Supplemental Material for *High-Efficiency Lossy Image Coding Through Adaptive Neighborhood Information Aggregation*

Ming Lu and Zhan Ma

Abstract—This companion document provides additional information to further evidence the generalization of the proposed *TinyLIC* for the support of extra functionalities such as the variable-rate compression, various input sources, in-loop filtering, etc.

Index Terms—Supplemental Material

1 Additional Details of *TinyLIC*

1.1 Model Parameters

Table 1 and 2 details the network parameters used in *TinyLIC*. Th example of "Conv: k3c128s2" stands for a convolutional layer having convolutions with spatial kernel size at 3×3 (k3), 128 channels (c128), and a stride of 2 based spatial downsampling (s2) at both dimensions. The same convention is applied to other convolutional settings. It is worth to point out that in transposed convolutions at decoder, "s2" stands for the spatial upsampling at a stride of 2. Interested parties can either follow these settings to implement the *TinyLIC* from the scratch or clone our project from https://njuvision.github.io/TinyLIC directly for reproducible research.

1.2 Extra Visualizations

We also offer more qualitative visualizations using Tecnick and CLIC image samples in Fig. 4 and Fig. 5 respectively. Similar as the results in the main content of this work, we can clearly observe the subjective improvements of the proposed *TinyLIC* in comparison to the BPG and VVC. For wall tile textures and flying hair in closeups of resepctive Fig. 4a and 5a, our *TinyLIC* provides sharper and less noisy reconstructions which are closer to the ground truth samples.

1.3 BD-rate Performance on Extra Dataset

In addition to the Kodak, CLIC and Tecnick datasets, we further evaluate the *TinyLIC* on common test dataset suggested by the IEEE 1857.11 Learning-based Image Coding committee. This dataset is referred to as the NIC_Dataset:

- The NIC_Dataset is a public dataset at https://pan.baidu.com/s/1dPTg9JRh4PS748zxdCUUtA with access code `p76h`.
- Test set contains 24×4 = 96 images with 4 different resolutions (ClassA_6K, ClassB_4K, ClassC_2K, ClassD_Kodak).

As quantitatively measured in Table 3, we can still observe the lead of BD-rate gains of *TinyLIC* to the most recent VVC Intra, for the compression of RGB images at various resolutions and bitrates. Note that NIC_Dataset also provides training and validation images. However, to evidence the model generalization, we directly reuse pretrained *TinyLIC* to compress image samples from the test set of NIC_Dataset.

2 Variable-rate Model

Past learned LICs [2]–[4] trained different models for different target bitrates by varying λ as discussed in the main content. Apparently, the use of rate-specific model needs to switch the model on-the-fly for the support of a wider

TABLE 1
Network Settings of *TinyLIC* at Low Bitrates

Main Encoder	Main Decoder	Hyper Encoder	Hyper Decoder	Context Prediction	Entropy Parameters
Conv: k5c128s2	RSTB×2	Conv: k3c128s2	RSTB×2	Masked: k3c384s1	Conv: k1c640s1
RSTB×2	TConv: k3c128s2	RSTB×2	TConv: k3c128s2	Masked: k3c384s1	GELU
Conv: k3c128s2	RSTB×6	Conv: k3c128s2	RSTB×2	Masked: k3c384s1	Conv: k1c512s1
RSTB×4	TConv: k3c128s2	RSTB×2	TConv: k3c384s2		GELU
Conv: k3c128s2	RSTB×4				Conv:k1c384s1
RSTB×6	TConv: k3c128s2				
Conv: k3c192s2	RSTB×2				
RSTB×2	TConv: k5c3s2				

TABLE 2
Network Settings of *TinyLIC* at High Bitrates

Main Encoder	Main Decoder	Hyper Encoder	Hyper Decoder	Context Prediction	Entropy Parameters
Conv: k5c192s2	RSTB×2	Conv: k3c192s2	RSTB×2	Masked: k3c640s1	Conv: k1c1066s1
RSTB×2	TConv: k3c192s2	RSTB×2	TConv: k3c192s2	Masked: k3c640s1	GELU
Conv: k3c192s2	RSTB×6	Conv: k3c192s2	RSTB×2	Masked: k3c640s1	Conv: k1c853s1
RSTB×4	TConv: k3c192s2	RSTB×2	TConv: k3c640s2		GELU
Conv: k3c192s2	RSTB×4				Conv:k1c640s1
RSTB×6	TConv: k3c192s2				
Conv: k3c192s2	RSTB×2				
RSTB×2	TConv: k5c3s2				

TABLE 3
BD-rate Performance of VVC Intra and *TinyLIC* on NIC_ Dataset. Anchor is the BPG. Distortion is measured by PSNR.

Class	VVC Intra		*TinyLIC*	
	High Bitrate	Low Bitrate	High Bitrate	Low Bitrate
A	-15.1%	-23.6%	-22.5%	-26.6%
B	-15.3%	-23.7%	-19.3%	-23.6%
C	-22.4%	-28.8%	-28.7%	-31.3%
D*	-19.0%	-23.5%	-20.5%	-26.4%
Ave.	-17.9%	-24.9%	-22.8%	-27.0%

* Class D images are from Kodak testing samples.

TABLE 4
BD-rate Performance of Variable-Rate Model Enabled by the ScalingNet [1] Against the Anchor Using Multiple Rate-Specific Models for the proposed *TinyLIC*. Numbers Are Averaged for each Dataset. *The smaller number the better.*

dataset	BD-rate	
	High Bitrate	Low Bitrate
Kodak	-1.35%	-0.9%
CLIC	-1.78%	+0.46%
Tecnick	-1.87%	+0.36%

bitrate range which requires a large amount of storage to cache model appropriately for cost-efficient optimization. Thus, variable-rate model that can support a fairly wide range of bitrates is of great importance for the enabling of learned LIC in practice.

Our early attempt in [5], [6] applied a set of quality scaling factors ($\mathbf{s}_f$) at the bottleneck (see Fig. 1a) to adapt bitrates in a specific range. Given a high bit-rate R_0, input image $\mathbf{x}$ is encoded by $\mathbb{E}$ to derive corresponding latent features $\mathbf{y}^0 = \mathbb{E}(\mathbf{x})$. Then, scaling factors ($\mathbf{s}_f \in \{a_0, b_0\}, \{a_1, b_1\}, ..., \{a_n, b_n\}$) are used to linearly scale each of n channels of latent features to produce new bitrates. Please refer to [5] for more details.

Later, Lin et al. [1] replaced aforementioned linear scaling factors with neural networks based approach, dubbed as ScalingNet. Additionally, instead of only placing the scaling operations at bottleneck layer, it suggested to devise them at each stage in main coder, as shown in Fig. 1b. Compared with simple linear factors, ScalingNet can enable fine-grained bitrate adaptation with negligible rate-distortion (R-D) loss against the anchors using multiple rate-specific models. The ScalingNet was adopted into the baseline mode of IEEE 1857.11 for next-generation learning-based image coding where the baseline model was migrated from our early work in [6].

We therefore simply extend the ScalingNet shown in Fig. 1b to the proposed *TinyLIC* and examine its efficiency. We use a total of four models to cover the whole bitrate range (e.g., roughly under 1.5bpp for typical use cases) and to reach arbitrary bitrate points desired in applications; while in default, we need to train specific model for a given bitrate, for which a great number of models need to be pretrained in advance for the enabling of aforementioned fine-grained bitrate adaptation. We then encode images using ScalingNet enhanced *TinyLIC* to match the bitrates of *TinyLIC* anchors. Results are listed in Table 4. As seen, the BD-rate is slightly increased but overall it is negligible, which promises the encouraging application prospects of the *TinyLIC*.

3 Adaptive Inloop Filtering

Having an enhancement network in either post-processing or in-loop processing can increase the quality of image reconstruction and then improve the end-to-end BD-rate performance for both learned and rules-based image/video coding methods [7]–[14]. As reviewed previously, Xie et al. 2021 [15] embedded a feature enhancement network with the invertible neural network for better compression.

Here we show that having an adaptive loopfilter (ALF) with *TinyLIC* can further improve the compression performance. Considering the fundamental challenge between model complexity (e.g., space and time complexity) versus model efficacy (e.g., performance and generalization), we choose to use the *multi-hypothesis sample refinement* (MSR) developed in [12]. We choose to apply the MSR because any popular CNN models can be integrated under this framework as detailed in [12]. For example having a CNN model with fairly large-scale model size could offer upto 10% additional BD-rate gain. This section uses a fairly small-scale neural model with 1.7M parameters (e.g., 20% of the default *TinyLIC*) for illustration.

As shown in Fig. 2 the proposed MSR is to linearly superimpose multiple distortion hypotheses (MDH) $\mathbf{d}_i, i \in [0, N-1]$ to best mitigate the compression noise in video codingby optimizing the MMSE between the ALF refined reconstruction block and its uncompressed counterpart, i.e.,

$$\arg\min_{a_i} \sum_{x,y\in\Phi} ||\mathbf{I}(x,y) - (\hat{\mathbf{I}}(x,y) + \hat{\mathbf{D}}(x,y))||^2, \quad (1)$$

with

$$\hat{\mathbf{D}}(x,y) = \sum\nolimits_{i\in[0,N-1]} a_i \cdot \mathbf{d}_i(x,y). \quad (2)$$

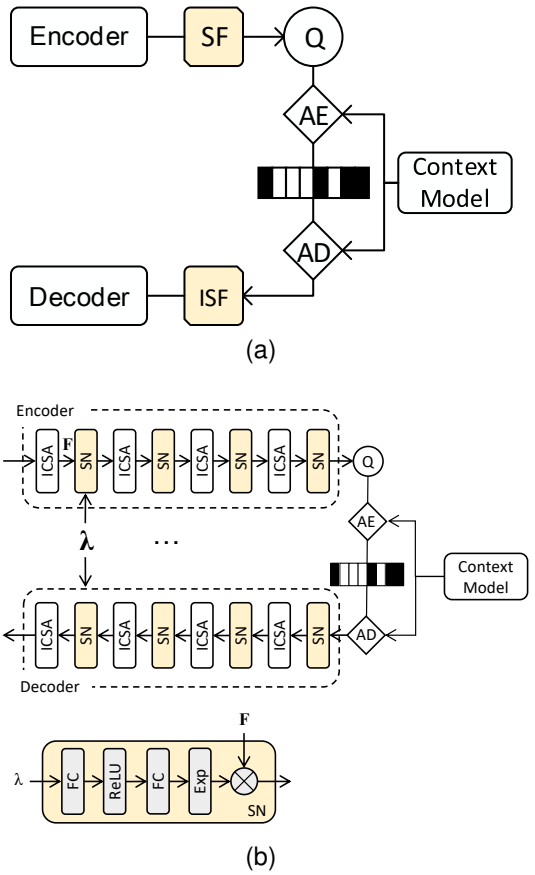


Fig. 1. **The Enabling of Variable-rate Model.** (a) Scaling factors [5]; (b) Neural networks based ScalingNet [1] (SN). The use of variable-rate control is exemplified in main encoder-decoder pair. FC stands for full-connected layer, and Exp is the exponential action.

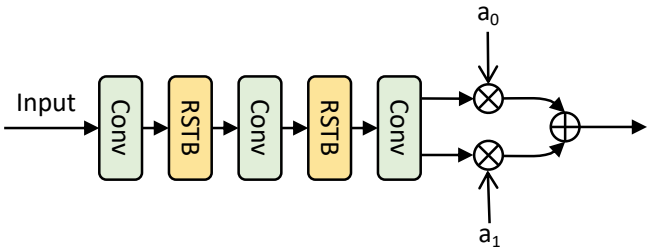


Fig. 2. **MSR Enhanced** *TinyLIC*. Convolutional (Conv) layers uniformly apply the 3×3 convolutions where 128 output channels are used for first two Conv layers and 6 output channels are devised at the third Conv layer to produce two hypotheses at the size of $H \times W \times 3$. RSTB is set the same as in main paper.

a_is are superimposition coefficients associated with MDHs that are encapsulated and signaled in compressed bitstream.

4 Support of Various Image Sources

To ensure broader adoption of the proposed *TinyLIC* in vast scenarios, one key feature is to support different image format as the input. In addition to the RGB sources, here we exemplify the use cases of the support of YUV420[1] and

1. The use of YUV420 allows us to use low-resolution chrominance for data saving without noticeable perceptual distortion [16] because the human visual system is more sensitive to luminance components.

TABLE 5
BD-rate Performance of MSR enhanced *TinyLIC* Against the default *TinyLIC* without In-loop Filtering. Numbers Are Averaged for each Dataset. *The smaller number the better*

dataset	BD-rate ↓	
	High Bitrate	Low Bitrate
Kodak	-1.74%	-2.19%
CLIC	-2.28%	-2.25%
Tecnick	-2.17%	-2.17%

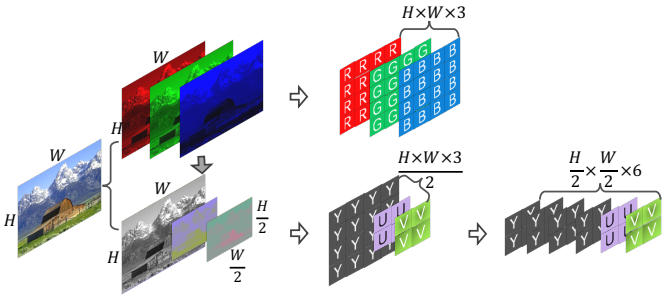


Fig. 3. **Pixel Arrangement** for *TinyLIC* to process various image sources in both training and inference stages.

Y (monochrome) images. As illustrated in Fig. 3, native RGB image at a size of $H \times W \times 3$ is processed directly by stacking R, G, B attributes of each pixel; while for image in YUV420 format, it is first converted from the native RGB representation, and then rearranged to a pile of YYYYUV at a size of $\frac{H}{2} \times \frac{W}{2} \times 6$ for compression. Besides, if we want to compress monochrome image, we can just need to process the luminance component of the native RGB content, a.k.a, Y attribute as in Fig. 3 if using YUV color space.

TABLE 6
BD-rate Performance of *TinyLIC* Upon YUV420 Images. Anchor is the VVC Intra. Numbers are Averaged for each Dataset. *The smaller number the better*

dataset	Y BD-rate ↓		YUV BD-rate ↓	
	High Bitrate	Low Bitrate	High Bitrate	Low Bitrate
Kodak	-20.72%	-16.77%	-18.74%	-13.57%

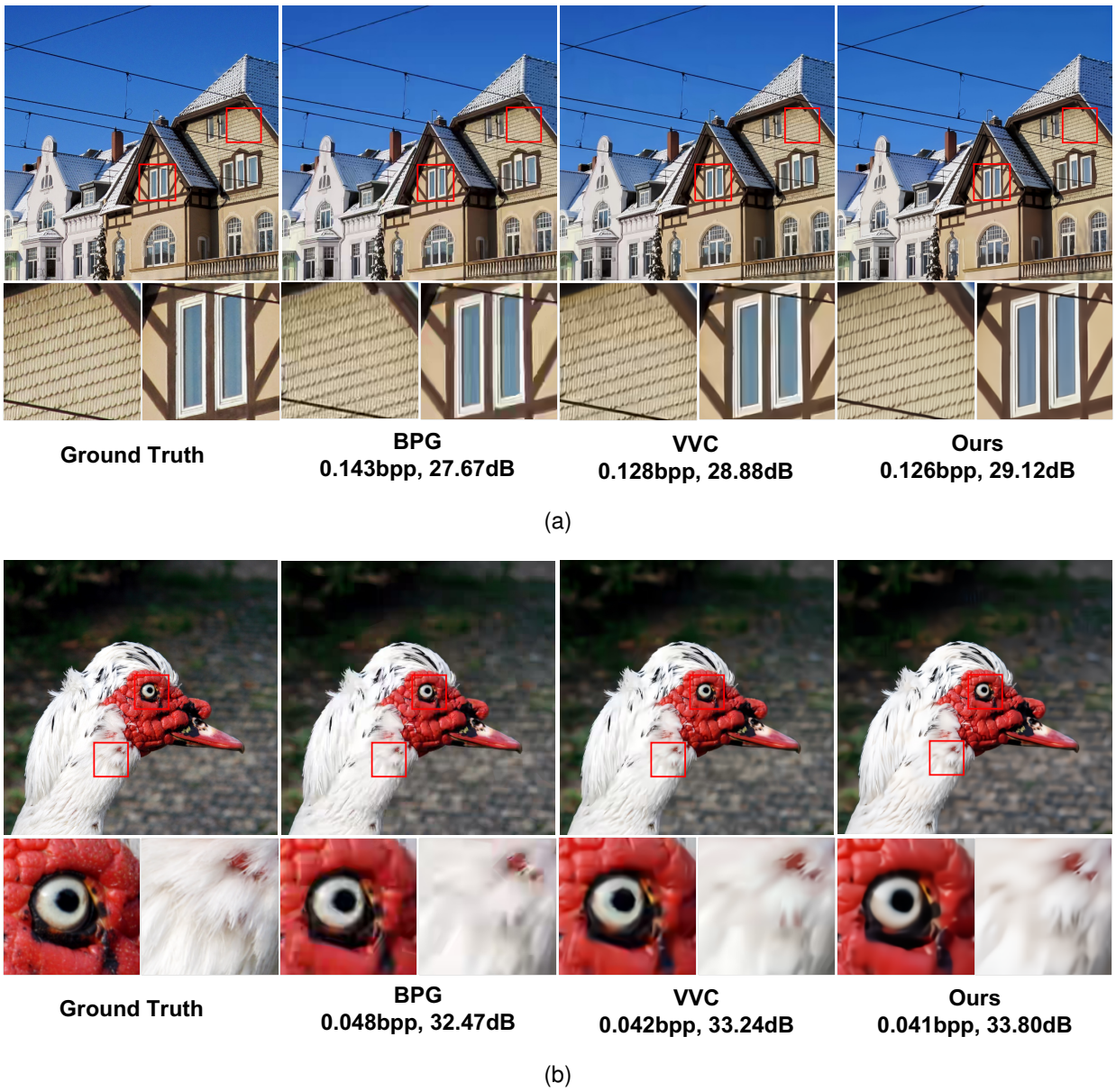


Fig. 4. **Qualitative Visualization on Tecnick Dataset.** Reconstructions and close-ups of the BPG, VVC and our *TinyLIC*. Both bpp and PSNR are marked. (a) RGB_OR_1200x1200_023, (b) RGB_OR_1200x1200_056.

REFERENCES

[1] J. Lin, M. Akbari, H. Fu, Q. Zhang, S. Wang, J. Liang, D. Liu, F. Liang, G. Zhang, and C. Tu, "Variable-rate multi-frequency image compression using modulated generalized octave convolution," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6. 2, 3

[2] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018. 1

[3] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 794–10 803. 1

[4] Z. Cheng, H. Sun, M. Takeuchi, and J. 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*, 2020, pp. 7939–7948. 1

[5] T. Chen and Z. Ma, "Variable bitrate image compression with quality scaling factors," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2163–2167. 2, 3

[6] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "End-to-end learnt image compression via non-local attention optimization and improved context modeling," *IEEE Transactions on Image Processing*, vol. 30, pp. 3179–3191, 2021. 2

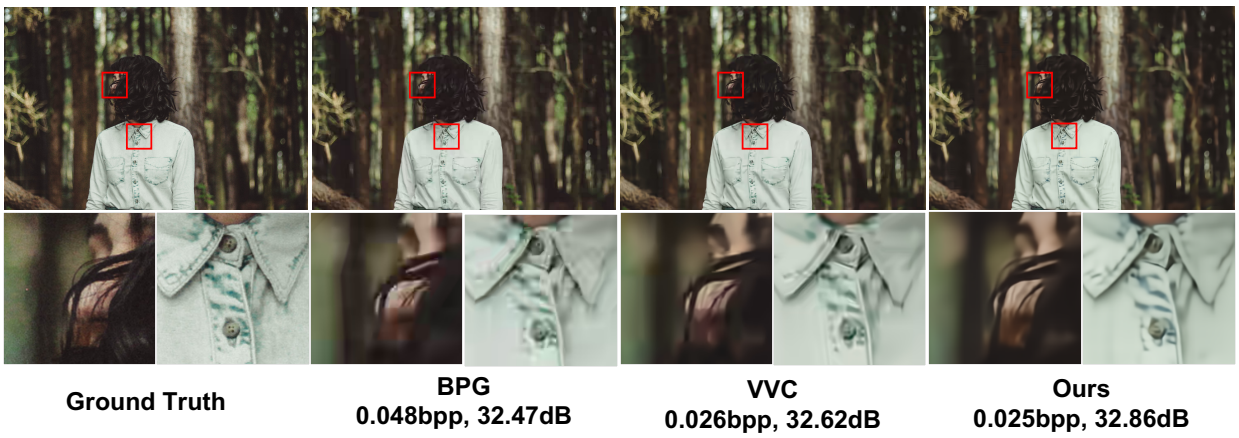
[7] H. Liu, T. Chen, Q. Shen, and Z. Ma, "Practical stacked non-local attention modules for image compression." in *CVPR Workshops*, 2019, p. 0. 2

[8] D. Ding, Z. Ma, D. Chen, Q. Chen, Z. Liu, and F. Zhu, "Advances in video compression system using deep neural network: A review and case studies," *Proceedings of the IEEE*, 2021. 2

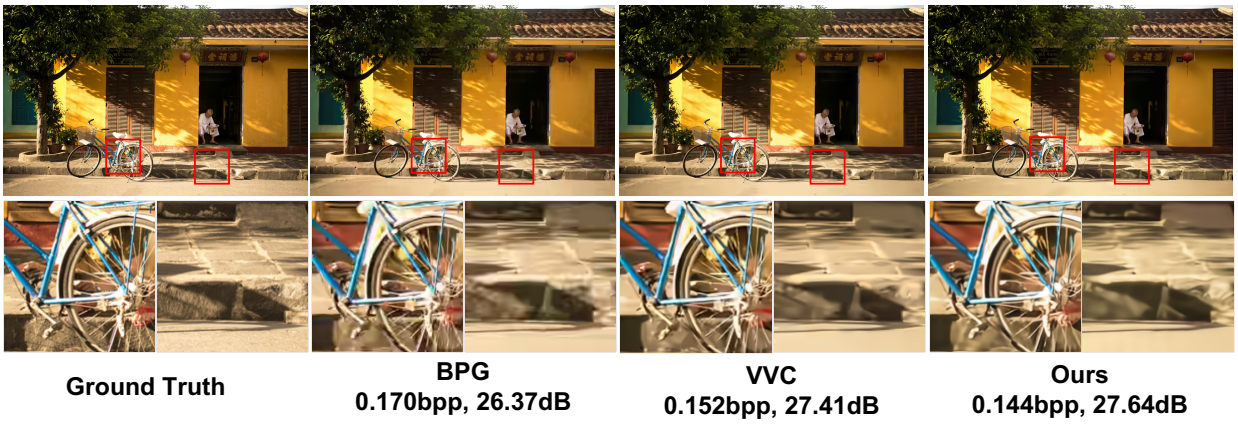
[9] M. Karczewicz, N. Hu, J. Taquet, C.-Y. Chen, K. Misra, K. Andersson, P. Yin, T. Lu, E. François, and J. Chen, "Vvc in-loop filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3907–3925, 2021. 2

[10] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. V. der Auwera, "HEVC deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012. 2

[11] D. Ding, X. Gao, C. Tang, and Z. Ma, "Neural reference synthesis for inter frame coding," *IEEE Transactions on Image Processing*, vol. 31, pp. 773–787, 2021. 2



(a)



(b)

Fig. 5. **Qualitative Visualization on CLIC Dataset.** Reconstructions and close-ups of the BPG, VVC and our *TinyLIC*. Both bpp and PSNR are marked. (a) allef-vinicius-109434, (b) thong-vo-428.

[12] D. Ding, G. Zhen, J. Wang, D. Mukherjee, U. Joshi, Y. Chen, and Z. Ma, "Neural adaptive loop filtering for video coding: Exploring multi-hypothesis sample refinement," *submitted to IEEE Trans. Image Processing*, 2022. 2

[13] M. Lu, T. Chen, H. Liu, and Z. Ma, "Learned image restoration for vvc intra coding." in *IEEE CVPR Workshops*, 2019. 2

[14] M. Lu, M. Cheng, Y. Xu, S. Pu, Q. Shen, and Z. Ma, "Learned quality enhancement via multi-frame priors for hevc compliant low-delay applications," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 934–938. 2

[15] Y. Xie, K. L. Cheng, and Q. Chen, "Enhanced invertible encoding for learned image compression," in *Proceedings of the ACM International Conference on Multimedia*, 2021. 2

[16] Y. Wang and Y.-Q. Zhang, *Video processing and communications*. Prentice hall Upper Saddle River, NJ, 2002, vol. 1. 3