Linear Attention Modeling for Learned Image Compression

Supplementary Material

A. Performance Details

This section provides additional details regarding the results presented in Table 1.

The rate-distortion (R-D) results can vary across different VTM anchors due to different evaluation process. To provide a generally accepted baseline on Kodak, we adopt the R-D results from the CompressAI repository [5], which are collected from VTM-9.1. For other datasets, we use the following script to evaluate images in YUV space using VTM-9.1, where QPs range from 22, 27, 32, 37, 42, 47.

```
VTM/bin/EncoderAppStatic -i [input.yuv]
  -c VTM/cfg/encoder_intra_vtm.cfg
  -o [output.yuv] -b [output.bin]
  -wdt [width] -hgt [height] -q [QP]
  --InputBitDepth=8 -fr 1 -f 1
  --InputChromaFormat=444
```

Regarding runtime, FAT is reported to have a decoding time of 242 ms; however, our tests indicate a significantly longer decoding time of 426 seconds. This discrepancy remains an unresolved issue documented in its GitHub repository. Based on our analysis, decoding a single slice with FAT's T-CA entropy model involves computing masked channel attention across 12 layers, whereas TCM requires only two layers for decoding each slice. Incorporating techniques such as KV caching could potentially reduce the FLOPs required for each slice during decoding. Furthermore, the authors have acknowledged that the entropy coder in FAT requires additional optimization to improve decoding efficiency.

For complexity measurements, we use the *thop* Python package to calculate parameters and FLOPs, ensuring consistency with the methodology employed for TCM [19]. However, *thop* has known limitations: it cannot account for FLOPs arising from non-layer-specific operations such as mathematical functions, matrix multiplications (e.g., in attention mechanisms), or CUDA-specific implementations. While the majority of FLOPs originate from Torch integrated layers, the values reported in Table 1 provide a reasonable and fair reference for comparison.

B. Additional Experiment Results

This section presents additional experimental results comparing our method with recent learned image compression (LIC) approaches. We present the BD-rate (MS-SSIM) results in Table 4, with VTM-9.1 as anchor. In fact, only a few recent works have publicly available MS-SSIM optimized models or corresponding curves on Tecnick/CLIC, resulting in some missing results.

Method	Kodak	CLIC	Tecnick
VTM-9.1	0.00%	0.00%	0.00%
ELIC	-7.57%	-	-
TCM-large	-49.76%	-	-
MLIC++	-52.99%	-47.43%	-53.14%
FAT	-51.64%	-	-
LALIC (Ours)	-51.23%	-46.97%	-49.47%

Table 4. BD-rate (MS-SSIM) performance relative to VTM-9.1 across different datasets. "-" indicates an unavailable result.

C. Linear Attention Mechanisms

Except the vanilla Attention which has a quadratic complexity, common modules have a linear complexity, including convolution, window-based attention. The recent linear attention methods, RWKV and Mamba are widely recognized for their efficiency in handling large-scale sequences to get a global reception filed, and also maintains the linear complexity with respect to the input size.

To provide a clearer comparison of these methods, Table 5 summarizes the theoretical time complexity of various attention mechanisms and the typical values of their number of operations (#OPs).

Methods	Time Complexity	#OPs
AFT [42]	$7LD$	$7LD$
AFT+Shift	$7LD + 50LD$	$57LD$
BiWKV+Shift	$29LD + 50LD$	$79LD$
Window Attention [21]	$2w^2LD\,(w = 8)$	$128LD$
Selective Scan [12]	$9NLD\,(N = 16)$	$144LD$
Selective Scan 2D [20]	$4 \times 9NLD\,(N = 16)$	$576LD$

Table 5. Theoretical time complexity of various attention mechanisms in terms of number of operations (#OPs).

In all cases, the computational cost is directly proportional to $L \cdot D$, where L represents the sequence length, and D denotes the latent dimension. The theoretical FLOPs for various mechanisms are outlined below:

- **AFT+Shift**: The complexity of the AFT (named AFT-simple in [42]) is estimated as $7LD$ by the *torch-operation-counter* package. Adding the 5x5 depth-wise convolution shift operation ($25LD \times 2$ for both spatial and channel mix modules) increases the total complexity to $57LD$.

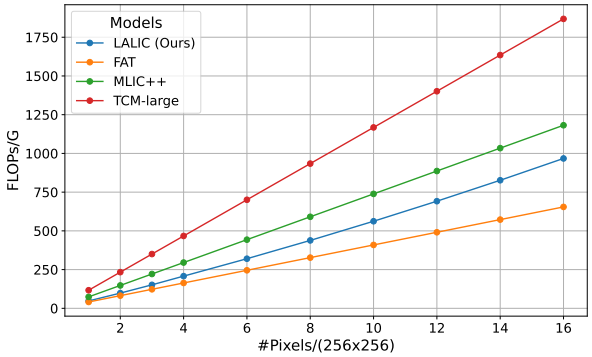
- **BiWKV+Shift**: The BiWKV [9] mechanism, computed as $29LD$ according to the Vision-RWKV GitHub repository, combined with the shift operation results in $79LD$.
- **Window Attention**: The window attention [21] mechanism has a complexity of $2w^2LD$, where w is the window size, typically set to 8, resulting in $128LD$.
- **Selective Scan**: In Mamba [12], the selective scan mechanism has a complexity of $9NLD$, where N is the state dimension, typically set to 16, leading to $144LD$. In SS2D [20], the selective scan is performed four times, resulting in a total complexity of $4 \times 9NLD = 576LD$.

As shown in Table 5, BiWKV attention demonstrates significant computational efficiency compared to these other mechanisms, making it a compelling choice for balancing performance and complexity.

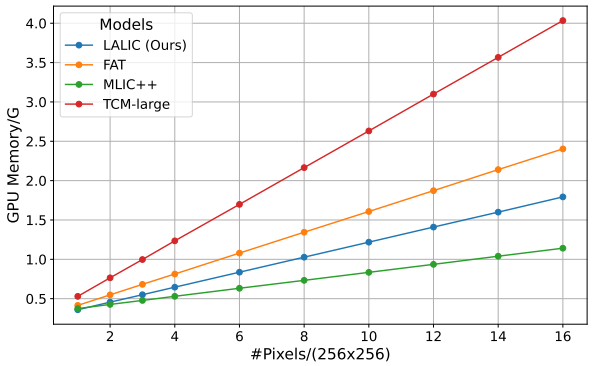
D. Linear Complexity on Scaling

Practical learned image compression (LIC) methods exhibit linear complexity with respect to the number of pixels, as shown in Figure 10. Unlike previous demonstrations [16] that used a quadratic x-axis and presented a quadratic trend for all methods, this figure employs a linear x-axis for clarity, providing a more intuitive understanding for readers. The maximum resolution tested is 1024×1024.

Among recent LIC methods, our proposed LALIC demonstrates medium-low FLOPs and forward GPU memory usage, striking a balance between computational efficiency and memory requirements.



(a) FLOPs vs. Resolution



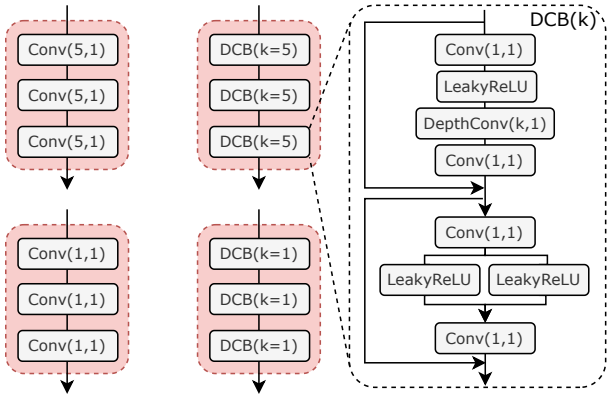
(b) GPU Memory Usage vs. Resolution

Figure 10. Linear scaling trends of FLOPs (a) and GPU memory usage (b) for different LIC methods as a function of image resolution. LALIC achieves competitive performance with medium-low computational and memory demands.

E. Entropy Model Architecture

For entropy models, we adopt the Conv SCCTX model [14] and an enhanced Conv Plus SCCTX configuration as reference baselines. The detailed network architectures of these models are illustrated in Figure 11.

The Conv SCCTX model consists of three 5×5 convolutional (Conv) layers designed to extract channel context, followed by three 1×1 Conv layers for entropy parameter estimation. The Conv Plus SCCTX configuration extends this architecture by incorporating Depth Conv Block (DCB) from the DCVC[33] learned video coding series, where the hyperparameter k denotes the kernel size of the depthwise convolution. To further enhance the modeling capacity, we increase the channel dimensions in the depthwise convolution layers, thereby raising the number of context parameters.



(a) Conv SCCTX (b) Conv Plus SCCTX

Figure 11. Network architecture of Conv SCCTX and Conv Plus SCCTX configurations. The upper module represents channel context extraction, while the lower module corresponds to parameter aggregation.

F. Subjective Results

We conducted a subjective comparison of reconstructed images generated by our LALIC model and our trained TCM-large model on the Kodak dataset. The results are shown in Figure 12 and Figure 13. By focusing on specific image regions, we observe that our proposed method preserves finer details compared to TCM-large. For instance, LALIC retains sharper textures in the wooden board on the right side

(a) Original (b) TCM-large 0.767 bpp / 32.07 dB (c) LALIC (Ours) 0.759 bpp / 32.12 dB

(d) Original crop (e) TCM-large crop (f) LALIC (Ours) crop

Figure 12. Subjective quality comparison on the $kodim01$ image from Kodak.

(a) Original (b) TCM-large 0.245 bpp / 34.20 dB (c) LALIC (Ours) 0.243 bpp / 34.30 dB

(d) Original crop (e) TCM-large crop (f) LALIC (Ours) crop

Figure 13. Subjective quality comparison on the $kodim02$ image from Kodak.

of Figure 12 and captures the intricate structure of the door handle in Figure 13.

In addition to qualitative improvements, our method achieves higher PSNR values while maintaining a lower bitrate, highlighting its superior rate-distortion performance over TCM-large.