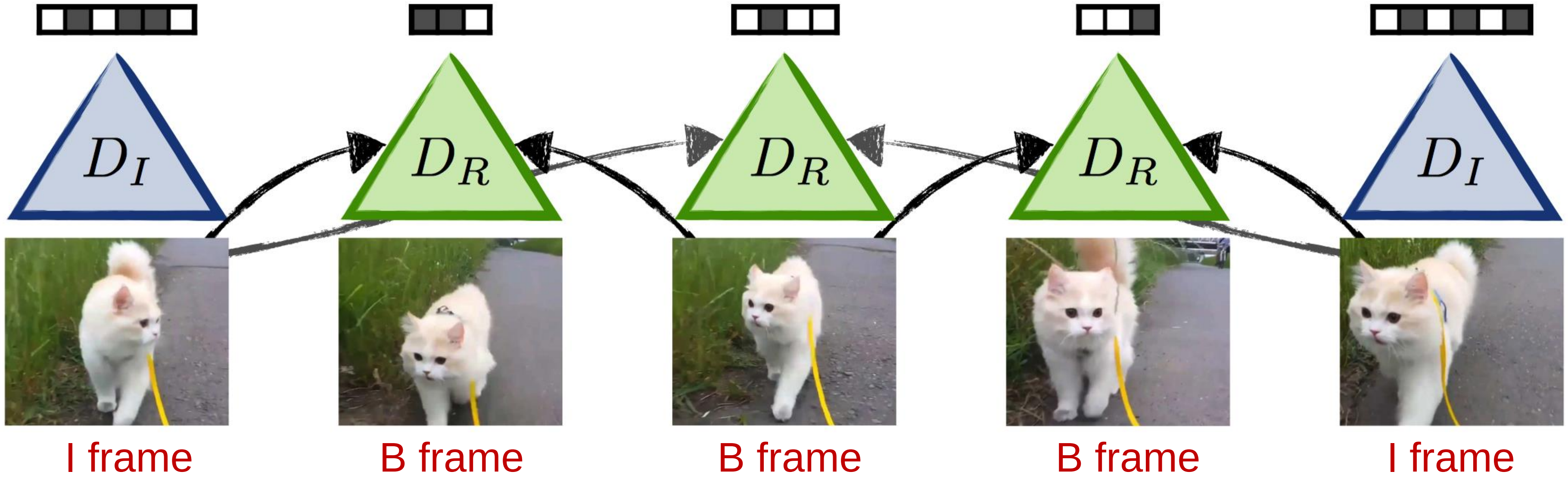
$$q_{it}(y_{it} \mid y_1, \dots, y_{t-1}) = \int_{y_{it}-0.5}^{y_{it}+0.5} \text{Logistic}(y; \mu_{it}, s_{it})dy$$

[8] Ren Yang, et al. "Learning for Video Compression with Recurrent Auto-Encoder and Recurrent Probability Model." in submission.

This slide is provided by Ren Yang.

End-to-End Learned B-Frame Video Compression

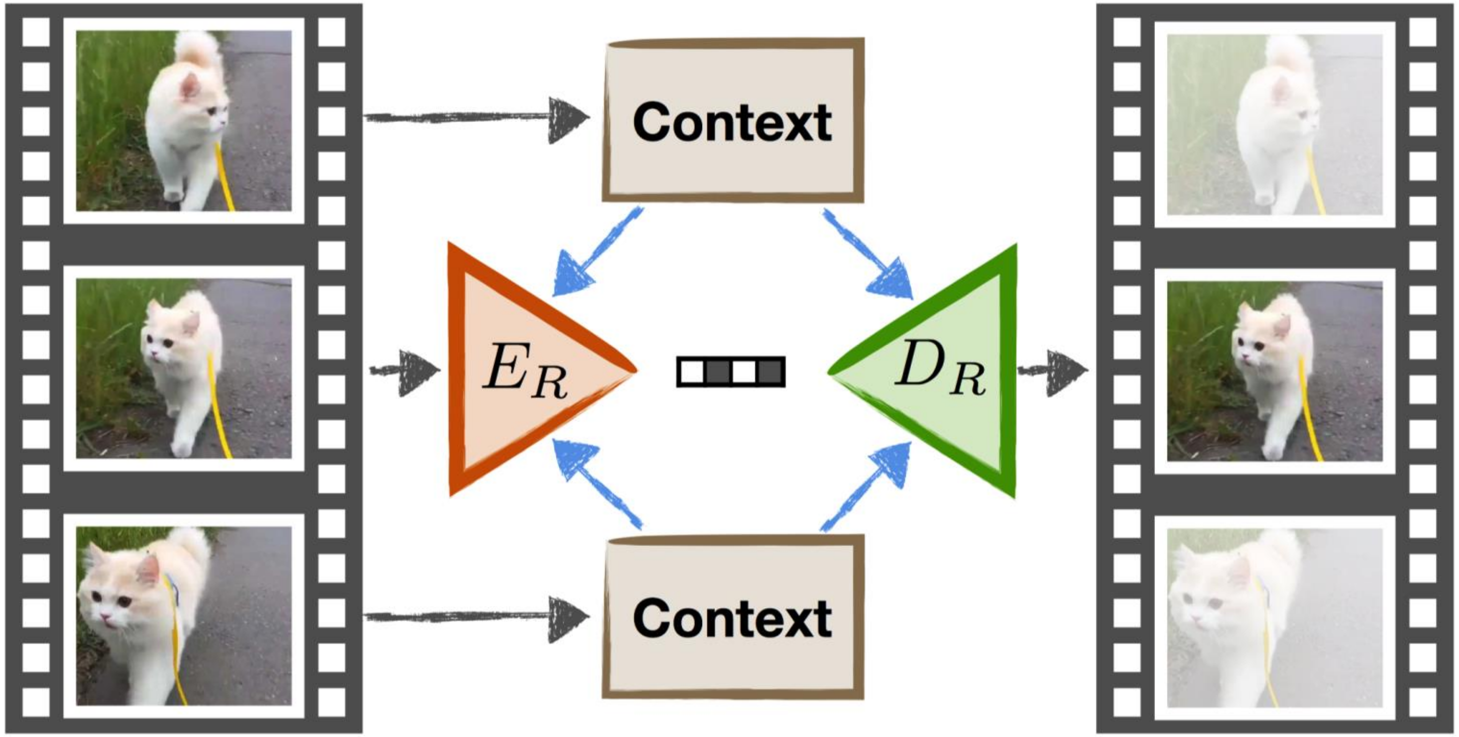
- Frame Interpolation based Video Compression



[9] Chao-yuan Wu, et al. "Video Compression through Image Interpolation," in ECCV, 2018

End-to-End Learned B-Frame Video Compression

- Frame Interpolation based Video Compression



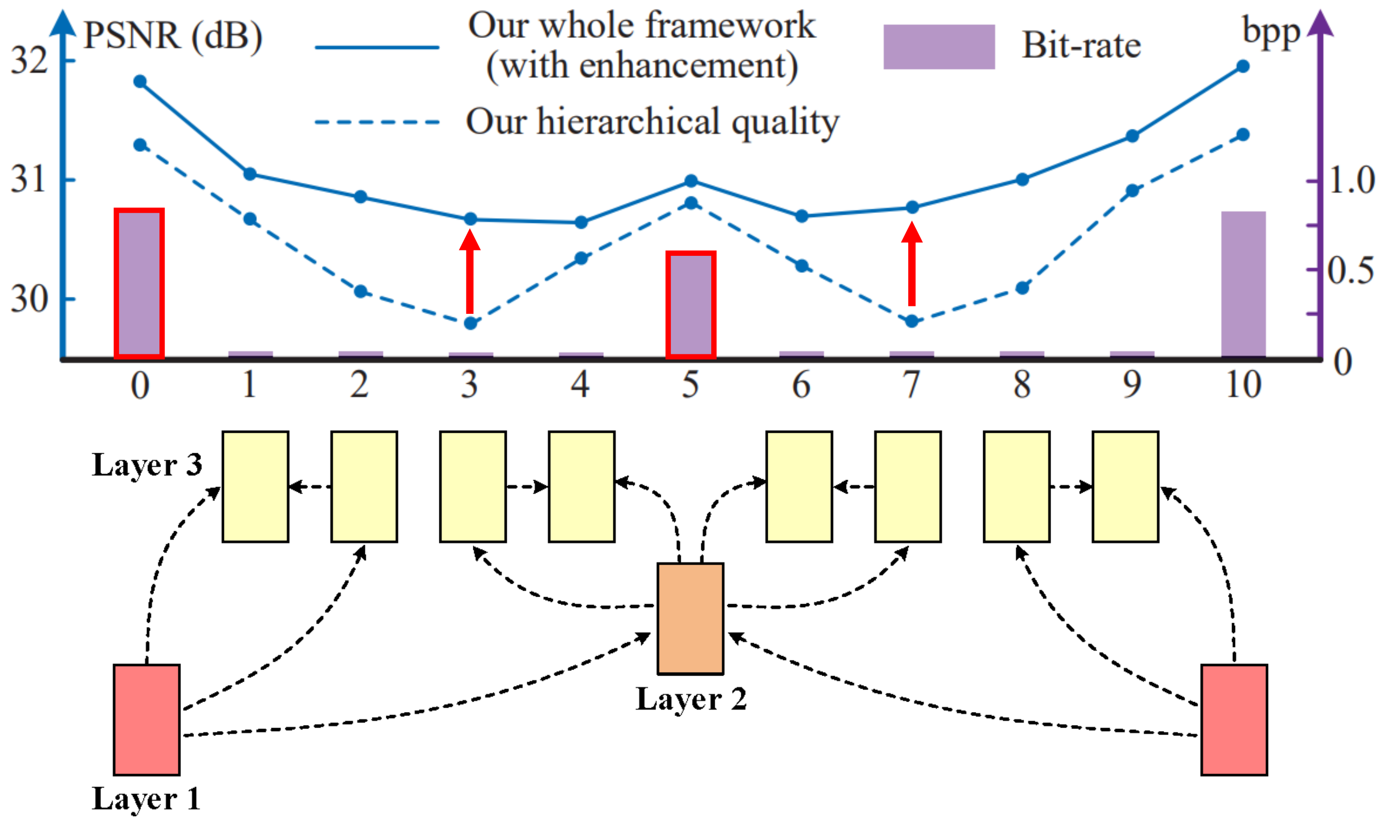
1. Extract features from reference images
2. Use block based motion estimation
3. Interpolate the current frame
4. Compress residual using learned image codec
5. Compress motion using traditional image codec

Limitations:
1. Not end-to-end optimized
2. Motion compression is not learnt

[9] Chao-yuan Wu, et al. "Video Compression through Image Interpolation," in ECCV, 2018

End-to-End Learned B-Frame Video Compression

- Hierarchical Learned Video Compression (HLVC) with recurrent enhancement [1]



The benefits of hierarchical quality are two-fold:

- **At encoder side**, the high quality frames provide high quality references to improve the compression performance of other frames.

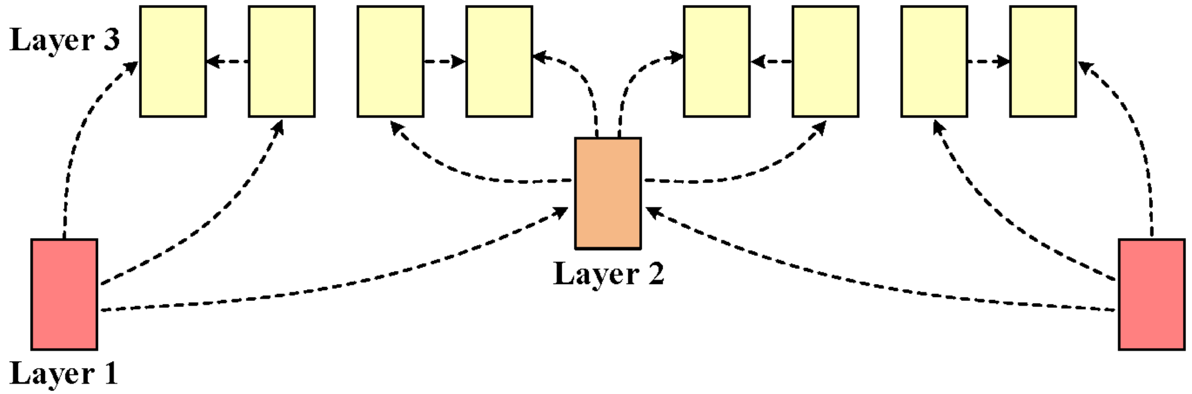
- **At decoder side**, the low quality frames can be enhanced by taking advantage of high quality frames without bit-rate overhead. It is equivalent to reducing bit-rate on low quality frames.

[10] Ren Yang, et al. "Learning for Video Compression with Hierarchical Quality and Recurrent Enhancement." in CVPR. 2020.

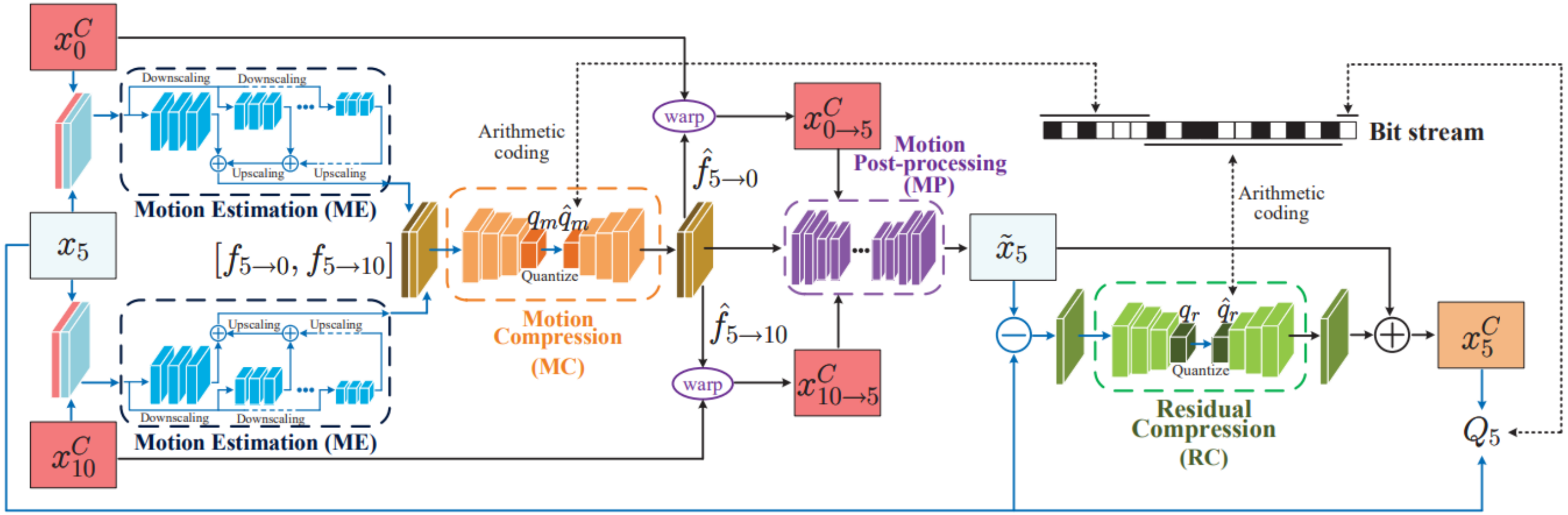
End-to-End Learned B-Frame Video Compression

- Hierarchical Learned Video Compression (HLVC) with recurrent enhancement [1]

Layer 1:
Compressed by BPG for PSNR model, and by Lee *et al.* ICLR 2019 for MS-SSIM model.



Layer 2:
Compressed by the proposed Bi-Directional Deep Compression (BDDC) network



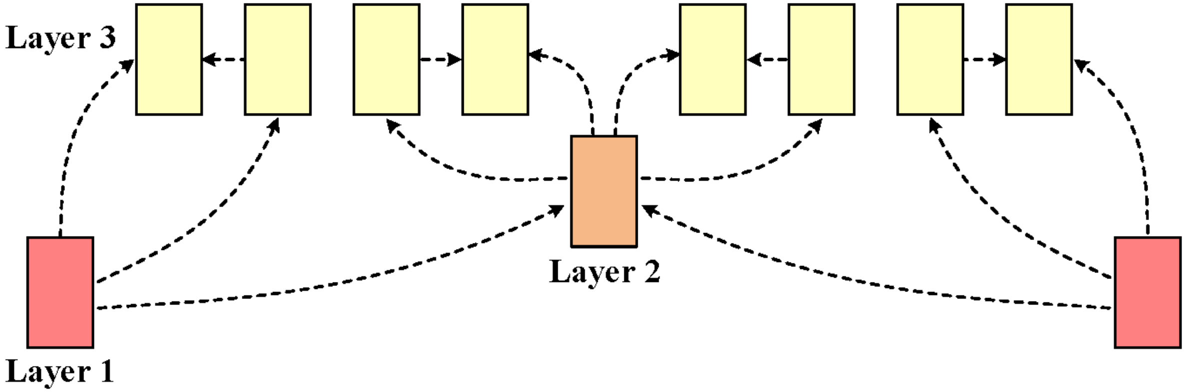
[10] Ren Yang, et al. "Learning for Video Compression with Hierarchical Quality and Recurrent Enhancement." in CVPR. 2020.

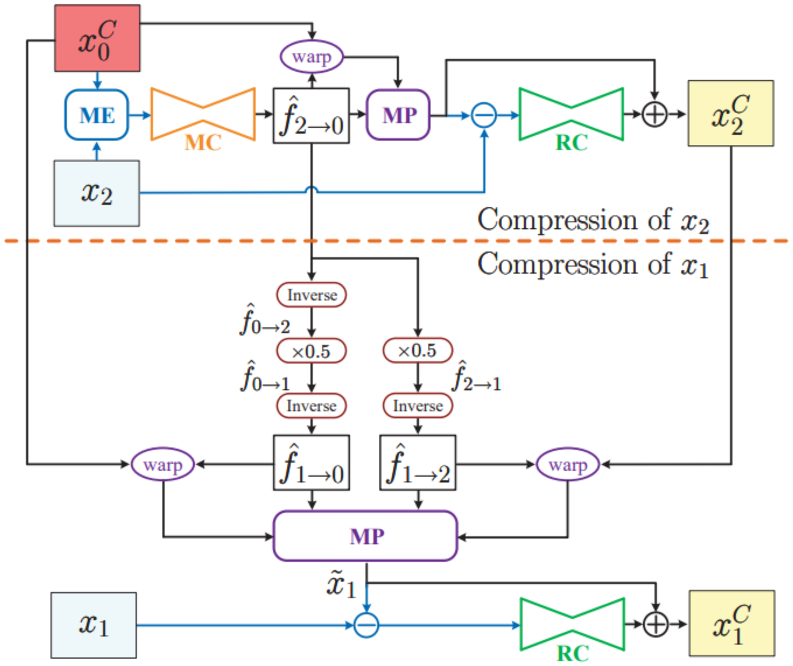
This slides is provided by Ren Yang

End-to-End Learned B-Frame Video Compression

- Hierarchical Learned Video Compression (HLVC) with recurrent enhancement [1]

Layer 3:
Compressed by the proposed Single Motion Deep Compression (SMDC) network





Due to the correlation of motions among multiple neighboring frames, we propose using the motion between x_0^C and x_2 to predict the motions between x_1 and x_0^C or x_2. That is,

$$\hat{f}_{1\to0} = \underbrace{\text{Inverse}(0.5 \times \underbrace{\text{Inverse}(\hat{f}_{2\to0})}_{\hat{f}_{0\to2}})}_{\hat{f}_{0\to1}}.$$

As such, x_1 can be compressed with the reference frames of x_0^C and x_2, **without bits consumed for motion map**, thus improving the rate-distortion performance.

End-to-End Learned B-Frame Video Compression

- Previous works use separate interpolation network and motion compression module

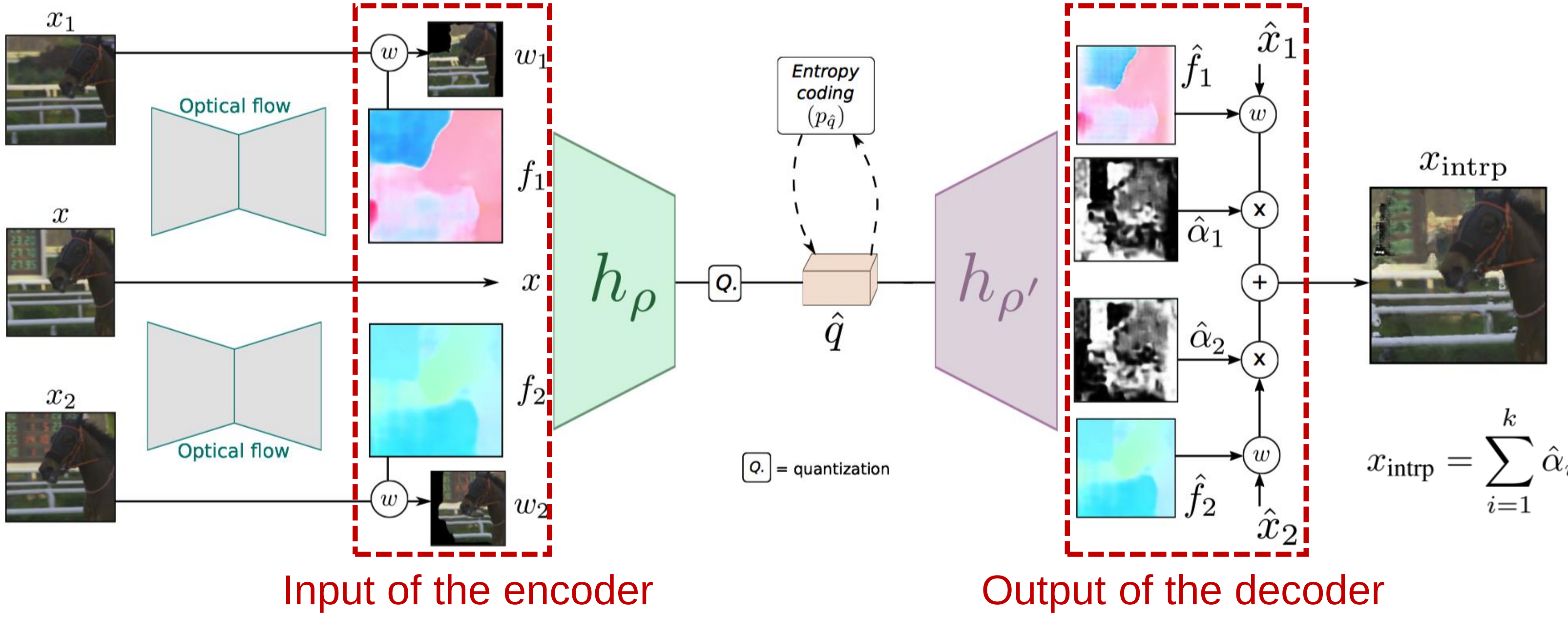
-> combine interpolation network and motion compression

- The residual is compressed in the pixel domain and it is a non-trivial task.

-> Feature space residual compression

End-to-End Learned B-Frame Video Compression

- Combine interpolation and flow compression and decode the flow and interpolation coefficients simultaneously.

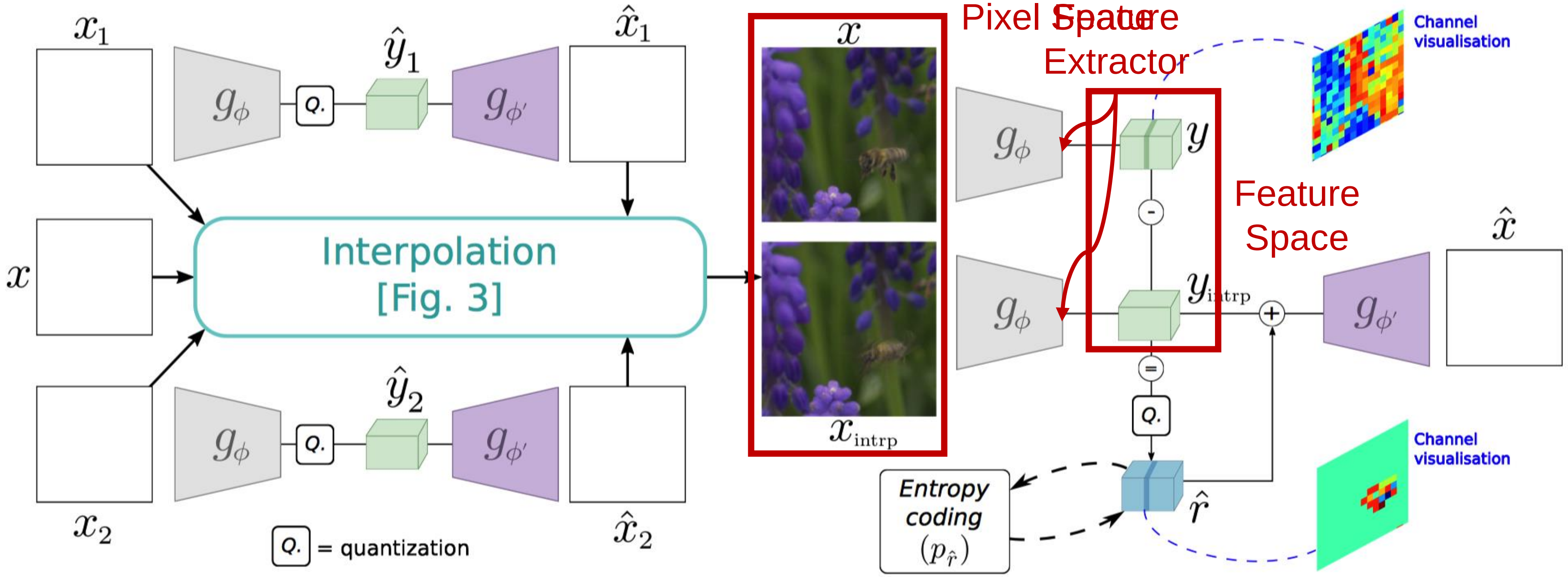


$$x_{\text{intrp}} = \sum_{i=1}^{k} \hat{\alpha}_i w(x_i, \hat{f}_i) \text{ with } \sum_{i=1}^{k} \hat{\alpha}_i = 1.$$

[11] Abdelaziz Djelouah, et al. "Neural Inter-Frame Compression for Video Coding." in ICCV. 2019.

End-to-End Learned B-Frame Video Compression

- Residual Compression in Latent Space



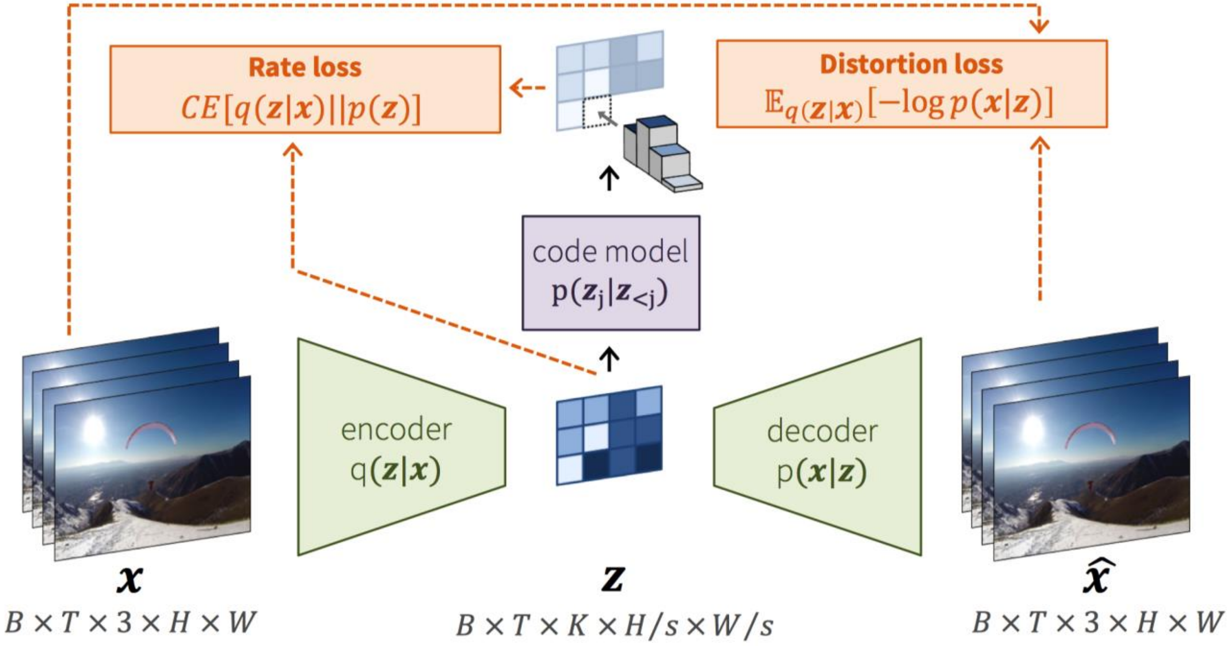
[11] Abdelaziz Djelouah, et al. "Neural Inter-Frame Compression for Video Coding." in ICCV. 2019.

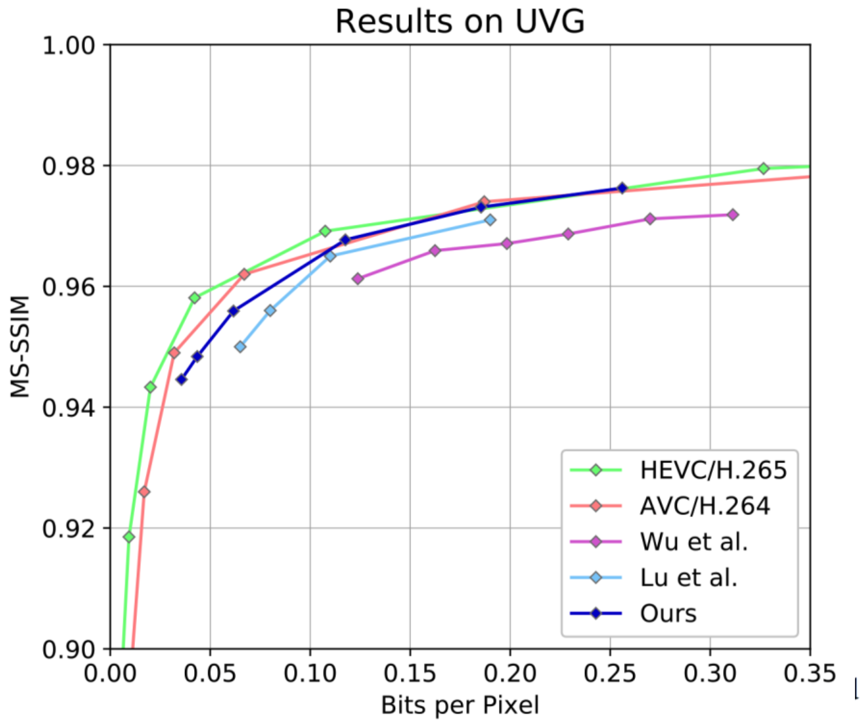
Learned Autoencoder based Video Compression

- Previous works follow the hybrid coding framework, i.e., motion compensation and residual coding.
- Using optical flow for explicitly motion estimation
- Separately motion and residual compression

Learned Autoencoder based Video Compression

- Use 3D autoencoders to compress video frames without explicitly motion estimation.





[12] Amirhossein Habibian, et al. "Video compression with rate-distortion autoencoders ." in ICCV. 2019.

Discussion

- Open-Source Project
  - Pytorch Data Compression
    - Learned Image Compression
      - Balle, ICLR2017
      - Balle, ICLR2018
      - Minne, NeurIPS2018
    - Learned Video Compression
      - DVC, CVPR2019
      - HU[4], ECCV2020(ongoing)
    - Learned Point Cloud Compression
      - OctSqueeze, CVPR2020

https://github.com/ZhihaoHu/PyTorchDataCompression/

Reference

- [1] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. “DVC: An End-to-end Deep Video Compression Framework.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11006-11015. 2019.
- [2] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu.“An End-to-End Learning Framework for Video Compression.” IEEE Transactions on Pattern Analysis and Machine Intelligence (T- PAMI), 2020.
- [3] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. “DVC: An End-to-end Deep Video Compression Framework.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11006-11015. 2019.
- [4] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, Shuhang Gu. “Improving Deep Video Compression by Resolution-adaptive Flow Coding.” In Proceedings of the European Conference on Computer Vision (ECCV),2020.
- [5] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. "Scale-Space Flow for End-to-End Optimized Video Compression." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8503-8512. 2020.
- [6] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. "M-LVC: Multiple Frames Prediction for Learned Video Compression." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3546-3554. 2020.
- [7] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G. Anderson, and Lubomir Bourdev. "Learned video compression." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3454-3463. 2019.
- [8] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. "Learning for Video Compression with Recurrent Auto-Encoder and Recurrent Probability Model." arXiv preprint arXiv:2006.13560 (2020).
- [9] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. "Video compression through image interpolation." In Proceedings of the European Conference on Computer Vision (ECCV), pp. 416-431. 2018.
- [10] Yang, Ren, Fabian Mentzer, Luc Van Gool, and Radu Timofte. "Learning for video compression with hierarchical quality and recurrent enhancement." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6628-6637. 2020.
- [11] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. "Neural inter-frame compression for video coding." In Proceedings of the IEEE International Conference on Computer Vision, pp. 6421-6429. 2019.
- [12] Amirhossein Habibian, Ties van Rozendaal, Jakub M. Tomczak, and Taco S. Cohen. "Video compression with rate-distortion autoencoders." In Proceedings of the IEEE International Conference on Computer Vision, pp. 7033-7042. 2019.

Acknowledge

- Co-authors
  - Chunlei Cai(SJTU)
  - Zhihao Hu(BUAA)
  - Zhenghao Chen(Usyd)
  - …
- Benchmark results from
  - Ren Yang(ETH Zurich)
  - Chao-yuan Wu(UT Austin)




Dong Xu
University of Sydney, Australia

Guo Lu
Beijing Institute of Technology, China

Ren Yang
ETH Zurich, Switzerland

Radu Timofte
ETH Zurich, Switzerland

Q&A