Transformer-based Learned Image Compression for Joint Decoding and Denoising

Yi-Hsin Chen[1] Kuan-Wei Ho[1] Shiau-Rung Tsai[1] Guan-Hsun Lin[1]
Alessandro Gnutti[2] Wen-Hsiao Peng[1] Riccardo Leonardi[2]
[1]National Yang Ming Chiao Tung University, Taiwan [2]University of Brescia, Italy

***Abstract*—This work introduces a Transformer-based image compression system. It has the flexibility to switch between the standard image reconstruction and the denoising reconstruction from a single compressed bitstream. Instead of training separate decoders for these tasks, we incorporate two add-on modules to adapt a pre-trained image decoder from performing the standard image reconstruction to joint decoding and denoising. Our scheme adopts a two-pronged approach. It features a latent refinement module to refine the latent representation of a noisy input image for reconstructing a noise-free image. Additionally, it incorporates an instance-specific prompt generator that adapts the decoding process to improve on the latent refinement. Experimental results show that our method achieves a similar level of denoising quality to training a separate decoder for joint decoding and denoising at the expense of only a modest increase in the decoder's model size and computational complexity.**

***Index Terms*—Learned image compression, compressed-domain image denoising, Transformer.**

I. Introduction

Natural images captured by digital imaging sensors often exhibit noise due to sensor limitations, ISO settings, low-light conditions, etc. To transmit a noisy image efficiently, a straightforward approach is to perform pre-processing (or post-processing) to remove the noise before (or after) compression. Recent studies [1], [2] show that, as compared to straightforwardly cascading an image codec and a denoising model, optimizing a learned image codec for joint decoding and denoising is more effective in terms of both compression performance and complexity. [1], [2] use clean-noise image pairs to optimize the entire image codec end-to-end with a rate-distortion loss. With this approach, the noise is filtered out to a large extent in the encoder. This results in the original noisy input image not being able to be recovered on the decoder side.

However, there are applications where the preservation of image noise is crucial to the trustworthiness of images. Examples include medical imaging, satellite imagery, and security surveillance, just to name a few. In 2022, the JPEG standard committee launched a JPEG AI project, issuing a call for proposals [3] with the aim of standardizing a learning-based image codec. This codec is designed to transmit a single bitstream that is catered to both the standard image reconstruction and the compressed-domain image processing and machine tasks. In this context, Larigauderie *et al.* [4] propose to optimize the decoder exclusively for joint decoding and denoising, given that the encoder is pre-trained for the standard image reconstruction. Taking a different approach, Alvar *et al.* [5] introduce a scalable coding scheme, allowing the clean image to be decoded from a subset of the compressed latents. To reconstruct the noisy input image, the entire latents needs to be decoded. Both [4] and [5] require separate decoders to switch between the standard image reconstruction and denoising reconstruction.

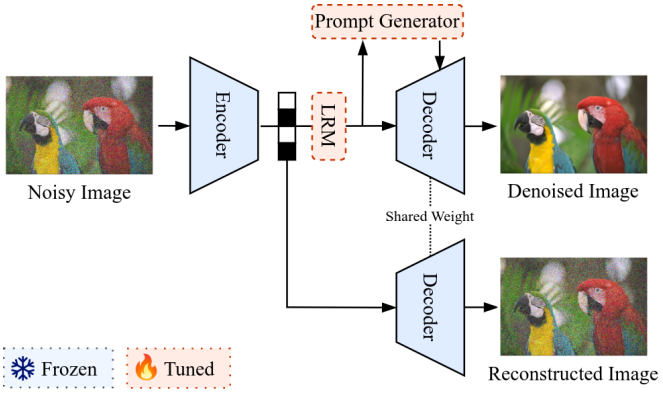


Fig. 1. Overview of the proposed joint decoding and denoising framework.

Our work proposes a Transformer-based image compression system that allows the user to switch between the standard image reconstruction and denoising reconstruction from a single compressed bitstream at inference time. As depicted in Fig. 1, our approach starts with a pre-trained base codec optimized for the standard image reconstruction. When the latents of a noisy input image are decoded, the decoder produces a noisy reconstruction of the input image. To perform joint decoding and denoising, we introduce two add-on modules, Latent Refinement Module (LRM) and Prompt Generator. The former aims to predict the latent representation of a clean image from that of a noisy input image, while the latter is designed to adapt the decoding process of the base decoder to suit the needs of joint decoding and denoising. Our add-on approach allows the base decoder to be re-used and re-purposed without retraining. It incurs only a marginal 28% increase in the decoder's model size, as compared to training a separate decoder (i.e. a 100% increase in the model size) dedicated to the denoising reconstruction. Furthermore, in

This work was supported by National Science and Technology Council, Taiwan, under Grants NSTC-112-2634-F-A49-007- and MOST-110-2221-E-A49-065-MY3, National Center for High-performance Computing, Taiwan, and partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

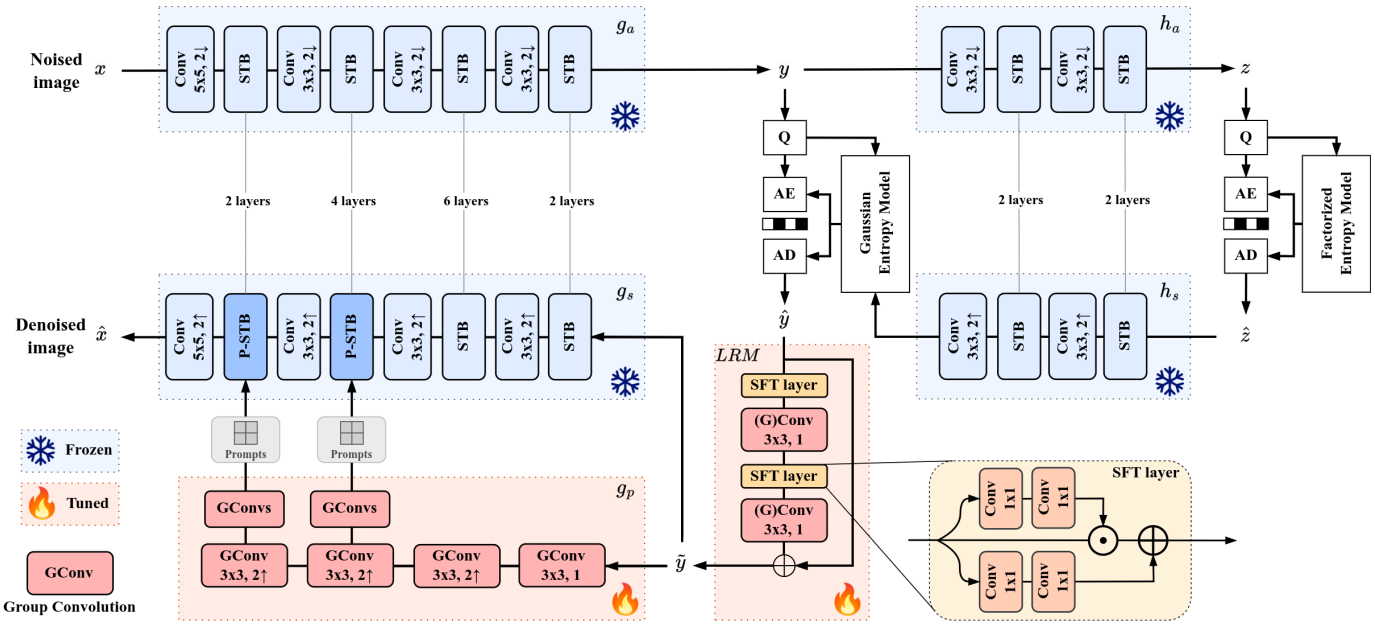


Fig. 2. Illustration of the proposed method. Our work adapts a pre-trained Transformer-based image codec to perform joint image decoding and denoising by introducing a latent refinement module LRM and a prompt generator g_p on the decoder side. The GConv (highlighted in pink color) represents sixteen groups of 3×3 group convolutions. The (G)Conv in LRM is convolutions for our proposed model, and group convolutions for its lightweight variant. Although the picture does not explicitly show it for better visualization, our system can directly process the latent $\hat{y}$ for standard image reconstruction.

terms of the number of the multiply–accumulate operations, our prompt-adapted decoder is only 11% higher than the base decoder.

II. Proposed Method

A. System Overview

This work aims at adapting a pre-trained Transformer-based image codec optimized for the image reconstruction task to perform joint image denoising and decoding. It provides the user with an option to choose between the standard image reconstruction and the denoising reconstruction from a single compressed bitstream. To this end, we introduce a decoder-side adaptation mechanism capable of re-purposing the pre-trained decoder according to the user's preference at inference time.

Fig. 2 illustrates our proposed scheme. We adopt the same pre-trained base codec (highlighted in blue and white colors) as [6] but replace the complicated context model with simple Gaussian prior for entropy coding. In Fig. 2, g_a, g_s and h_a, h_s represent the main and hyperprior autoencoders, respectively. The main autoencoder g_a, g_s consists of multiple Swin-Transformer blocks (STBs), with the convolutional and de-convolutional layers interspersed between STBs to adjust the resolution of the feature maps. Since our base image codec is pre-trained for the standard image reconstruction task, a noisy input image leads to a noisy reconstructed image at the decoder's output. To reuse this pre-trained base codec for compressed-domain image denoising, we introduce several add-on modules, including an LRM and a prompt generator g_p, on the decoder side while leaving the base codec untouched. Notably, given a compressed bitstream, our scheme supports both the standard image reconstruction task and the joint decoding and denoising task. The add-on modules represent a fractional increase in the decoder's model size and computational complexity.

B. Latent Refinement Module

The LRM is to refine the latent representation $\hat{y}$ of a noisy input image, with the aim of predicting the latents of the clean (denoised) image. When the prediction is perfect, the pre-trained decoder should ideally produce a clean reconstruction of the input image, i.e. a denoised image. In a sense, LRM performs compressed-domain image denoising.

As depicted in Fig 2, LRM is a residual block in which the residual generation involves two Spatial Feature Transforms (SFT) [7] interleaved with two 3×3 convolutional layers. The SFT applies spatially adaptive affine transformation, in order to update the feature maps F by $\text{SFT}(F) = (\alpha(F) \odot F) \oplus \beta(F)$, where $\odot$ and $\oplus$ denote element-wise multiplication and addition, respectively. Both $\alpha(\cdot)$ and $\beta(\cdot)$ are learned neural networks. In an effort to reduce the model size and computational cost of LRM, we also explore an alternative implementation of LRM that replaces 3×3 convolutions with sixteen groups of 3×3 group convolutions [8]. This is referred hereafter to as our lightweight variant.

C. Prompt-adapted Swin-Transformer

Recognizing that the predictions made by LRM are not perfect and our Transformer-based base codec is pre-trained and frozen for the standard image reconstruction, we propose a decoder-side, instance-specific prompting technique to adapt the decoder to suit the needs of joint decoding and denoising. Specifically, additional tokens, known as prompts, are generated and injected into the STBs of the decoder without re-training the decoder.

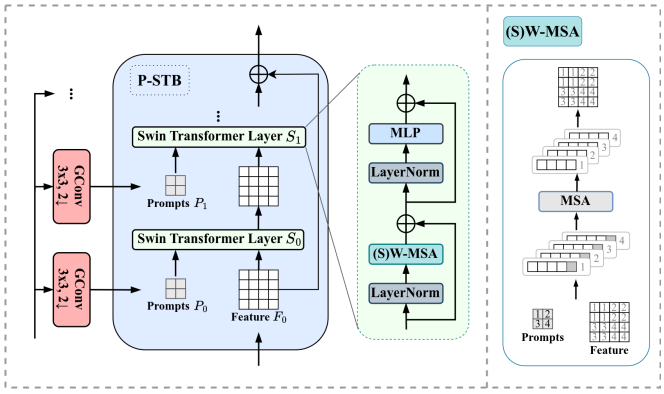


Fig. 3. Network details of the prompt-adapted Swin-Transformer Layer.

As shown in Fig 2, our prompt generator g_p consists of several 3×3 group convolutions. It takes the refined latents $\tilde{y}$ to generate prompts for the last two STBs (closer to the input image space) in the decoder. These prompts change with $\tilde{y}$ and are thus instance-specific (i.e. image-dependent). Fig. 3 illustrates the structure of our prompt-adapted Swin-Transformer block (P-STB). The P-STB comprises several Swin-Transformer layers [9], with shifted window-based multi-head self-attention (W-MSA) serving as the main mechanism for signal transformation. In the absence of prompts, P-STB is an ordinary STB, where W-MSA divides the input tokens F into non-overlapping windows (groups). The tokens in every window are updated through self-attention. In symbols, the attention mechanism is given by

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^\top/\sqrt{d} + B)V, \qquad (1)$$

where $Q = FW_Q$, $K = FW_K$, $V = FW_V$, $W_Q, W_K, W_V \in \mathbb{R}^{C\times C}$ are learnable matrices mapping flattened input $F \in \mathbb{R}^{N\times C}$ into query $Q \in \mathbb{R}^{N\times C}$, key $K \in \mathbb{R}^{N\times C}$, and value $V \in \mathbb{R}^{N\times C}$. Here, N is the number of input tokens in a window, C is the channel dimension of F, and $B \in \mathbb{R}^{N\times N}$ is a learnable positional embedding matrix.

When the prompts P are present, they are spatially divided in the same way as input tokens F. In our design, the number of prompts is one-quarter of that of the input tokens to reduce complexity. In a self-attention window, prompts from the corresponding window update the input tokens following a similar self-attention mechanism to Eq. (1), except that the key K and value V matrices are augmented as $K = [F; P]W_K \in \mathbb{R}^{(N+\frac{N}{4})\times C}$, $V = [F; P]W_V \in \mathbb{R}^{(N+\frac{N}{4})\times C}$, where $[\cdot;\cdot]$ represents concatenation along the token dimension, while query $Q = FW_Q$ remain unchanged. This prompting technique allows the decoding process to be adapted without retraining the base decoder.

III. Experiments

Training details: Following the common test protocol [1], we test our model for both real-world and synthetic noise scenarios. For the real-world case, we train and validate our model on SIDD dataset [10], which has 320 high-resolution images and 1280 256x256 patches. For the synthetic case, we use Flicker2W dataset [11] for training and Urban100 [12] dataset for validation. To construct clean-noise image pairs, we use the same noise simulator [13] as adopted by JPEG AI [14]. It is optimized for estimating the parameters of Poissonian-Gaussian noise model on SIDD dataset [10].

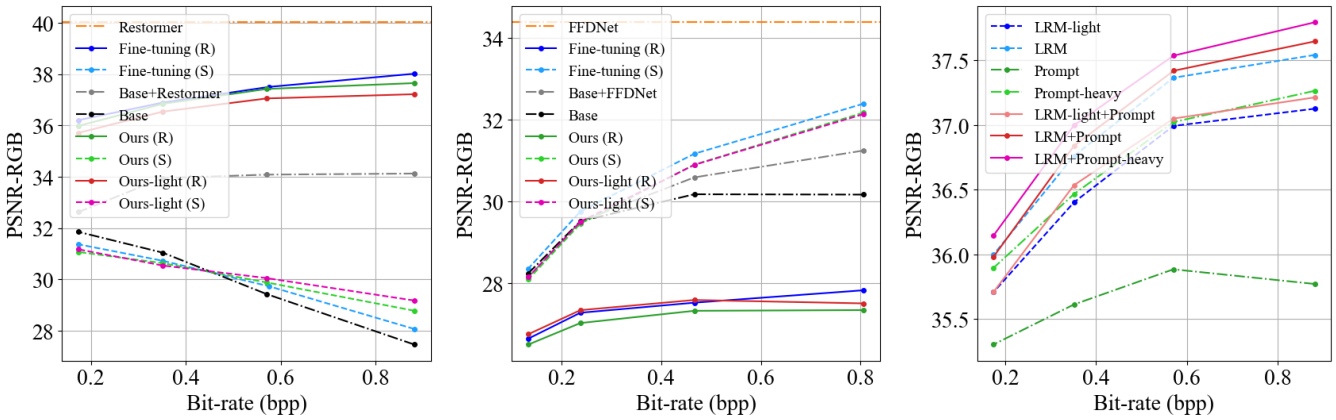
We adopt a two-stage training strategy. In the first stage, the base codec g_a, g_s, h_a, h_s is trained on Flicker2W [11] for the standard image reconstruction task. The training objective is to minimize the rate-distortion cost $R(\hat{z})+R(\hat{y})+\lambda\times D(x,\hat{x})$, where R denotes the bit rates of the image latents and hyperprior, and D measures the mean-squared error between the input x and reconstructed $\hat{x}$ images. λ is set to 0.0018, 0.0035, 0.0067, and 0.013 for separate rate points. In the second stage, we fix the base codec g_a, g_s, h_a, h_s, and train 4 pairs of LRM and the prompt generator g_p for 4 distinct rate points by following the common training strategy for the image restoration task to minimize the l_1 loss between the clean image x and the denoising reconstruction $\hat{x}$.

Evaluation: We test our model on SIDD test set [10] for real-world noise, and Kodak [15] for synthetic noise, which is generated with the same noise simulator [13] as that for training. The quality of denoised images is measured in PSNR-RGB with the noise-free image serving as the original, and the bit-rate in bits-per-pixel (bpp).

Baselines: The baseline methods include (1) using the base codec (g_a, g_s, h_a, h_s) trained for the standard image reconstruction, termed *Base*, (2) fine-tuning the main decoder g_s of the base codec, termed *Fine-tuning*, and (3) using a pre-trained denoising model for post-processing, denoted as *Base + model_name*, e.g. *Base + Restormer*. We adopt Restormer [16] for real-world noise and FFDNet [17] for synthetic noise. Both are the state-of-the-art models. Recall that our work allows the user to switch between the standard image reconstruction and the denoising reconstruction. The decoder is still allowed to reconstruct the noisy input image from the compressed bitstream. As such, we exclude the pre-processing approach, namely denoising followed by compression, for a fair comparison.

A. Rate-Distortion Performance

Figs. 4(a) and 4(b) compare the rate-distortion performance of the competing methods for real-world and synthesis noise, respectively. The terms (R) and (S) indicate the models trained on real-world and synthetic noise, respectively. From the figure, we make the following observations. (1) First, both the normal and lightweight versions of our proposed method achieve substantial gains over *Base*, which simply performs image reconstruction without denoising. This showcases the effectiveness of our approach in performing joint decoding and denoising without retraining the base codec. (2) Second, both our method and *Fine-tuning* outperform the post-processing approaches, *Base+Restormer* and *Base+FFDNet*. Note that these approaches cascade two pre-trained models, forming collectively a large compound model. (3) Third, our method shows comparable or slightly inferior performance to *Fine-tuning*. However, when trained for a specific noise distribution and tested under a different noise distribution (e.g. *Fine-*



(a) Real-world noise (b) Synthetic noise (c) Ablation

Fig. 4. The rate-distortion comparison of the competing methods tested for (a) real-world noise and (b) synthetic noise. Restormer and FFDNet are trained on real-world noise and synthetic noise, respectively. (R) and (S) indicate the models trained on real-world noise and synthetic noise, respectively. (c) shows the ablation experiment on real-world noise.

TABLE I
COMPARISON OF THE KMACS/PIXEL AND MODEL SIZE.

	kMACs/pixel	**Params (M)**
Base	188.52	3.87
Fine-tuning	188.52 (+ 0%)	7.74 (+100%)
Base+Restormer	1461.10 (+675%)	28.40 (+634%)
Base+FFDNet	402.72 (+114%)	4.72 (+22%)
Ours	209.53 (+11%)	4.94 (+28%)
Ours (light)	207.10 (+10%)	4.32 (+12%)

tuning (S) in Fig. 4(a) and *Fine-tuning (R)* in Fig. 4(b)), the performance of *Fine-tuning* deteriorates significantly. This indicates the poor generalization of *Fine-tuning* to different noise distributions. This limitation calls for the need to train separate decoders for drastically different noise distributions. Although suffering from similar generalization issues, our approach only needs to update the lightweight add-on modules, i.e. LRM and the prompt generator g_p, while reusing the base decoder completely. It offers a more cost-effective solution than *Fine-tuning*.

Fig. 5 further presents the subjective quality comparison. Both our method and *Fine-tuning* are able to denoise the input image to a large extent. In comparison, the post-processing methods exhibit rather noticeable noise or artifacts. It is expected that training these compound models end-to-end should improve the performance at the cost of high complexity.

B. Complexity Comparison

Table I reports the decoder-side complexity of the competing methods in terms of (1) the required model size for providing the user with the option to switch between the standard image reconstruction and denoising reconstruction and (2) the number of kilo-multiply-accumulate-operations (kMACs/pixel) for the denoising reconstruction.

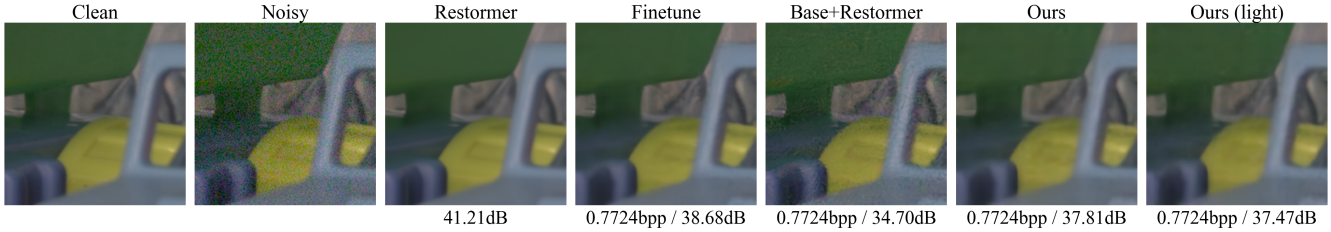
Compared with *Fine-tuning*, which requires one separate full decoder (100%) for the denoising reconstruction, our method incurs only a 28% increase in model size. Remarkably, our lightweight variant has a marginal 12% increase in model size while showing a modest impact on the image quality. Both of these variants increase the kMACs/pixel by about 10% relative to a full decoder for the denoising reconstruction. Note that the post-processing approaches are least preferred due to their high kMACs/pixel.

TABLE II
COMPLEXITY COMPARISON BETWEEN DIFFERENT VARIANTS OF LRMS AND THE PROMPT GENERATORS.

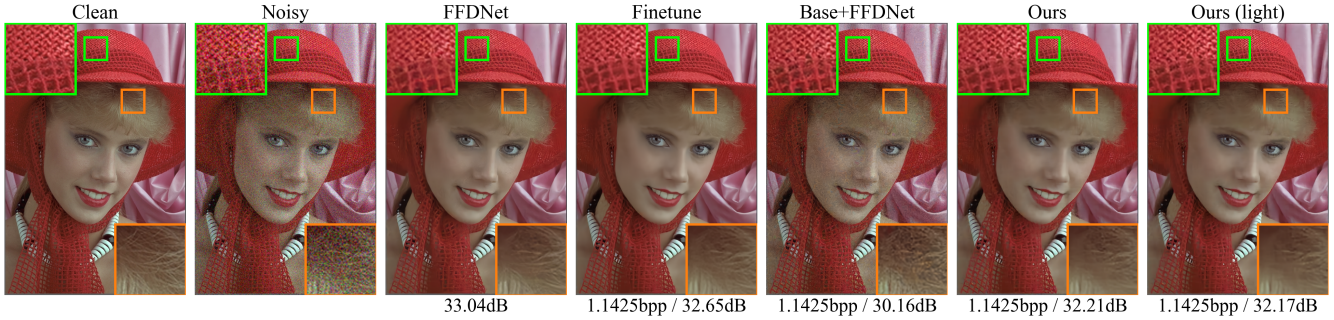
	kMACs/pixel	**Params (M)**
Base	188.52	3.87
LRM-light	189.84 (+ 1%)	4.21 (+ 9%)
LRM	192.27 (+ 2%)	4.83 (+25%)
Prompt	205.78 (+ 9%)	3.98 (+ 3%)
Prompt-heavy	285.26 (+51%)	7.15 (+84%)
LRM-light+Prompt	207.10 (+10%)	4.32 (+12%)
LRM+Prompt	209.53 (+11%)	4.94 (+28%)
LRM+Prompt-heavy	289.01 (+53%)	8.11 (+110%)

C. Ablation Experiment

Fig. 4(c) presents an ablation study of our add-on modules, LRM and the prompt generator g_p, for real-world noise. Note that these add-on modules together represent a two-pronged approach to joint decoding and denoising. LRM is a compressed-domain mechanism that aims to predict the latent representation of the denoised image from a noisy one; in comparison, g_p is to adapt the decoding process of a pre-trained base codec. Specifically, we investigate two LRM implementations, *LRM* and *LRM-light*, and two prompt generators, *Prompt* and *Prompt-heavy*. For prompt generators, the former is our proposed method and the latter is its heavy variant. With *Prompt-heavy*, the prompt generator uses convolutions instead of group convolutions, and it generates prompts for all STBs in the decoder in order to explore the full potential of adapting the decoding process only. We first note that all seven variants notably outperform *Base* in Fig. 4(a), showing their effectiveness in the denoising reconstruction. In addition, when enabling one module at a time, *LRM* proves more effective than *Prompt* or *Prompt-heavy* for the



(a) Real-world noise



(b) Synthetic noise

Fig. 5. Visualization of the noise-free, noisy, and denoised images. The top row are results with real-world noise. The second row are results with synthetic noise. Bits-per-pixel (bpp) / PSNR-RGB are provided for comparison. Zoom in for better visualization.

denoising reconstruction. *LRM-light* outperforms *Prompt* and shows slightly worse performance than *Prompt-heavy*. This is intuitively agreeable because the base codec is pre-trained for the standard image reconstruction. When *LRM* or *LRM-light* works well, the pre-trained decoder should recover a noise-free image easily. We also observe that *Prompt* and *Prompt-heavy* are able to improve further on LRMs. Table II reports their complexity characteristics. The combinations *LRM-light* + *Prompt* and LRM + *Prompt* strike a good balance between compression performance and complexity. The result justifies our design choices.

IV. Conclusion

This work introduces a Transformer-based image compression system that is capable of switching between the standard image reconstruction and denoising reconstruction without training separate decoders. It features an LRM module in the latent space and an instance-specific prompt generator. LRM is effective in refining the latents of a noisy input image for the denoising reconstruction. Our instance-specific prompt generator further complements LRM by adapting the decoding process to suit the needs of joint decoding and denoising. Both are add-on modules that have only a modest impact on the decoder's model size and computational complexity. How to extend the proposed framework to more challenging tasks, e.g. joint decoding and de-blurring, is among our future work.

References

[1] K. L. Cheng, Y. Xie, and Q. Chen, "Optimizing image compression via joint learning with denoising," in *European Conference on Computer Vision*. Springer, 2022, pp. 56–73.

[2] B. Brummer and C. De Vleeschouwer, "On the importance of denoising when learning to compress images," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 2440–2448.

[3] ISO/IEC JTC 1/SC29/WG1 N100095, REQ, "Final call for proposals for JPEG AI," 94th Meeting, Online, January 2022.

[4] L. Larigauderie, M. Testolina, and T. Ebrahimi, "On combining denoising with learning-based image decoding," in *Applications of Digital Image Processing XLV*, vol. 12226. SPIE, 2022, pp. 193–206.

[5] S. Ranjbar Alvar, M. Ulhaq, H. Choi, and I. V. Bajić, "Joint image compression and denoising via latent-space scalability," *Frontiers in Signal Processing*, vol. 2, p. 932873, 2022.

[6] M. Lu, P. Guo, H. Shi, C. Cao, and Z. Ma, "Transformer-based image compression," in *Data Compression Conference*, 2022.

[7] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 606–615.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[10] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1692–1700.

[11] J. Liu, G. Lu, Z. Hu, and D. Xu, "A unified end-to-end framework for efficient deep image compression," *arXiv preprint arXiv:2002.03370*, 2020.

[12] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5197–5206.

[13] "JPEG-AI Anchors," https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors.

[14] ISO/IEC JTC 1/SC29/WG1 N100600, "JPEG AI common training & test conditions v8.0," 100th Meeting, Covilhã, Portugal, July 2023.

[15] R. Franzen, "Kodak lossless true color image suite," *source: http://r0k.us/graphics/kodak*, vol. 4, no. 2, p. 9, 1999.

[16] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5728–5739.

[17] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.