Towards Efficient Image Compression Without Autoregressive Models

Muhammad Salman Ali   Yeongwoong Kim   Maryam Qamar
Sung-Chang Lim   Donghyun Kim   Chaoning Zhang
Sung-Ho Bae   Hui Yong Kim

STRUCT Group Seminar
Presenter: Haowei Kuang
2024.03.03

NeurIPS 2023

OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

BACKGROUND: Image Compression

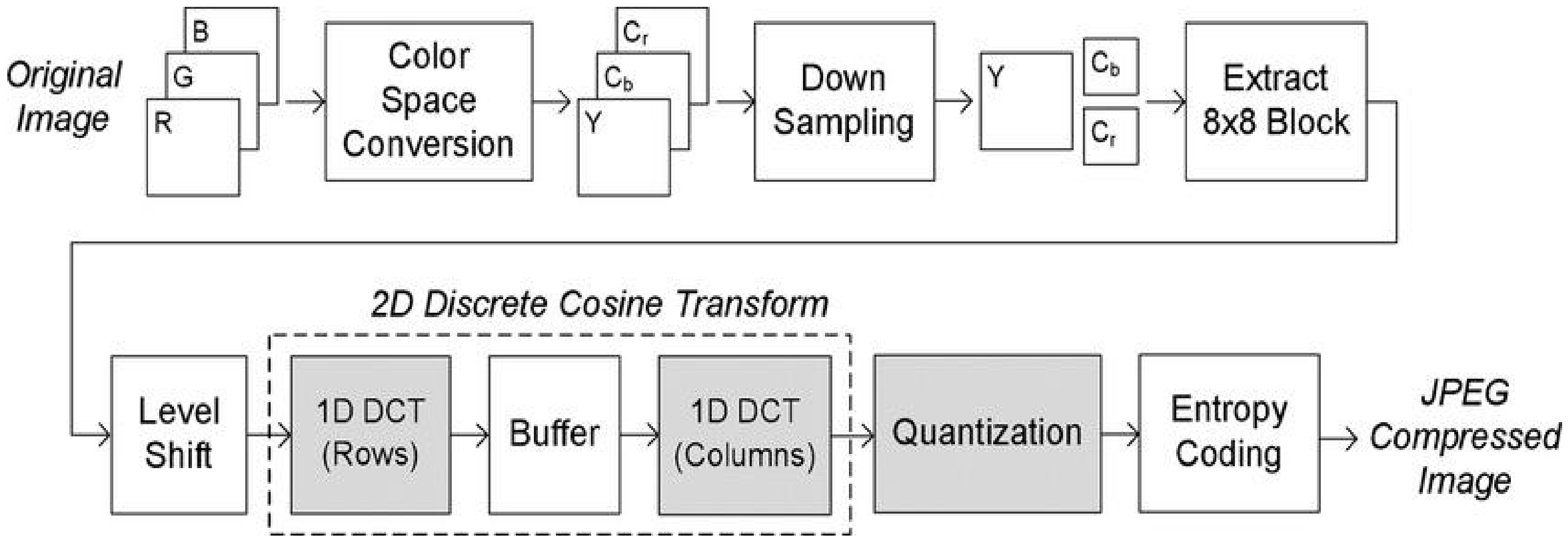
Memory Used:
$768 \times 512 \times 3 \times 8 = 9437184$ bits ≈ 1.2MB
(24 bits per pixel)

PNG Format:
736.5 KB
(15 bits per pixel)

JPEG Format:
34 KB
(0.7 bits per pixel)

BACKGROUND: Conventional Image Compression

- Transform
- Quantization
- Entropy coding

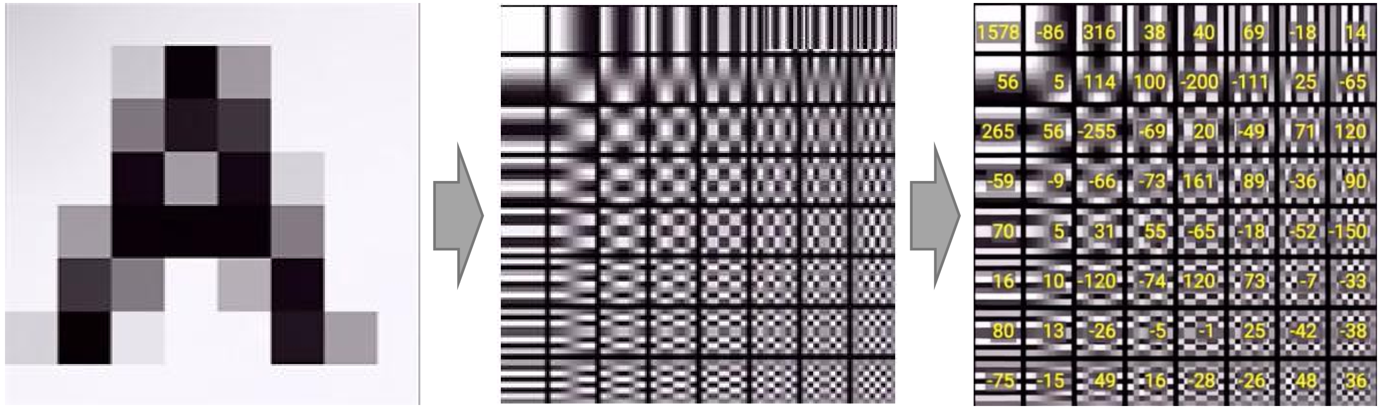


BACKGROUND: Conventional Image Compression

Transform

Transform to a better space
(in JPEG, remove information that eyes are not great at perceiving)

- Color Space Conversion
- Discrete Cosine Transform



$$F(u,v) = \frac{1}{4} C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) * \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}\right]$$

$$f(x,y) = \frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v) \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}\right]$$

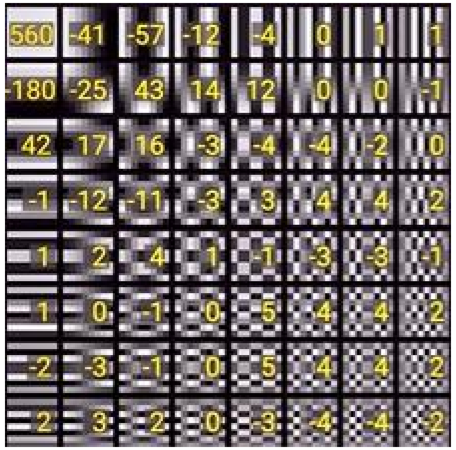
where: $C(u),\ C(v) = 1/\sqrt{2}$ for $u,v = 0$;

$C(u),\ C(v) = 1$ otherwise.

BACKGROUND: Conventional Image Compression

Quantization

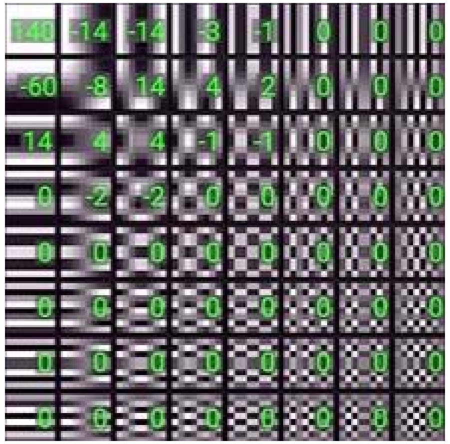
$$F^{Q}(u,v) = Integer\ Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$



÷

04	03	04	04	04	06	11	15
03	03	03	04	05	08	14	19
03	04	04	05	08	12	16	20
04	05	06	07	12	14	18	20
06	06	09	11	14	17	21	23
09	12	12	18	23	22	25	21
11	13	15	17	21	23	25	21
13	12	12	13	16	19	21	21

=

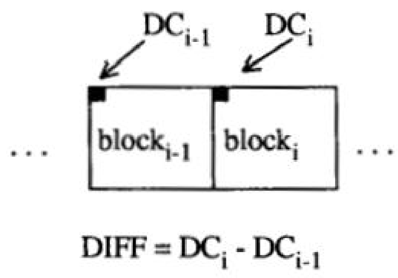


$$F^{Q'}(u,v) = F^{Q}(u,v) * Q(u,v)$$

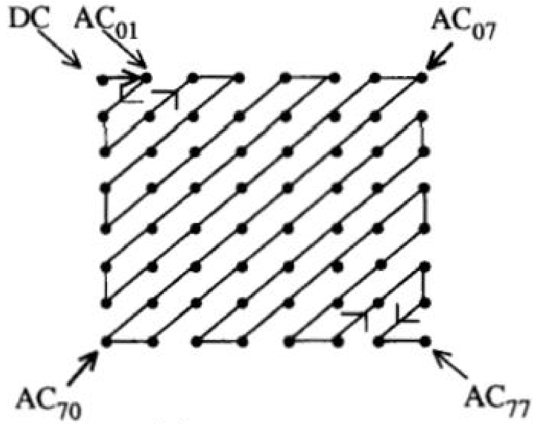
BACKGROUND: Conventional Image Compression

Entropy coding

- Huffman Coding
- Arithmetic coding



Differential DC encoding



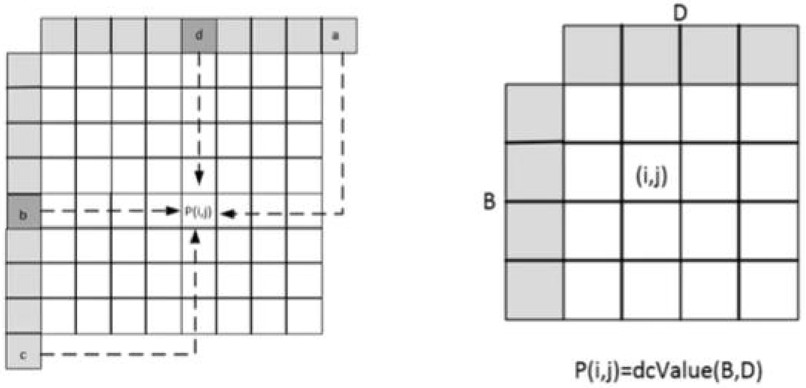
Zig–zag sequence

BACKGROUND: Conventional Image Compression

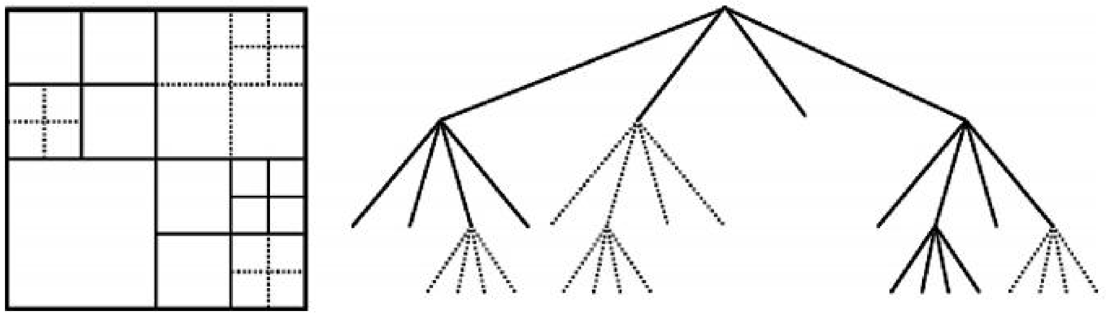
Other tricks

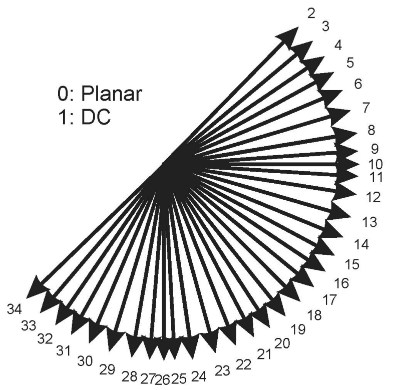
- Residual Prediction
- Coding Tree Units
- Coarse to Fine Prediction

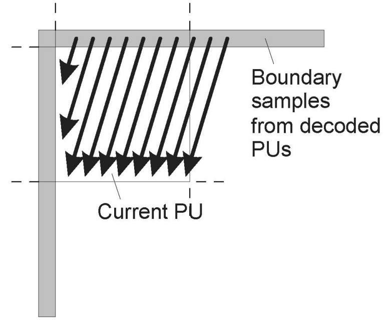
…………



Reduce coded content through various predictive strategies







BACKGROUND: E2E Image Compression

◆ Transform

Transfer to a latent space

◆ Quantization

--> add noise

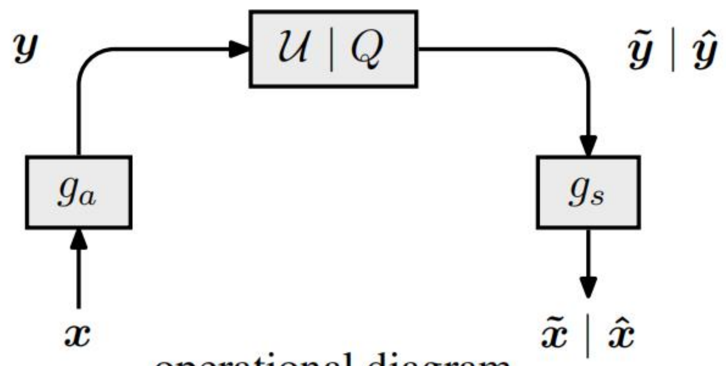
◆ Entropy coding

The lower the entropy of the latent variable y, the lower the bitrate

$$R = E_{\hat{y} \sim m}[-log_2 p_{\hat{y}}(\hat{y})]$$

◆ Training Strategy

Trade of between distortion and rate: *L* = *λ*·*D* + *R*



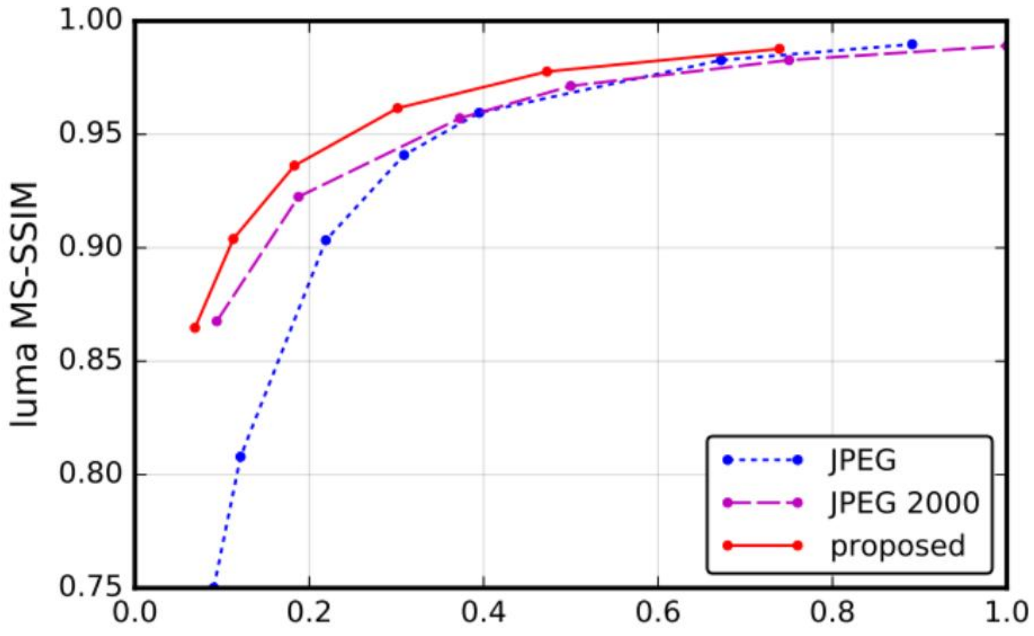
operational diagram

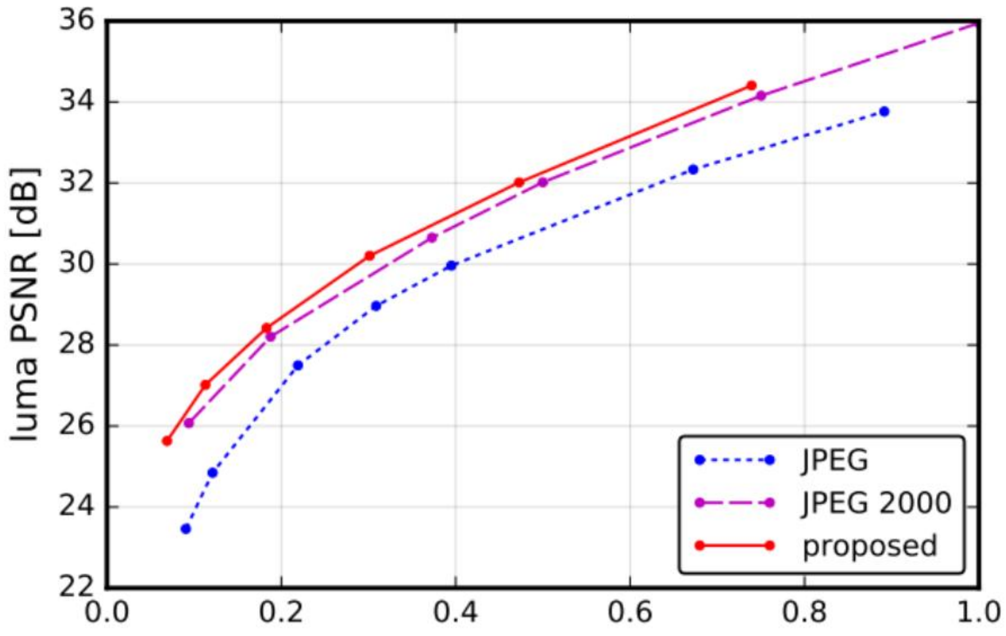
BACKGROUND: E2E Image Compression

- Metrics

  RD-Curve

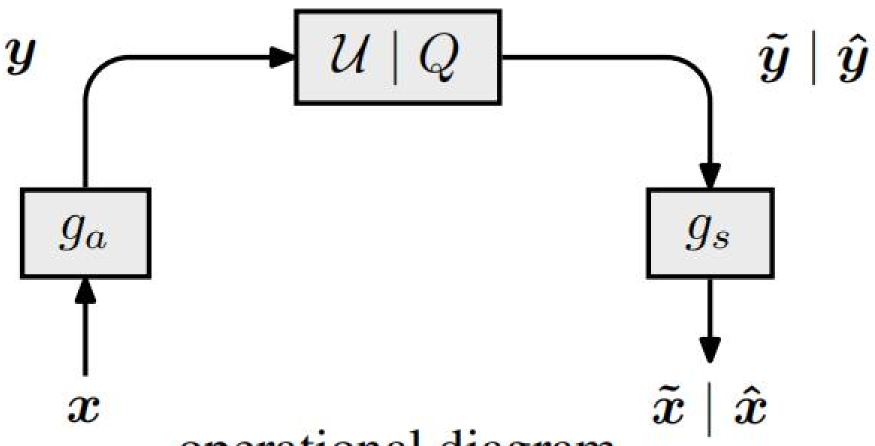
  BD-rate



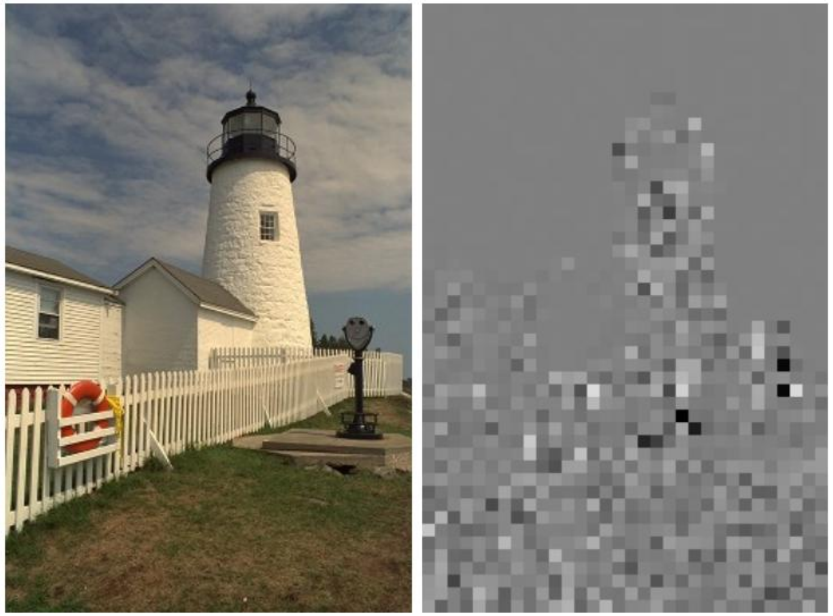


BACKGROUND: E2E Image Compression

- Visualize a subset of the y
  - Ideal state: No spatial coupling
  - Actually: Non-zero responses are highly clustered in areas of high contrast



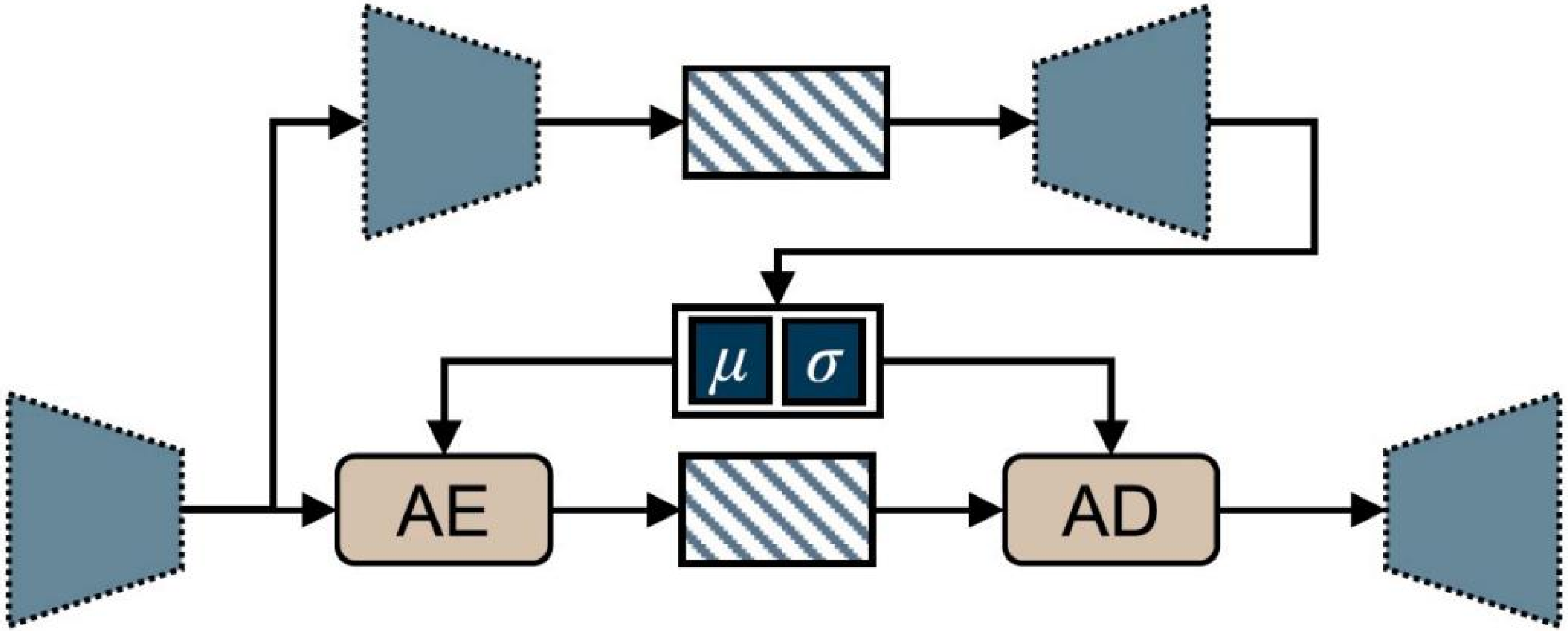
operational diagram



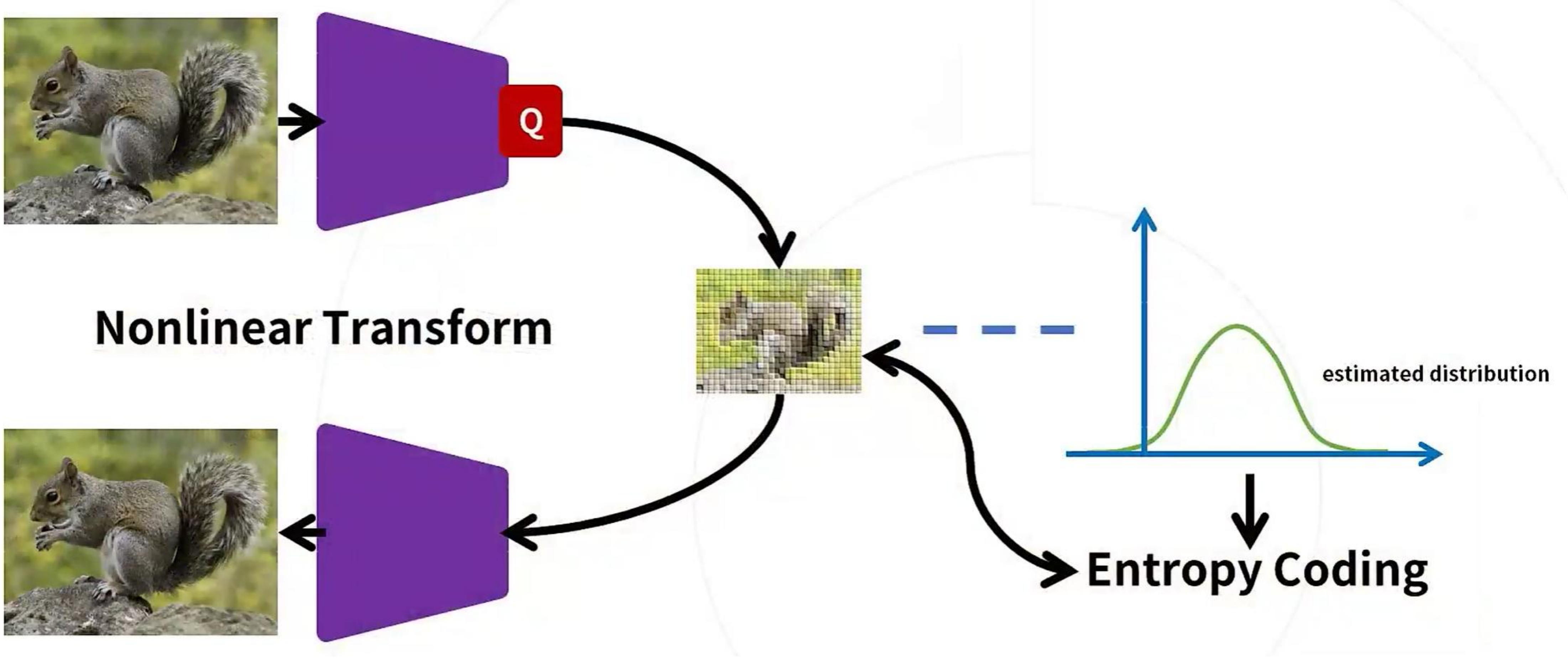
BACKGROUND: E2E Image Compression

Introducing a hyperprior

- When the entropy distribution is consistent with the real distribution, the stream is minimum
- Simplified the distribution of each pixel to a Gaussian distribution
- Predict μ and σ for each pixel use hyperprior

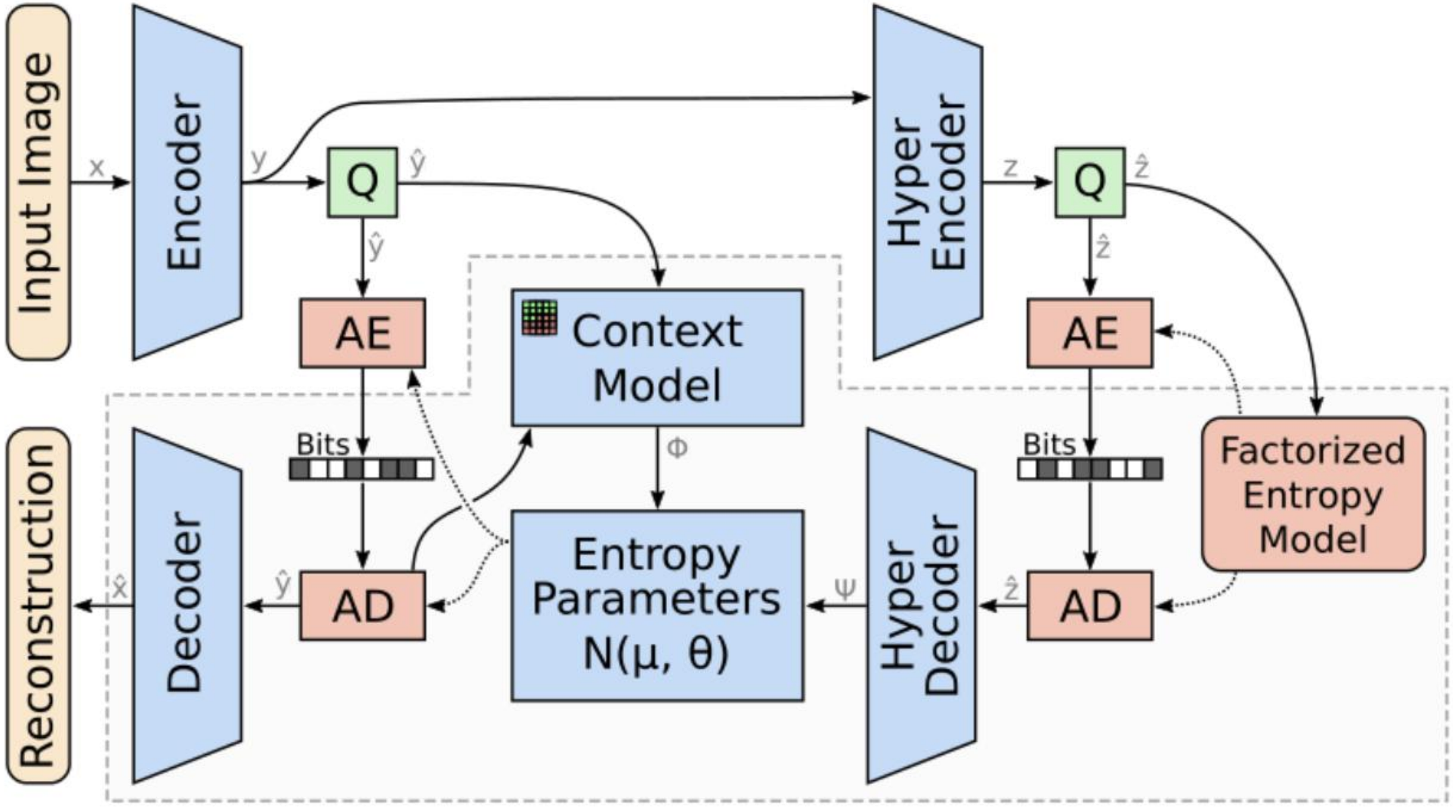


BACKGROUND: E2E Image Compression



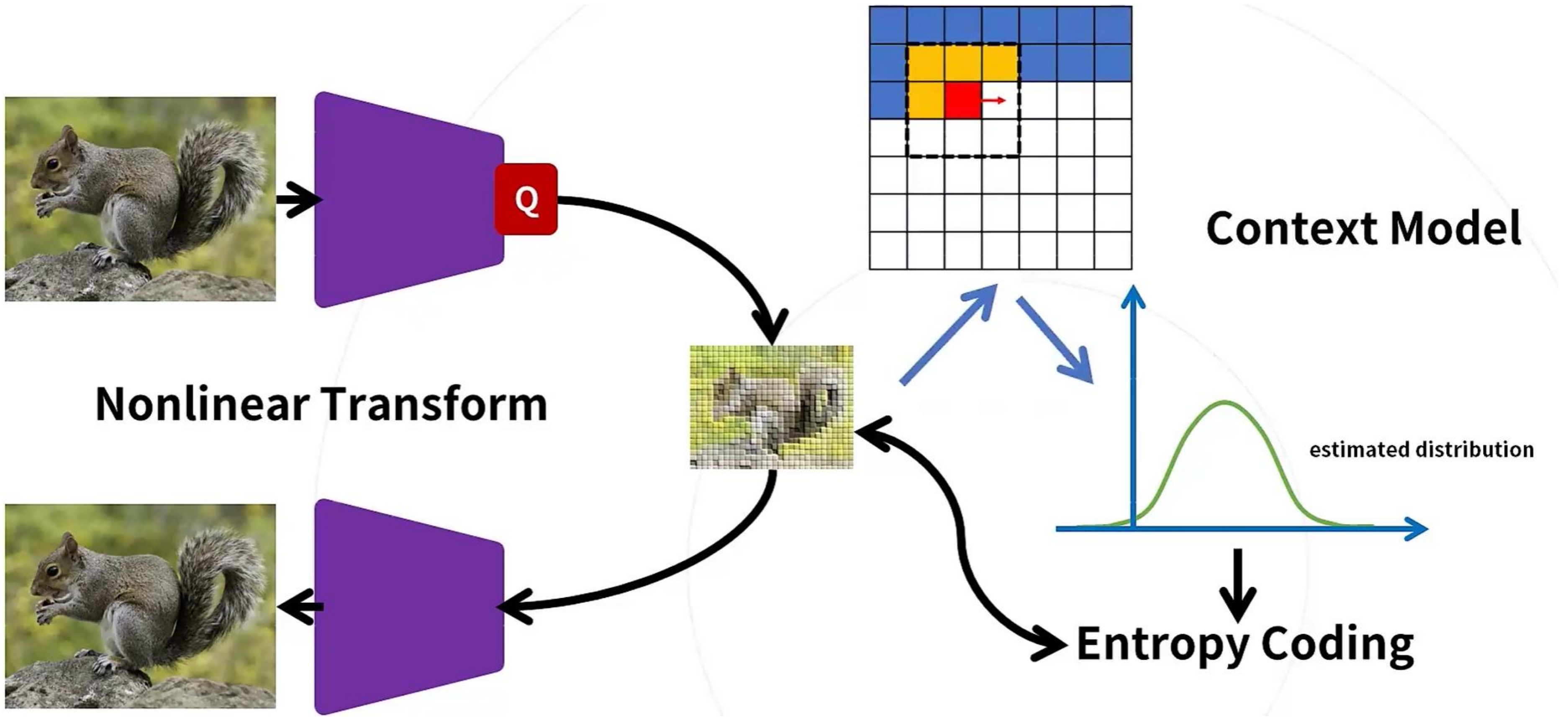
BACKGROUND: E2E Image Compression

Introducing autoregressive



BACKGROUND: E2E Image Compression

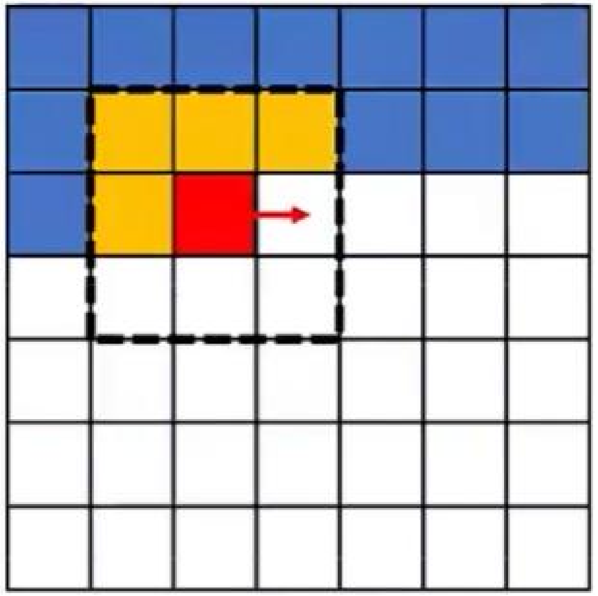
Introducing autoregressive

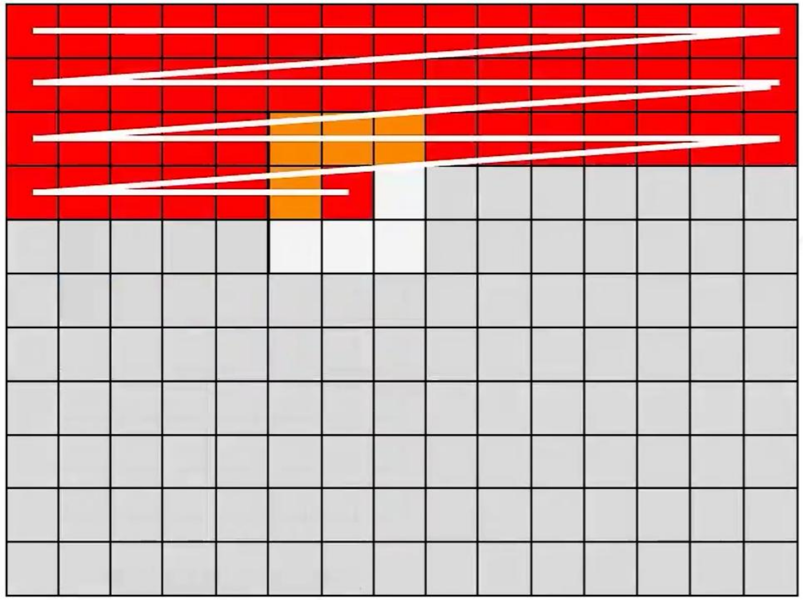


BACKGROUND: E2E Image Compression

Introducing autoregressive

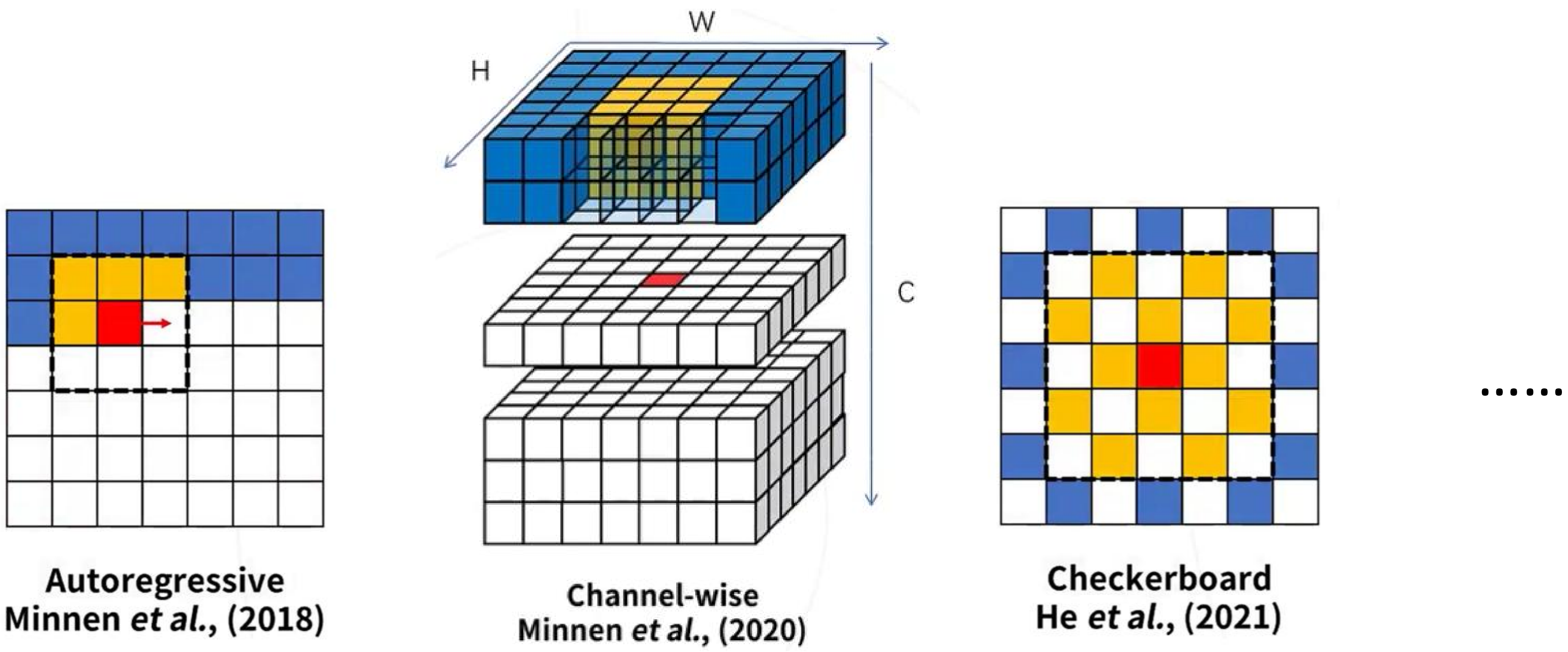
- More spatial context information
- No additional bitstream
- Serial coding, high time complexity O(HW)





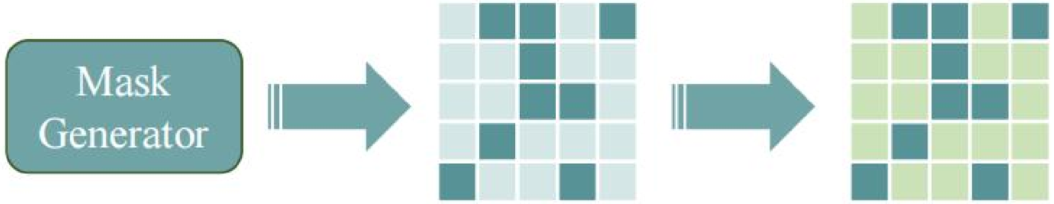
BACKGROUND: E2E Image Compression

Autoregressive acceleration

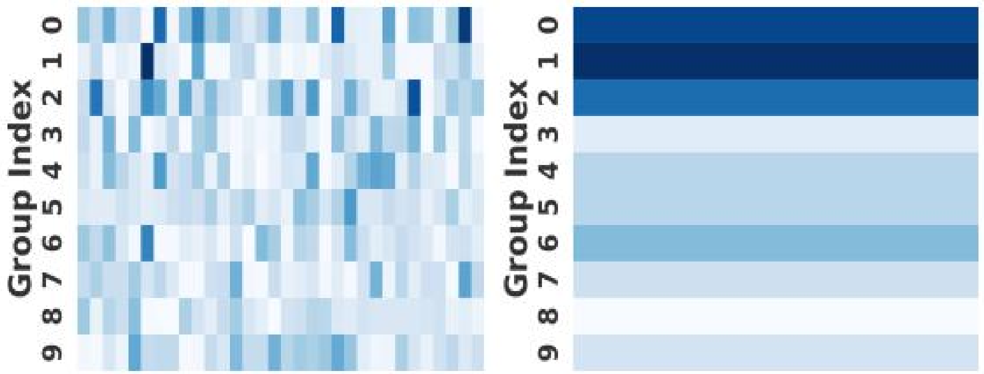


BACKGROUND: E2E Image Compression

Autoregressive acceleration



CAPEM
Li et al., (2023)



(a) Entropy of each channel. (b) Maximal entropy of each group.

ELIC
He et al., (2022)

……

OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

METHOD: Motivation

- Why do we use autoregressive models?

Spatial location independence of elements

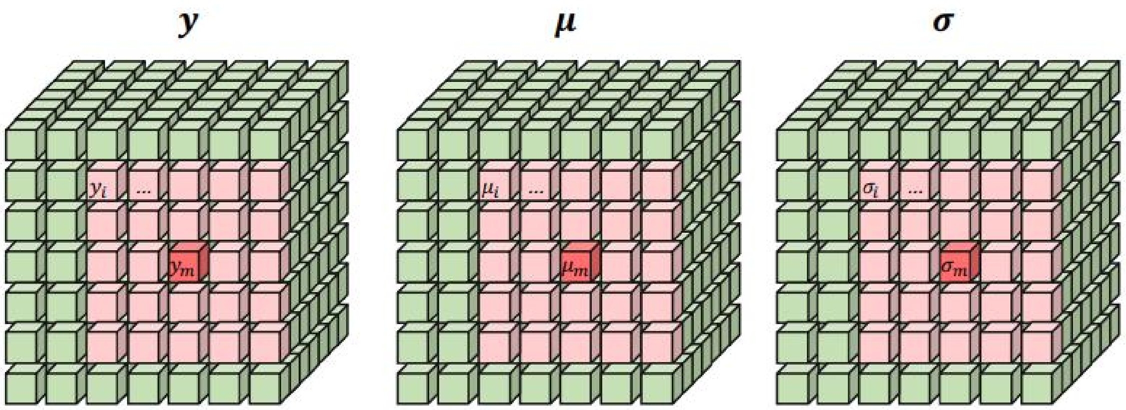
- Basic idea

Use correlation loss to reduce the spatial correlation

METHOD: Correlation Loss

- Calculate the correlation of each point

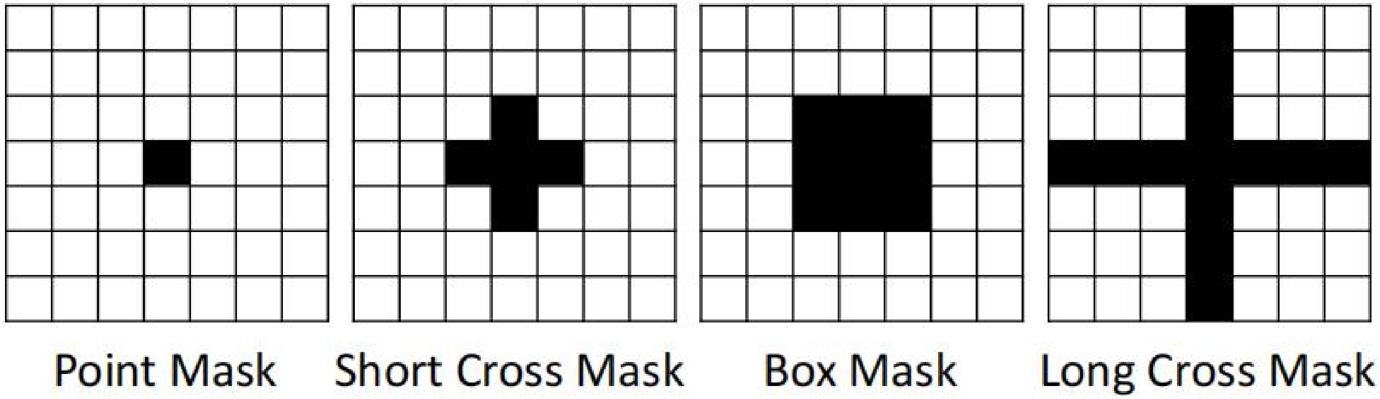
$$Corr\_Map_{k\times k}[i] = \mathbb{E}_{x\sim p(x)}\left[\left(\frac{y_i-\mu_i}{\sigma_i}\right)\left(\frac{y_m-\mu_m}{\sigma_m}\right)\right], 0\leq i<k^2$$



METHOD: Correlation Loss

- Apply masks to limit autocorrelation

$$Masked\_Map_{k\times k}[i] = Corr\_Map_{k\times k}[i] \odot Mask$$



- Calculate L2Loss as correlation loss

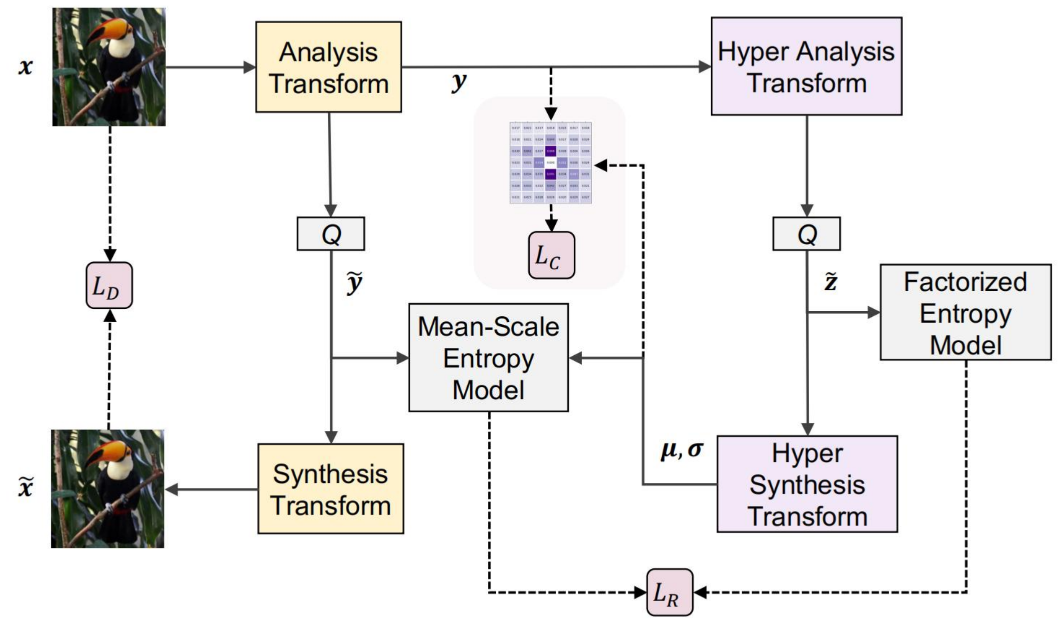
$$L_{corr} = ||Masked\_Map_{k\times k}[i]||^2$$

METHOD: Correlation Loss

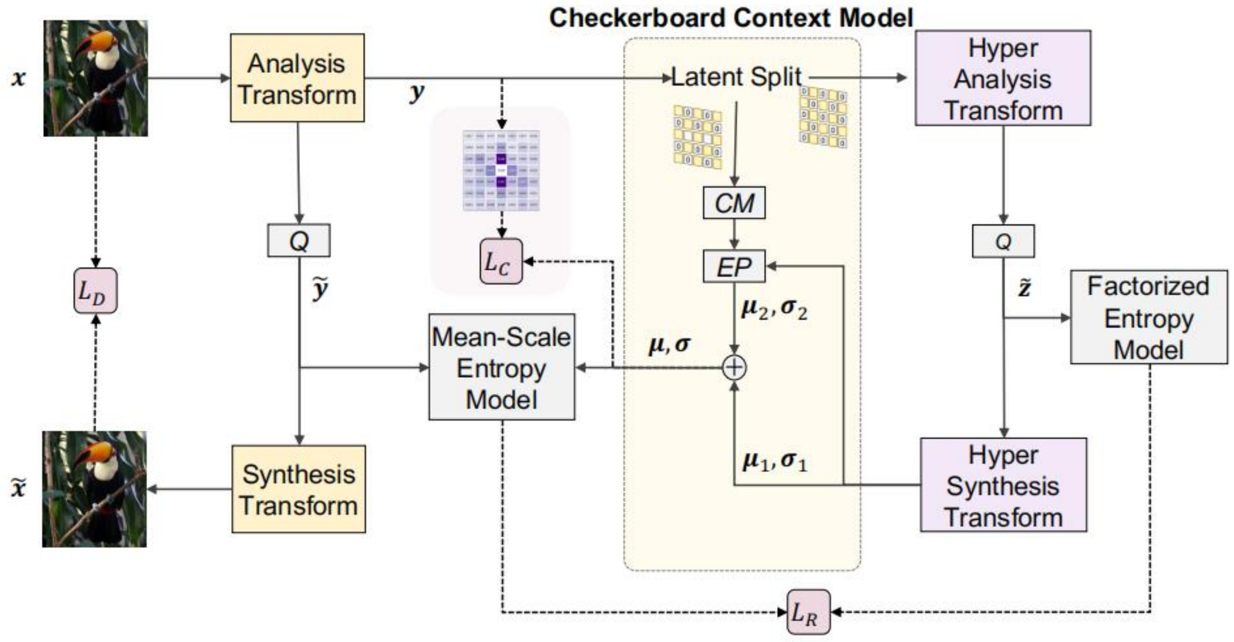
- Loss Function

$$RD_{loss} = E_{x \sim p_{(x)}} \left[-\log_2 p_{\hat{y}|\hat{z}}(\hat{y} \mid \hat{z}) - \log_2 p_{\hat{z}}(\hat{z}) \right] + \lambda \cdot E_{x \sim p_{(x)}} [d(x, \hat{x})] + \alpha \cdot [L_{corr}]$$

METHOD: Structure



Combine Hyperprior



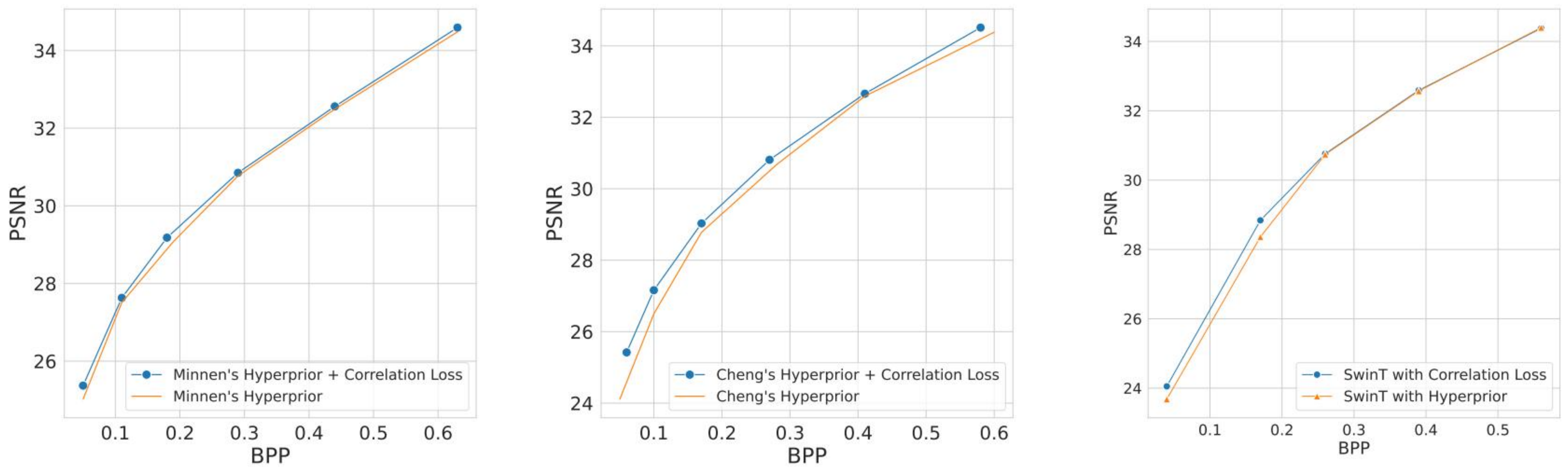
Combine Checkerboard

OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

EXPERIMENTS: RD-Performance

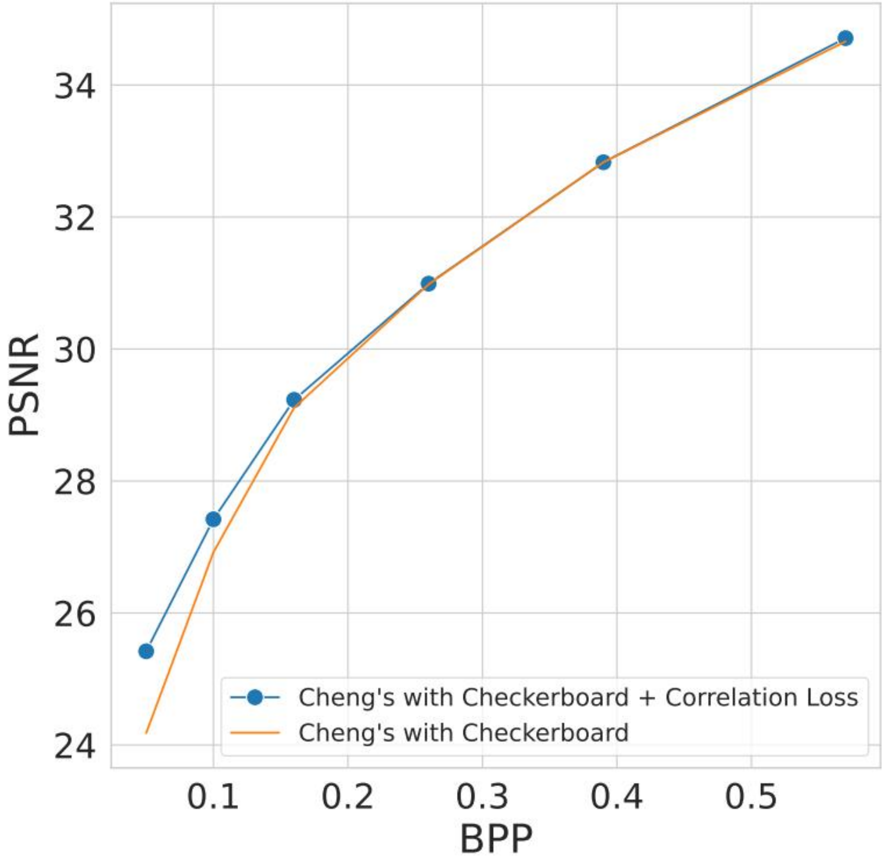
Combine with baseline methods

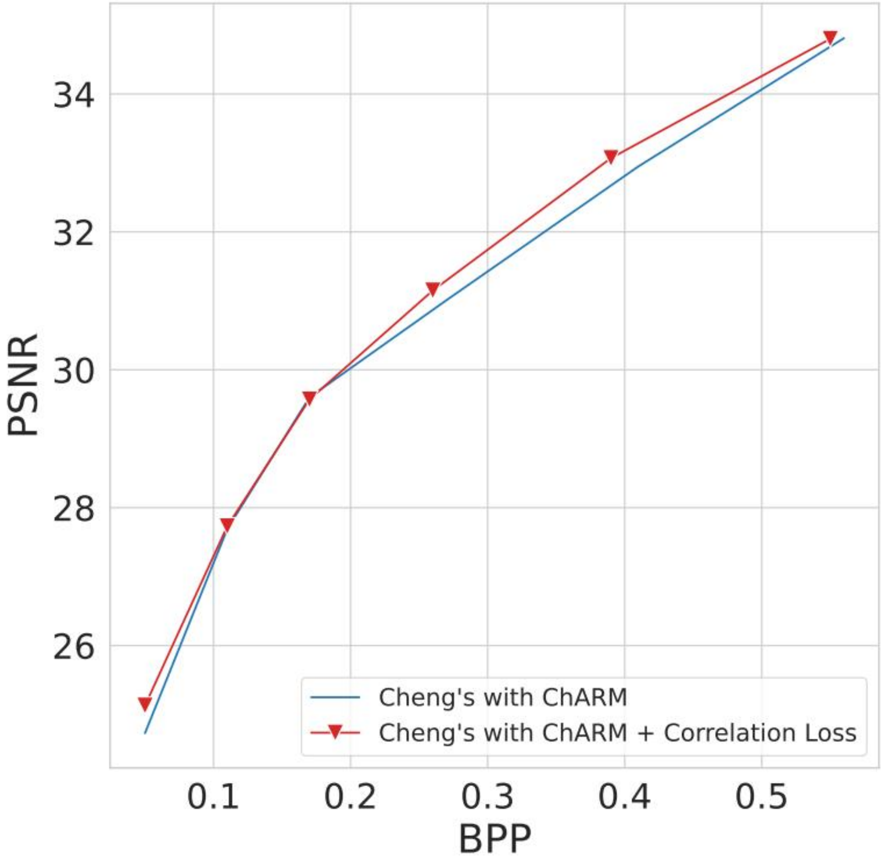


(a) Minnen's mean scale hyperprior (b) Cheng's mean scale hyperprior (c) SwinT mean scale hyperprior

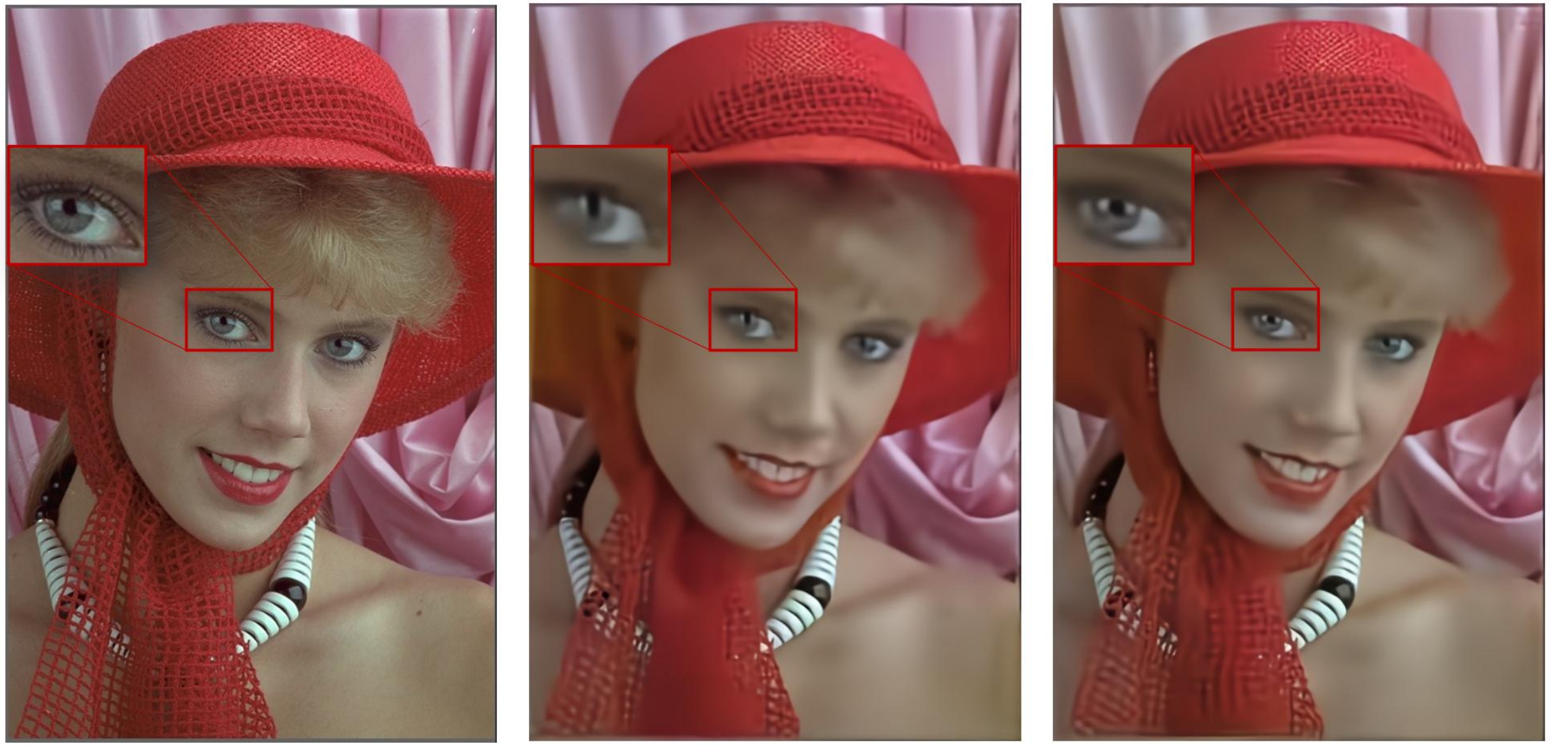
EXPERIMENTS: RD-Performance

Combine with autoregressive models





EXPERIMENTS: Reconstructed image quality

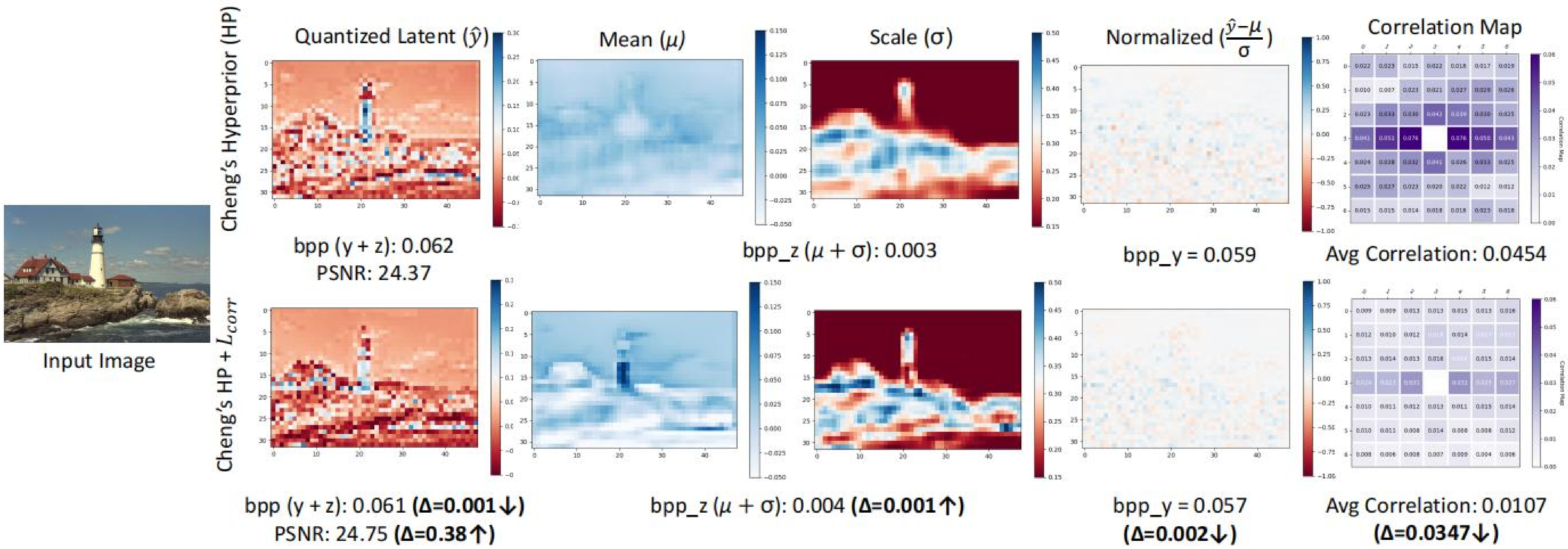


Original Image

Cheng's with Hyperprior
BPP: 0.06, **PSNR: 27.74dB**

Cheng's with Correlation loss
BPP: 0.06, **PSNR: 28.36dB**

EXPERIMENTS: Visualization

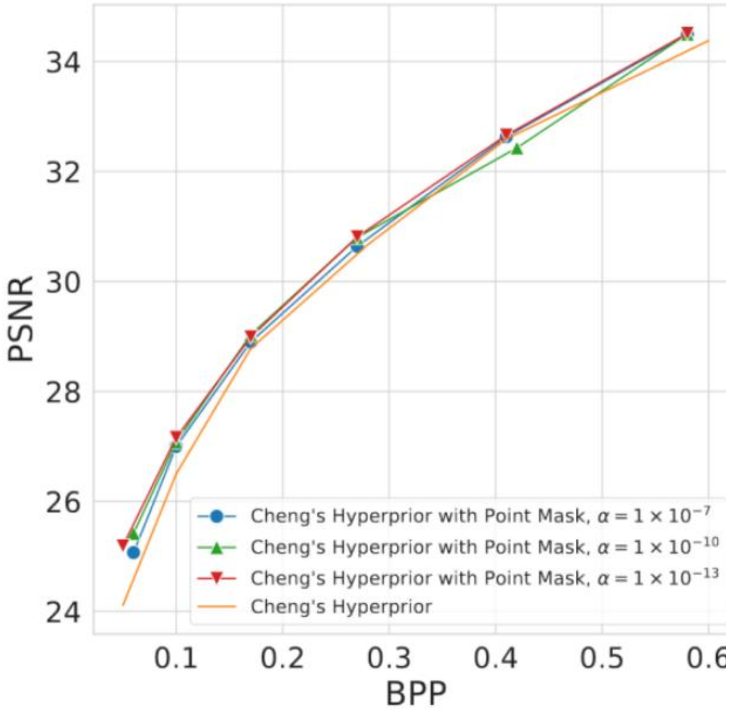


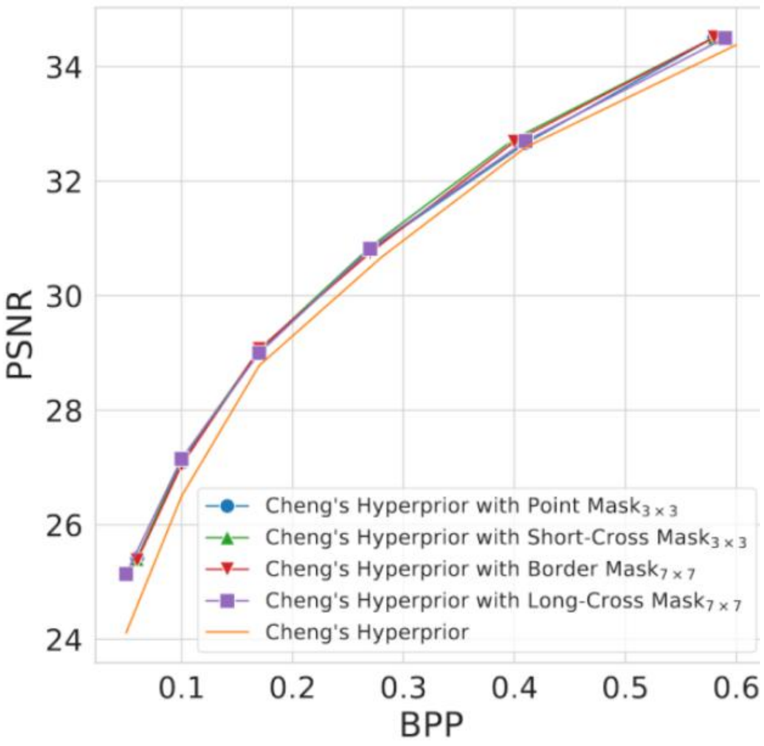
EXPERIMENTS: Complexity tradeoff

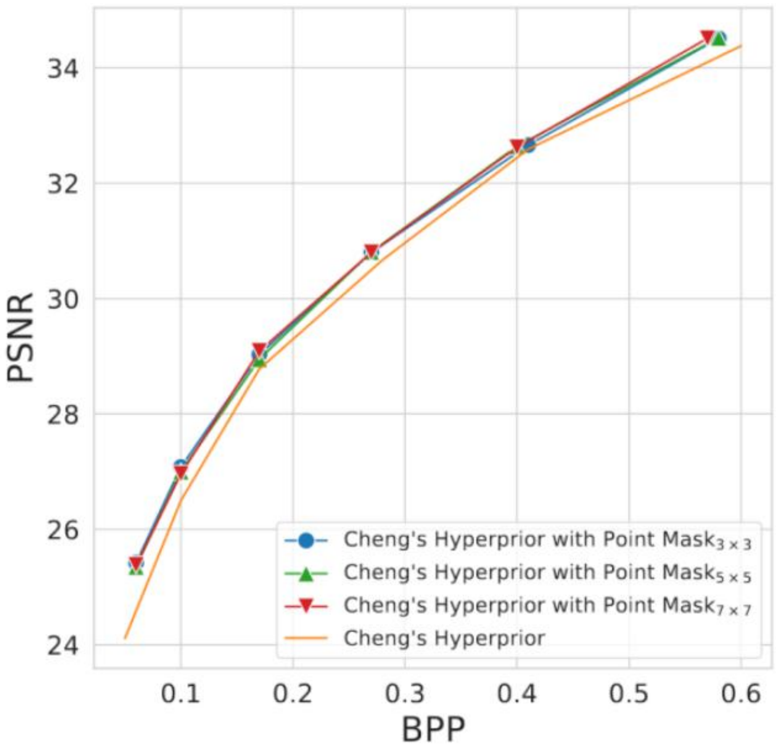
Architecture	BD Rate Gains (%)	Inference Time (sec)
Cheng's Hyperprior (CH)	—	4.66
CH + Correlation Loss (Proposed)	**9.5**	4.66
CH + Checkerboard	10.37	5.33
CH + Checkerboard + Correlation Loss (Proposed)	**16.50**	5.33
CH + ChARM	14.69	8.14
CH + ChARM + Correlation Loss (Proposed)	**17.99**	8.14
CH + AR Context	18.47	251.65

EXPERIMENTS: Ablation study

- Different α
- Different mask
- Different windows







OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

CONCLUSION

- Design a loss function to narrow the difference between the predicted distribution and real distribution

- Improves the performance of existing methods without loss

Thanks for listening!