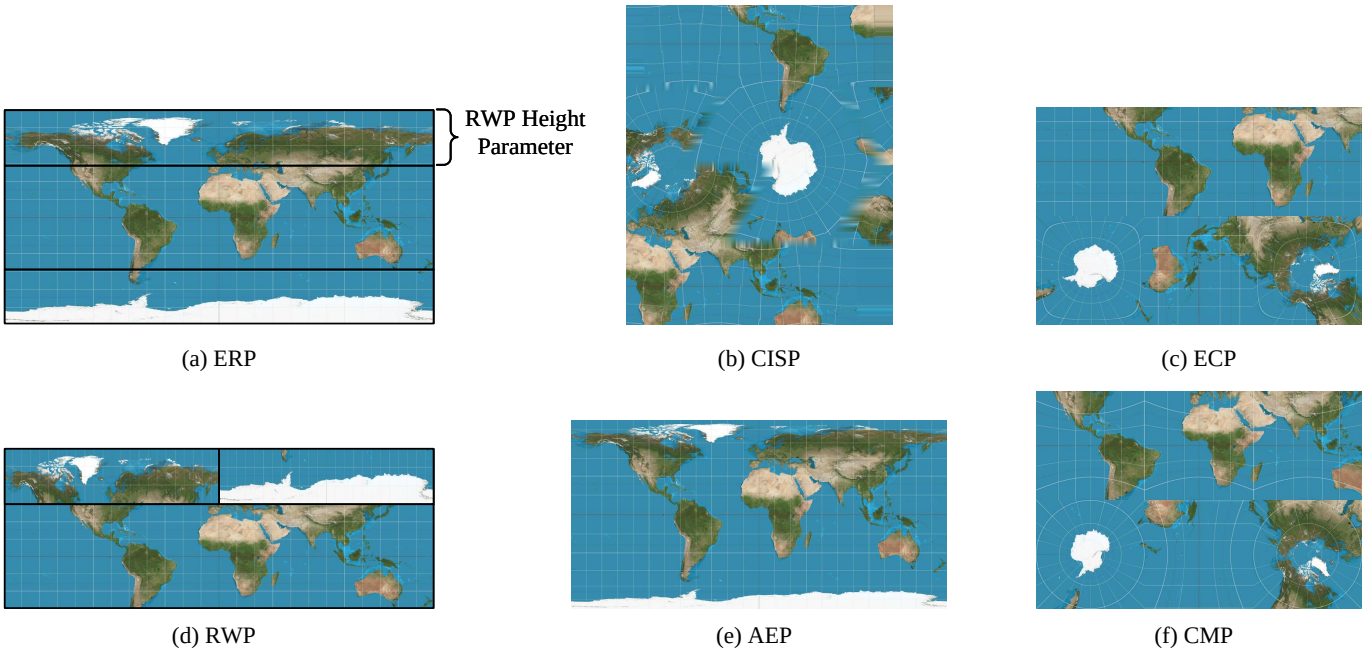
Learning Content-Weighted Pseudocylindrical Representation for 360° Image Compression: Supplemental Document



(a) ERP (b) CISP (c) ECP

(d) RWP (e) AEP (f) CMP

Fig. 1. Different projection methods of 360° images of the world map.

I. Experiments

A. Optimization for Existing Codecs

To broaden our comparative analysis and assess how techniques originally developed for traditional codecs adapt to learned image compression methods, we applied enhancements to seven codecs designed for central-perspective images, utilizing two strategies: regional resampling and reprojections. This selection encompasses four conventional codecs: JPEG [1], JPEG2000 [2], BPG [3] (representing HEVC intra coding), and VVC [4] (VVC intra coding). Additionally, three learned image compression methods were included: Ball'e18 [5], Minnen18 [6], and Cheng20 [7].

In our study, we implemented the Regional Wise Packing (RWP) strategy [8] for regional resampling. Fig. 1 (a) and (d) illustrate how RWP divides ERP images into three sections, using a parameter to regulate the height of the north and south pole regions. The reassembled image then serves as the input for compression. Although RWP mitigates non-uniform sampling to an extent, its simplistic split-and-merge approach may disrupt image continuity and context, potentially affecting compression performance adversely.

We assessed a range of height parameters for RWP, $\{8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 100\}$, across four compression standards (JPEG, JPEG2000, BPG, VVC) using BD-V-PSNR and BD-BR metrics, with the original method without RWP as the baseline. Our findings indicate that RWP enhances compression efficiency for all codecs. For instance, it yielded a 0.76 dB improvement and saved 12.94% bits for JPEG. The optimal heights for JPEG, JPEG2000, BPG, and VVC were found to be 64, 40, 48, and 48 respectively. For DNN-based methods, a height of 48 was selected, demonstrating satisfactory performance across all four standards. As Table I shows, RWP had a negligible impact on learned image compression methods, slightly benefiting Ballé18 and Cheng20, but impeding Minnen18's performance.

Regarding reprojections, we incorporated four types: adjusted equal-area projection (AEP), cubemap projection (CMP), equatorial cylindrical projection (ECP), and compact icosahedron projection (CISP). We reprojected ERP images into these formats[1], compressed them, and then converted them back to viewports for evaluation. As Fig. 2 depicts, reprojection seems to

[1]Reprojections were executed using default settings in 360Lib-9.1, with face height and width in the coding set to 0 for automatic parameter calculation. FFmpeg was used for YUV-RGB transformations.

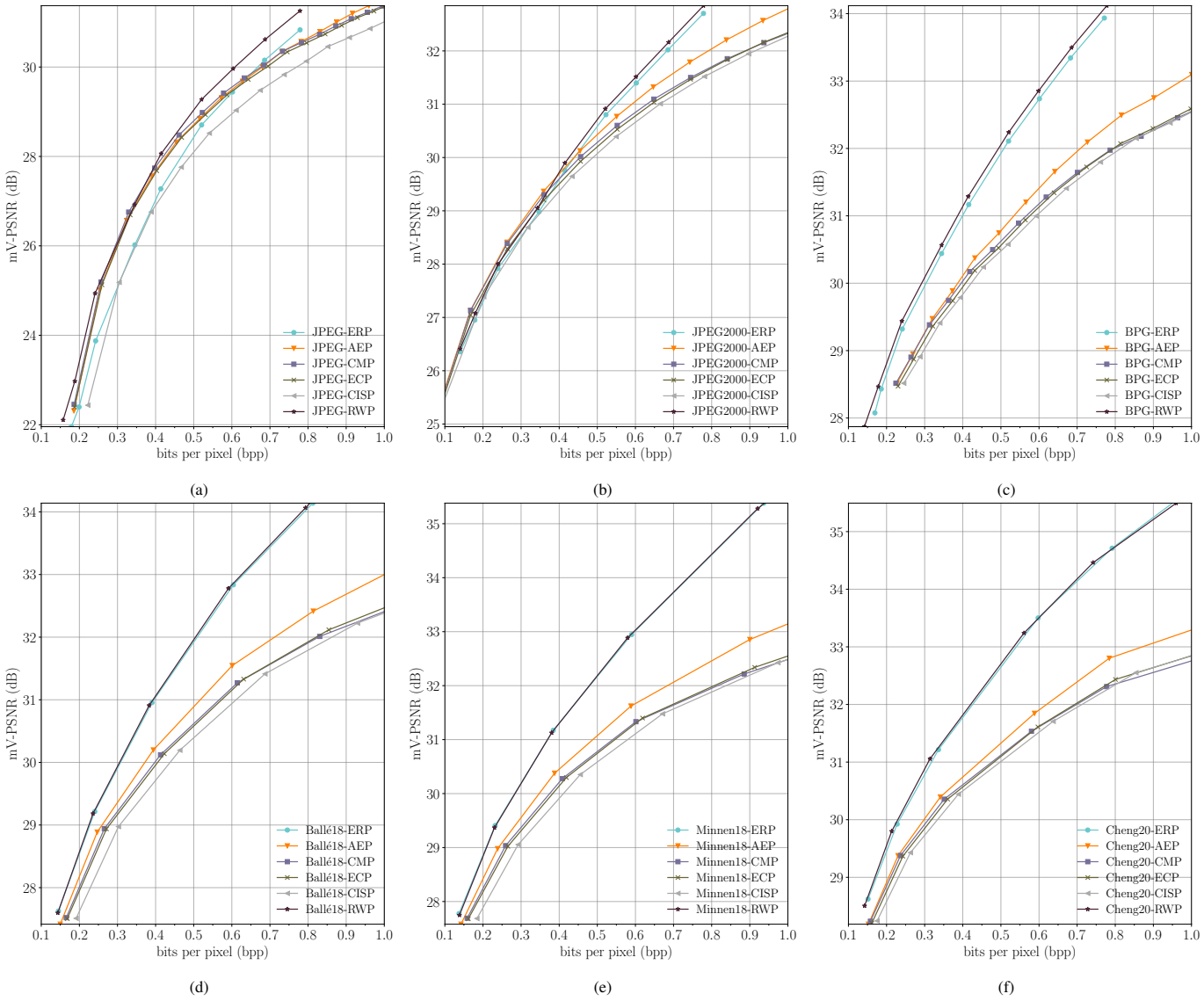


Fig. 2. Rate-mV-PSNR curves of different compression methods. (a) JPEG. (b) JPEG2000. (c) BPG. (d) Ballé18. (e) Minnen18. (f) Cheng20.

improve the performance of traditional codecs like JPEG and JPEG2000 at lower bitrates, but slightly hampers performance at higher bitrates. For learned image compression codecs, no significant enhancements were observed in viewport-based metrics, likely due to information loss during frequent interpolations and image content reorganization in projections.

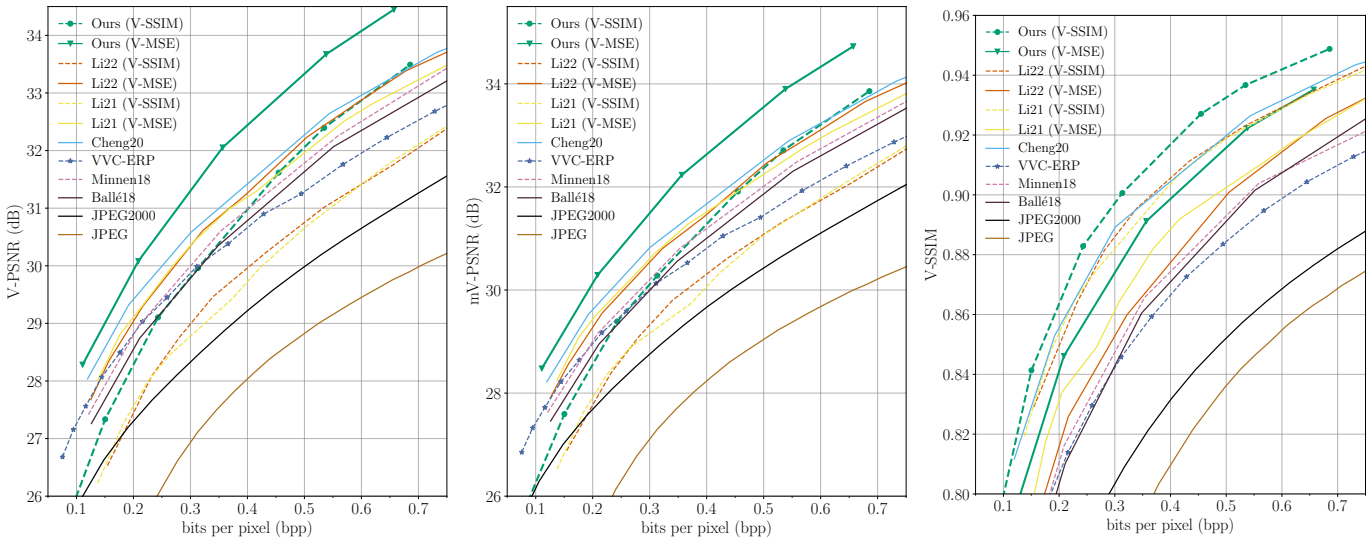
In Table I, we used ERP results as the baseline to calculate BD-V-PSNR, BD-mV-PSNR, and BD-RATE metrics for each codec. Among the tested strategies, RWP demonstrated superior performance, largely due to its hyperparameter optimization tailored to the joint rate-distortion objective.

It's important to note, however, that while RWP significantly aids traditional codecs like JPEG, JPEG2000, and BPG in compressing 360° images, its impact on DNN-based image compression methods is less clear-cut. For instance, RWP slightly improved Ballé18 by 0.02 dB, yet negatively affected Minnen18 by the same margin. Reprojection methods similarly showed a more marked performance decline in learned image compression methods compared to traditional codecs. This difference could be attributed to how RWP and reprojections disrupt image context and create discontinuities, posing challenges to the flexibility of learned compression techniques.

These findings underscore the necessity for a specialized deep-learning compression approach, such as the pseudocylindrical representation and convolution detailed in this paper. Such an approach should not only address the unbalanced sampling issue but also maintain image context and continuity, a crucial aspect where traditional resampling and reprojection methods fall short in the context of learned 360° image compression.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT PROJECTION METHODS IN TERMS OF BD-V-PSNR, BD-MV-PSNR, BD-V-SSIM, AND BD-BR

Method	RATE-V-PSNR		RATE-mV-PSNR		RATE-V-SSIM	
	BD-V-PSNR (dB) ↑	BD-BR (%) ↓	BD-mV-PSNR (dB) ↑	BD-BR (%) ↓	BD-V-SSIM ↑	BD-BR (%) ↓
JPEG-ERP as the anchor method						
JPEG-AEP	0.29	−5.51	0.46	−8.19	0.012	−6.17
JPEG-CISP	−0.51	9.83	−0.33	6.45	−0.015	9.17
JPEG-CMP	0.36	−6.37	0.57	−9.14	0.017	−8.27
JPEG-ECP	0.23	−4.06	0.46	−7.31	0.011	−5.17
JPEG-RWP	0.76	−12.94	0.80	−12.78	0.024	−12.70
JPEG2000-ERP as the anchor method						
JPEG2000-AEP	−0.62	23.13	0.06	−0.61	−0.005	5.73
JPEG2000-CISP	−0.90	36.69	−0.27	11.32	−0.013	15.29
JPEG2000-CMP	−0.66	28.57	−0.03	4.08	−0.004	4.31
JPEG2000-ECP	−0.77	32.20	−0.12	6.51	−0.008	9.77
JPEG2000-RWP	0.13	−3.52	0.11	−2.86	0.003	−2.85
BPG-ERP as the anchor method						
BPG-AEP	−0.51	17.88	−1.01	37.28	−0.005	6.25
BPG-CISP	−0.89	37.03	−1.34	54.36	−0.012	16.02
BPG-CMP	−0.70	30.72	−1.13	45.74	−0.004	5.28
BPG-ECP	−0.78	33.51	−1.21	48.88	−0.008	10.99
BPG-RWP	0.15	−3.93	0.14	−3.73	0.004	−4.53
Ballé18-ERP as the anchor method						
Ballé18-AEP	−1.14	26.35	−1.35	32.77	−0.006	7.87
Ballé18-CISP	−1.92	49.67	−2.09	56.23	−0.021	31.13
Ballé18-CMP	−1.58	35.96	−1.73	42.01	−0.012	14.52
Ballé18-ECP	−1.66	39.65	−1.79	44.60	−0.014	19.12
Ballé18-RWP	0.02	−0.52	0.03	−0.63	0.001	−0.80
Minnen18-ERP as the anchor method						
Minnen18-AEP	−1.27	29.21	−1.45	36.31	−0.007	9.17
Minnen18-CISP	−2.08	53.58	−2.21	61.03	−0.022	32.83
Minnen18-CMP	−1.76	40.24	−1.87	47.26	−0.013	16.73
Minnen18-ECP	−1.84	44.14	−1.94	50.18	−0.015	21.48
Minnen18-RWP	−0.02	0.38	−0.01	0.26	−0.000	0.16
Cheng20-ERP as the anchor method						
Cheng20-AEP	−0.89	28.10	−1.17	36.96	−0.006	8.68
Cheng20-CISP	−1.31	44.93	−1.58	53.98	−0.016	23.59
Cheng20-CMP	−1.15	35.52	−1.39	44.07	−0.008	10.68
Cheng20-ECP	−1.21	39.58	−1.44	47.06	−0.012	16.35
Cheng20-RWP	0.04	−1.11	0.05	−1.35	0.002	−2.72
VVC-ERP as the anchor method						
VVC-AEP	−0.16	3.95	−0.10	2.27	−0.000	0.20
VVC-CISP	−0.77	21.74	−0.61	17.13	−0.010	12.99
VVC-CMP	−0.58	14.59	−0.40	9.74	−0.002	1.80
VVC-ECP	−0.65	17.19	−0.47	12.20	−0.006	7.22
VVC-RWP	0.12	−2.98	0.13	−3.30	0.003	−4.11



(a) (b) (c)

Fig. 3. Rate-distortion curves of different compression methods on LIC3602K. (a) V-PSNR. (b) mV-PSNR. (c) V-SSIM.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT COMPRESSION METHODS IN TERMS OF BD-V-PSNR, BD-MV-PSNR, BD-V-SSIM, AND BD-BR ON LIC3602K. CHENG20 IS THE ANCHOR METHOD

Method	RATE-V-PSNR		RATE-mV-PSNR		RATE-V-SSIM	
	BD-V-PSNR (dB) ↑	BD-BR (%) ↓	BD-mV-PSNR (dB) ↑	BD-BR (%) ↓	BD-V-SSIM ↑	BD-BR (%) ↓
JPEG	−3.97	187.52	−4.043	186.61	−0.100	124.95
JPEG2000	−2.11	81.64	−1.937	71.50	−0.051	79.40
HEVC	−1.18	45.59	−1.302	50.39	−0.030	44.78
Ballé18	−0.61	19.22	−0.640	19.28	−0.013	16.06
Minnen18	−0.39	12.32	−0.435	13.17	−0.009	11.91
Cheng20	0.07	−2.23	0.075	−2.25	0.025	−27.98
Li21 (V-MSE)	−0.23	7.51	−0.181	5.79	0.002	−2.24
Li21 (V-SSIM)	−1.69	64.42	−1.566	57.51	0.022	−23.99
Li22 (V-MSE)	−0.18	5.58	−0.195	5.89	−0.006	6.71
Li22 (V-SSIM)	−1.58	58.67	−1.489	53.63	0.022	−24.20
Ours (V-MSE)	0.83	−22.84	0.809	−21.82	0.017	−18.08
Ours (V-SSIM)	−0.67	17.38	−0.599	15.04	0.037	−37.05

B. Quantitative Evaluation

We expanded our testing to include an additional dataset, LIC3602K, consisting of 100 ERP images with a resolution of $1,024 \times 2,048$. The results of this test are detailed in Fig.3 and TableII, using VVC as the anchor. Remarkably, our methods optimized for V-MSE and V-SSIM continue to outperform others, particularly in the V-PSNR and V-SSIM metrics. It's noteworthy that our method demonstrates consistently high performance even when applied to images of varying resolutions. This consistency highlights the method's adaptability across diverse image resolutions.

C. Qualitative Evaluation

We provide additional visual comparisons of our method, optimized for V-SSIM, against VVC, Li22, Li21, and Cheng20 at comparable bitrates in Fig. 4 and Fig. 5. As discussed in the manuscript, Cheng20 and VVC excel in preserving sharp edges but tend to blur textures. Conversely, Li21 and Li22 effectively retain texture details but struggle with maintaining sharp edges. Our method combines the strengths of both approaches. It successfully preserves both sharp edges and detailed textures, showcasing a comprehensive enhancement in image quality.

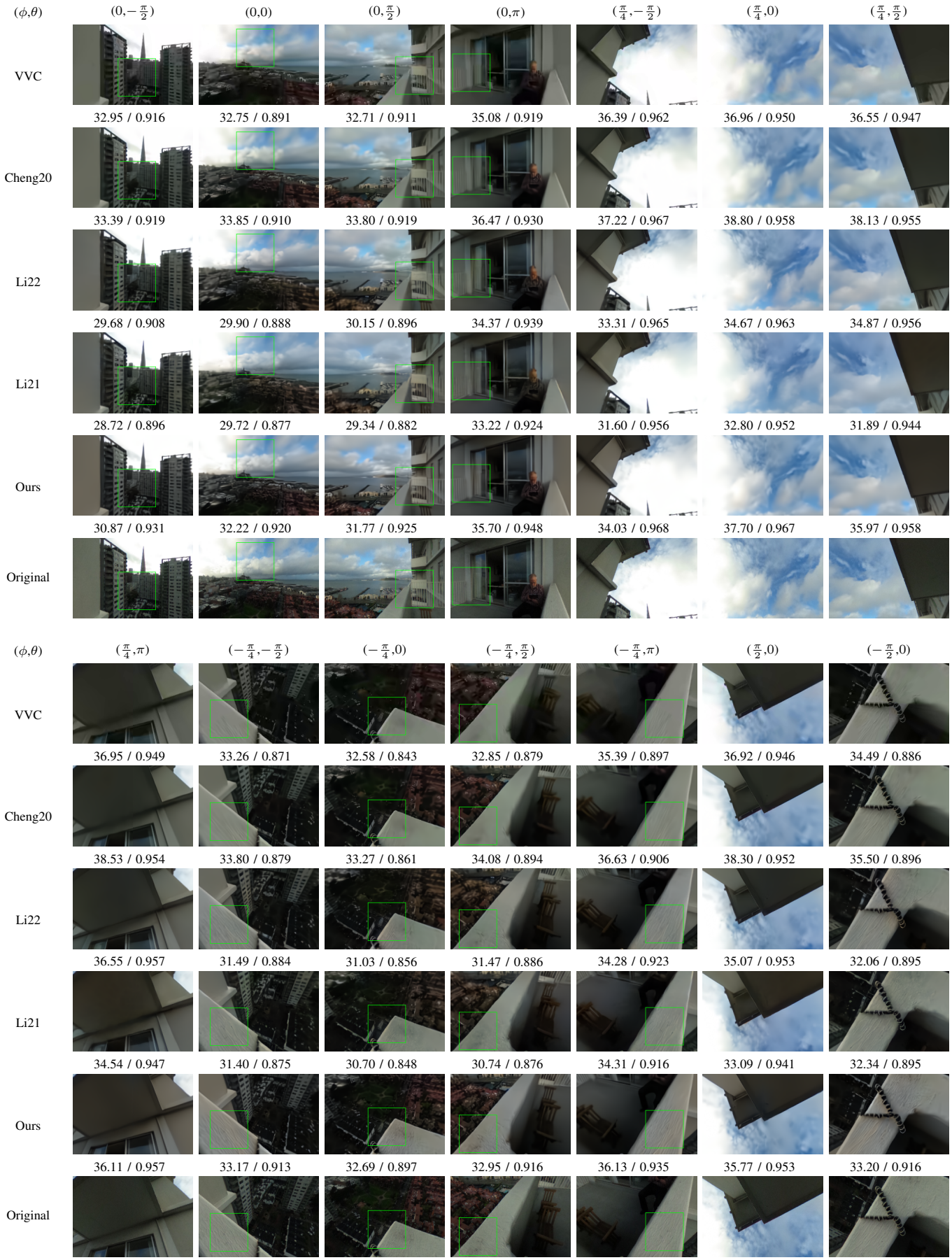


Fig. 4. 14 viewports indexed by (ϕ, θ) at different visual angles of the ERP images produced by VVC, Cheng20, Li22, Li21 and Ours. We provide the distortion in form of PSNR (dB) / SSIM under each viewport. The bitrates of the ERP images produced by VVC, Cheng20, Li22, Li21 and Ours are separately 0.124bpp, 0.123bpp, 0.116bpp, 0.110bpp, and 0.120bpp.

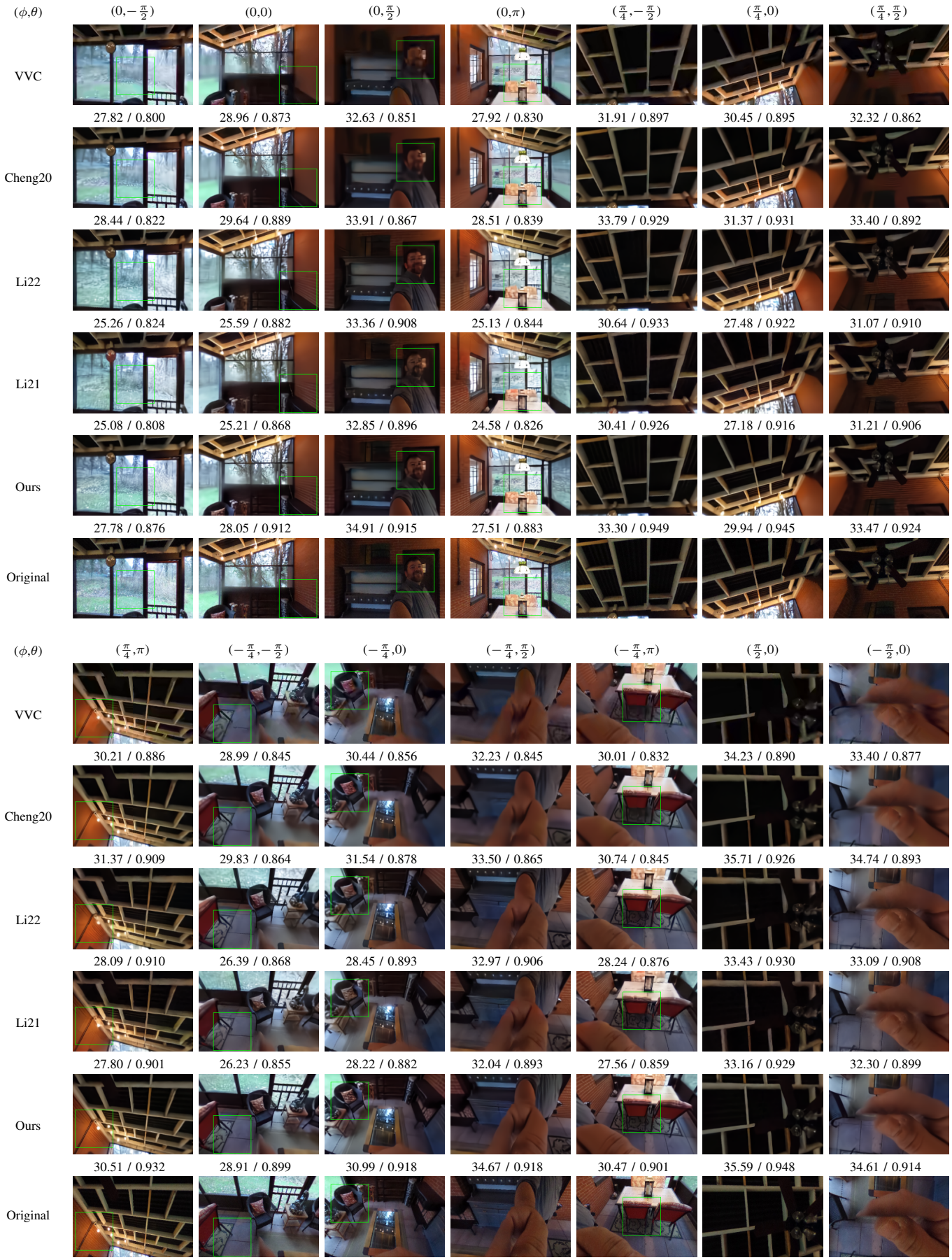
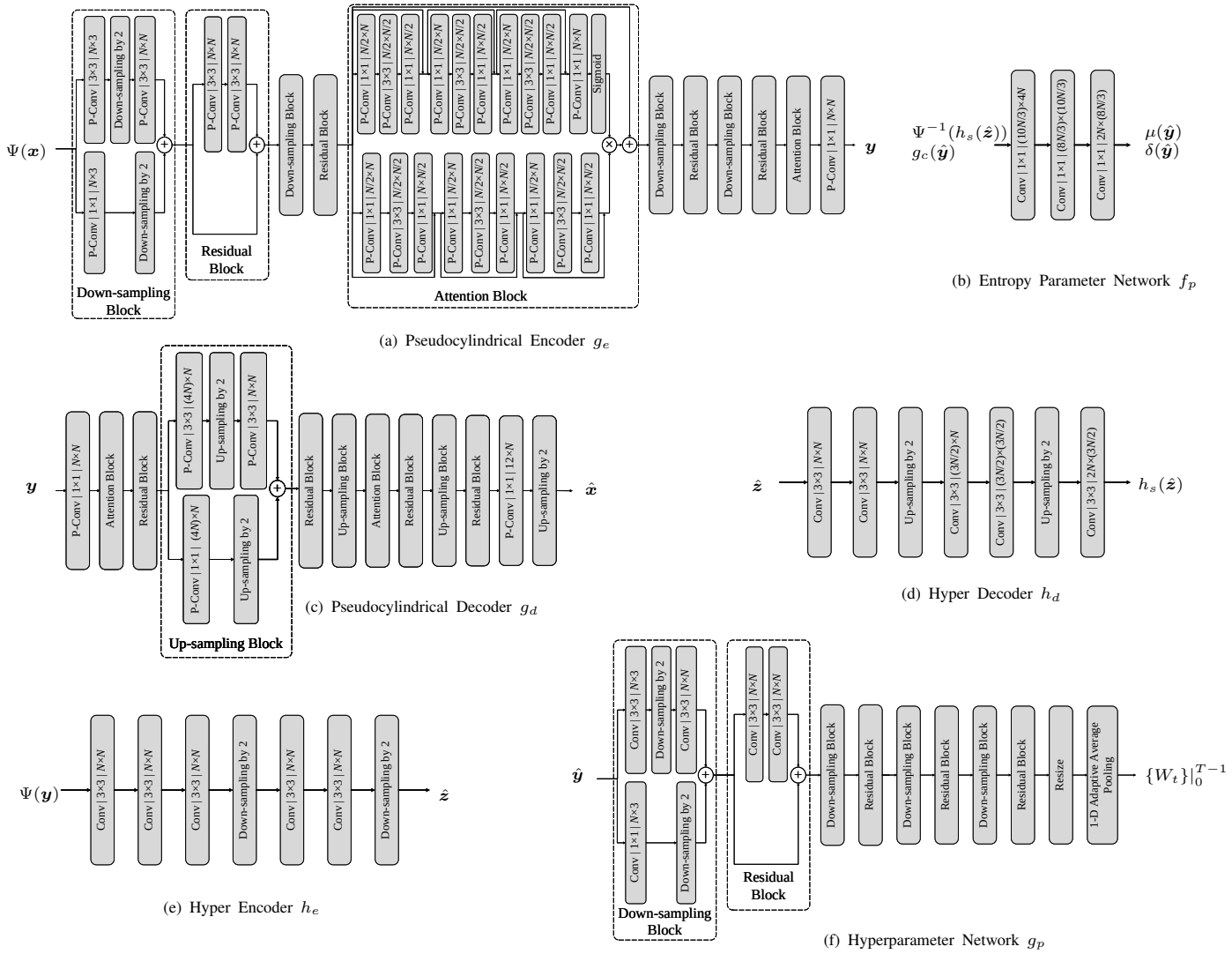


Fig. 5. 14 viewports indexed by (ϕ, θ) at different visual angles of the ERP images produced by VVC, Cheng20, Li22, Li21 and Ours. We provide the distortion in form of PSNR (dB) / SSIM under each viewport. The bitrates of the ERP images produced by VVC, Cheng20, Li22, Li21 and Ours are separately 0.148bpp, 0.139bpp, 0.143bpp, 0.135bpp, and 0.156bpp.

II. Details of Network Structure

We offer the network structure of the proposed learned omnidirectional image compression framework in Fig. 6.



(a) Pseudocylindrical Encoder g_e

(b) Entropy Parameter Network f_p

(c) Pseudocylindrical Decoder g_d

(d) Hyper Decoder h_d

(e) Hyper Encoder h_e

(f) Hyperparameter Network g_p

Fig. 6. Network structure of the proposed method. P-Conv: proposed pseudocylindrical convolution with filter support ($S \times S$) and the number of channels (output×input). Conv: standard convolution with filter support ($S \times S$) and number of channels (output×input). Masked P-Conv: proposed pseudocylindrical convolution with the mask defined by Minnen *et al*. [6]. N is the hyperparameter for the channels.

The hyperparameter network g_p (see Fig. 6 (f)) takes the ERP image $\boldsymbol{x}$ as input and predicts the hyperparameter of the pseudocylindrical representation. g_p consists of four down-sampling blocks with each down-sampling block followed by a residual block. The extracted feature is then resized as a 2D vector with T rows. Finally, a final 1D adaptive average pooling is introduced to produce the hyperparameter $\{W_t\}_0^{T-1}$.

The pseudocylindrical encoder g_e (see Fig. 6 (a)) takes the parametric pseudocylindrical representation of the ERP image $\Psi(\boldsymbol{x})$ as input and produces the pseudocylindrical code representation $\boldsymbol{y}$. g_e is made of four down-sampling blocks, four residual blocks, two attention blocks, a back-end pseudocylindrical convolution, and a sigmoid layer. The down-sampling block processes and down-samples the pseudocylindrical feature maps by a factor of two. The residual block has two convolution layers with a skip connection, following each down-sampling block. A simplified attention block [7] is added right after the second and fourth residual blocks to increase the model capacity and expand the receptive field. A final convolution layer with N filters is used to produce the code representation $\boldsymbol{y}$. Finally, we produce the discrete code $\hat{\boldsymbol{y}}$ with a uniform quantizer.

The pseudocylindrical decoder g_d (see Fig. 6 (c)) is a mirror of the analysis transform where the up-sampling blocks replace the down-sampling blocks. Instead of performing deconvolution for upsampling, we expand the feature representation by a factor of four in the channel dimension and reshape it such that the height and width grow by a factor of two [9], [10]. The last pseudocylindrical convolution contains 12 filters followed by an upsampling operation to reconstruct the pseudocylindrical representation, which is then transformed back to an ERP image. Generalized divisive normalization (GDN) and inverse GDN [11] are separately adopted after the last convolution of the down-sampling and up-sampling blocks. Unless stated

otherwise, the Leaky rectified linear unit (Leaky ReLU) is used as the nonlinear activation function for other convolution layers.

As for the entropy network, we model the probability distributions of the quantized code $\hat{\boldsymbol{y}}$ as Gaussian, whose estimation can be enhanced by considering the code context (also referred to as the auto-regressive prior) and the hyper-prior. Specifically, we apply a masked pseudocylindrical convolution layer g_c to the quantized code to generate the auto-regressive prior[2].

The hyper-prior is established through a combination of a hyper encoder h_e and a hyper decoder h_d. h_e reduces the dimensionality of the code representation $\boldsymbol{y}$ by a factor of 4, resulting in a hyper code $\boldsymbol{z}$. This hyper code is subsequently quantized to $\hat{\boldsymbol{z}}$ and upsampled back to the code space using h_e. The hyper encoder h_e (see Fig. 6 (d)) comprises five 3×3 convolution layers with two down-sampling operations after the third and fifth convolutions. The hyper decoder h_d (see Fig. 6 (e)) involves two convolution layers, an up-sampling operation, two more convolution layers, another up-sampling operation, and finally, a back-end convolution layer to generate the hyper-prior $h_s(\hat{\boldsymbol{z}})$. The height of the hyper code is given by $H/(64T)$, which could be less than 1 in case of more tiles in the pseudocylindrical representation (*i.e.*, a larger T) or a lower-resolution ERP image (*i.e.*, a smaller H). To mitigate this, we merge the tiles by projecting the pseudocylindrical representation to ERP during the hyper-prior modeling, and convert the output hyper-prior back to the pseudocylindrical representation.

Lastly, we combine the hyper-prior and the auto-regressive prior, and feed them to a parameter net f_p (see Fig. 6 (b)) composed of three 1×1 convolution layers. f_p generates both the Gausasian mean $\mu(\hat{\boldsymbol{y}})$ and variance $\sigma(\hat{\boldsymbol{y}})$ estimates.

REFERENCES

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18–34, 1992.

[2] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[3] F. Bellard, "BPG image format," 2019. [Online]. Available: https://bellard.org/bpg/

[4] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, "Versatile video coding (draft 5)," *Joint Video Experts Team (JVET) of ITU-T SG*, vol. 16, pp. 3–12, 2019.

[5] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference Learning Representations*, 2018.

[6] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, vol. 31, 2018, p. 10794–10803.

[7] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7939–7948.

[8] J. Boyce, A. Ramasubramanian, R. Skupin, G. J. Sullivan, A. Tourapis, and Y. Wang, "HEVC additional supplemental enhancement information (draft 4)," *Joint Collaborative Team on Video Coding of ITU-T SG*, vol. 16, pp. 12–20, 2017.

[9] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.

[10] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[11] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference Learning Representations*, 2017.

[2]When employing pseudocylindrical convolution, it's essential to adhere to the raster coding order, necessitating careful handling of linear interpolation and pseudocylindrical padding.