Learned Image Compression with Mixed Transformer-CNN Architectures (Supplementary Materials)

1. Classical Image Compression Standard Setting

1.1. VVC

We use VTM-12.1 which is built form the website[1] to achieve VVC. The script from CompressAI [2] is utilized to evaluate model. The command is as the following:

```
python -m compressai.utils.bench vtm [path of image folder];
-c [path of VTM folder]/cfg/encoder_intra_vtm.cfg
-b [path of VTM folder]/bin
-q 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40
```

1.2. WebP

We use the API of Pillow (PIL) to achieve WebP algorithm. The code is:

```
img.save(REC_WEBP, 'webp', quality=quality)
```

where quality is set as {5,10,15,20,25,30,35,40,45,50}.

2. Detailed Network Architecture

The architecture of the our method is shown in Fig. 1. The head dimensions of TCM blocks in g_a and g_s are set as {8, 16, 32, 32, 16, 8}, while the head dimensions of TCM blocks in h_a and h_s are set as 32. We set channel numbers C of TCM blocks as 128/192/256 for Small/Medium/Large model. RBS and RBU have the same architectures as in [4]. The numbers of channels of the middle convolutional layers in RBS and RBU are 64/96/128 for our Small/Medium/Large model, while the number of last layer of RBS and RBU is 128/192/256. Here, to achieve the balance between running speed and RD-performance, we reduce the slices number in [8] from 10 to 5. Therefore, we have 5 Channel-Conditional Parameter Nets with SWAtten to get $\{\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \boldsymbol{\mu_3}, \boldsymbol{\mu_4}\}$ and $\{\boldsymbol{\sigma_0}, \boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \boldsymbol{\sigma_3}, \boldsymbol{\sigma_4}\}$. Also, we have 5 Latent Residual Prediction to get $\{\overline{\boldsymbol{y}}_0, \overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \overline{\boldsymbol{y}}_3, \overline{\boldsymbol{y}}_4\}$. All the restored slices are concatenated as $\overline{\boldsymbol{y}}$ which is sent to decoder g_s to get a decompressed image.

[1] https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/releases/VTM-12.1

[2] https://github.com/InterDigitalInc/CompressAI/tree/master/compressai/utils/bench

Table 1. BD-rate improvements against the VVC anchor. For different datasets, the anchor is recalculated based the corresponding dataset. Lower BD-rate represents higher performance.

Methods	Dataset	BD-Rate
Cheng *et al.* [4]	Kodak 768x512	3.16
Xie *et al.* [9]		-1.65
Chen *et al.* [3]		-6.21
He *et al.* [5]		-7.49
Ours (Large)		**-12.30**
Ours (Medium)		**-9.65**
Ours (Small)		**-7.39**
Ballé *et al.* [1]	Tecnick 1200x1200	30.66
Xie *et al.* [9]		-4.07
Kim *et al.* [6]		6.98
Ours (Large)		**-13.71**
Ours (Medium)		**-11.29**
Ours (Small)		**-9.53**
Cheng *et al.* [4]	CLIC-P val 2K	6.77
Xie *et al.* [9]		-2.60
Chen *et al.* [3]		-7.15
Zou *et al.* [11]		-3.68
Ours (Large)		**-11.85**
Ours (Medium)		**-10.27**
Ours (Small)		**-8.94**
VVC	-	0

3. Comparison with Recent LIC Works

To get quantitative results, we present the BD-rate [2] computed from PSNR-BPP curves as the quantitative metric. The anchor RD-performance is set as the results of VVC on different datasets (BD-rate=0%). The Table 1 shows the results. As results show, we outperform the previous works and achieve SOTA performance based on the three datasets with different resolutions.

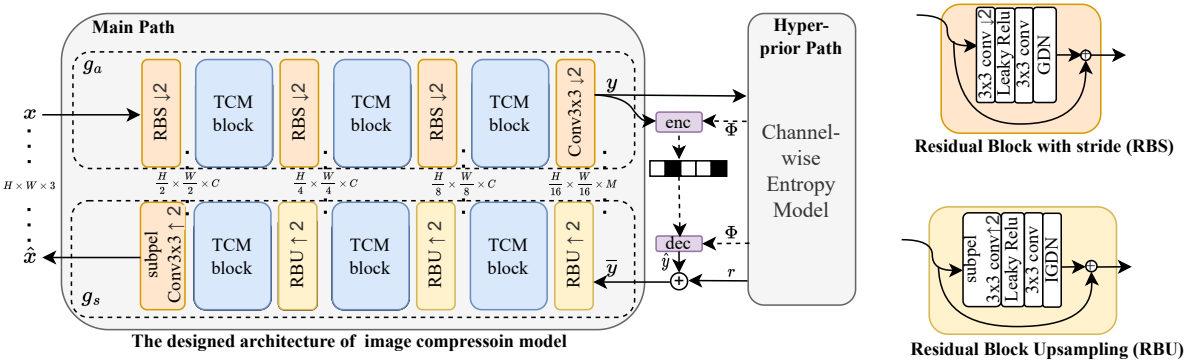

Figure 1. The overall framework (left). The architectures of RBS and RBU in [4] (right).

4. Ablation Studies on Various Entropy Estimation Models

To verify our TCM blocks can improve the overall RD-performance, in addition to test the model with the channel-wise entropy model in [8], we also try the model using the spatial-wise entropy model in [7]. The results are shown in Fig. 2. We define the model where the main path uses TCM block as "TCMmain". We compare the TCMmain model using spatial-wise entropy model with "SwinT-Hyperprior" model in [10] and the model in [4]. All of these three methods use spatial-wise entropy models. The difference is that our model is based on TCM block, the model in [4] is based on CNN, and "SwinT-Hyperprior" is based on swin-transformer. As we can see, our method can get the best RD-performance. This suggests that the TCM blocks can significantly improve image compression, and are robust to different entropy models.

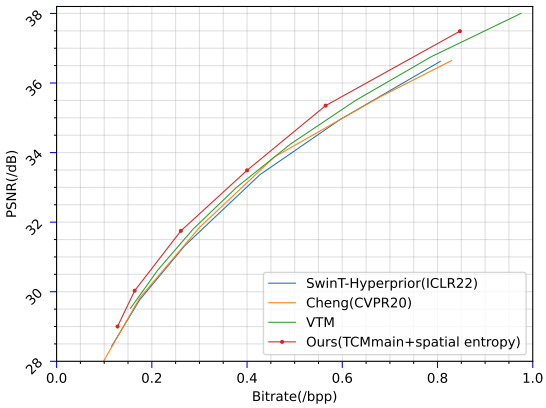

Figure 2. Performance evaluation of the models using the spatial-wise entropy model [7] on the Kodak dataset.

5. Ablation Studies on the Numbers of Slices

The number of slices s is an important hyper-parameter for channel-wise entropy model in [8]. A larger number leads to lower efficiency, while a lower number causes a worse RD-performance. To find a suitable number, we test some different number setting $s = \{2, 4, 5, 8, 10\}$ for our entropy model with the proposed SWAtten. The main path of the tested model is the same as the main path in [8]. The entropy model is also similar, the difference is that we add SWAtten. The results are shown in Fig. 3. As we can see, when s is low, we get a bad RD-performance. With s increasing, the performance is improved. But when $s > 5$, the improvement of the performance is not obvious, and even decreases. This indicates that 5 slices have been able to learn enough information in our model. Therefore, we set s as 5 in our model to achieve the balance between running speed and RD-performance.

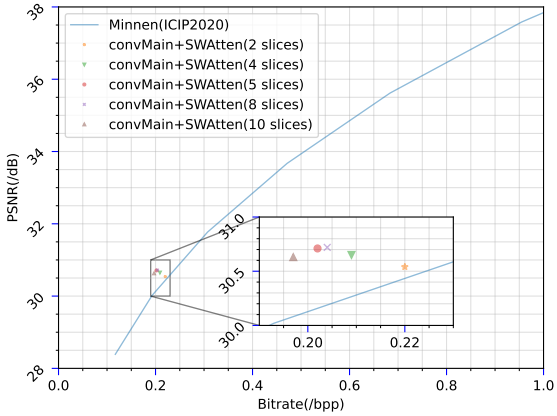

Figure 3. Effect of different slices numbers on RD-performance. "convMain" means we use the main path in [8].

6. Abaltion Studies on the Design of SWAtten

We test the SWAtten w/o CNN for attention map (green point) in Fig. 4. In addition, we also evaluate the case w/o the swin transformer (yellow point). From the results, we can see that using either of these two modules can bring about 0.1dB PSNR improvement with fewer bitrates.

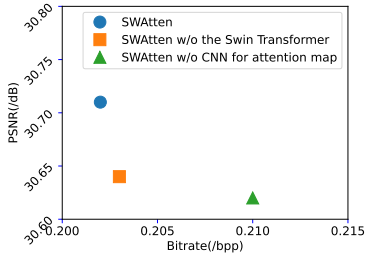

Figure 4. The ablation study on SWAtten (using the same main path as [8].

7. Visualization

We conducted a comparison between our TCM-based model and both a CNN-based [4] and Transformer-based [11] model using the Kodak dataset's $kodim19$ and $kodim20$ images. The results of this comparison are presented in Fig. 5. We focused our analysis on two local regions, and the differences between the three models are noticeable. In the upper local area of $kodim19$, our method effectively reconstructed some of the road sign details while making it easier to differentiate between the back fences. The pasted poster color is also clearer and not distorted. In contrast, our method generated fewer artifacts compared to the CNN-based/Transformer-based models when reconstructing the lower local region. For $kodim20$, our method generated the clearest sign in the left local region, and we achieved a clearer "E" letter than the other two methods in the right local region.

References

[1] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *Proceedings of the International Conference on Learning Representations*, 2018. 1

[2] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. In *VCEG-M33*, 2001. 1

[3] Fangdong Chen, Yumeng Xu, and Li Wang. Two-stage octave residual network for end-to-end image compression. In *Proceedings of AAAI conference on artificial intelligence*, 2022. 1

[4] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020. 1, 2, 3, 4

[5] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[6] Jun-Hyuk Kim, Byeongho Heo, and Jong-Seok Lee. Joint global and local hierarchical priors for learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5992–6001, 2022. 1

[7] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 2

[8] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020. 1, 2, 3

[9] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 162–170, 2021. 1

[10] Yinhao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International Conference on Learning Representations*, 2022. 2

[11] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 1, 3, 4

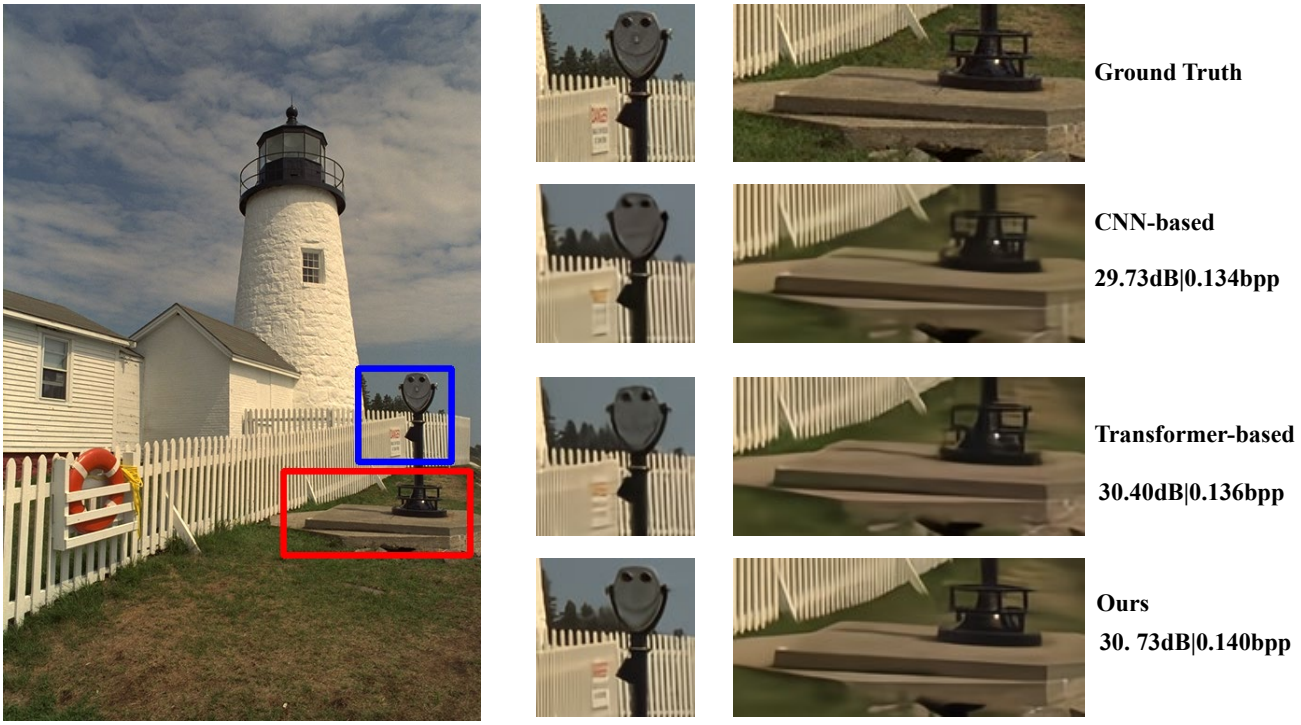

Figure 5. The visualization of $kodim19$ in Kodak dataset by using our TCM-based model, CNN-based model [4], and transformer-based model [11]. PSNR|Bit-rate is listed in the last column.

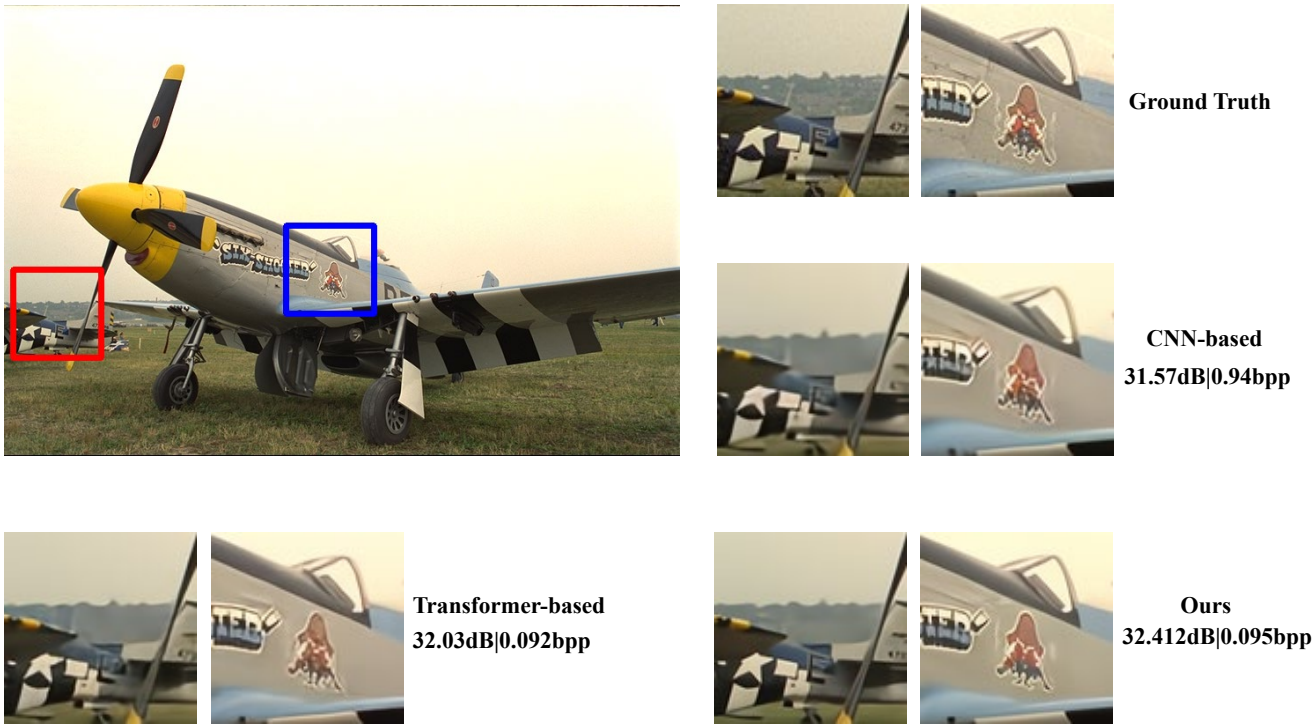

Figure 6. The visualization of $kodim20$ in Kodak dataset by using our TCM-based model, CNN-based model [4], and transformer-based model [11]. PSNR|Bit-rate is listed on the subfigures' right.