# TransTIC: Transferring Transformer-based Image Compression from Human Perception to Machine Perception
## *Supplementary Materials*

Yi-Hsin Chen  Ying-Chieh Weng  Chia-Hao Kao  Cheng Chien
Wei-Chen Chiu  Wen-Hsiao Peng
National Yang Ming Chiao Tung University, Taiwan
{yhchen12101.cs09@, wengyc.cs09@, chiahaok.cs10@, cchien1999@cs.}nycu.edu.tw
{walon, wpeng}@cs.nctu.edu.tw

This supplementary document provides the following additional materials and results to assist with the understanding of our *TransTIC*:

- Implementation details in Section A1;
- Rate-accuracy comparison with the state-of-the-art traditional codec VVC in Section A2;
- More ablation experiments in Section A3;
- More qualitative results in Section A4.

## A1. Implementation Details

### A1.1. Perceptual Loss

To train the prompt generator network $g_p$ and the decoder-side prompts for downstream recognition tasks, the distortion measure $d(\cdot,\cdot)$ in Eq. (3) of the main paper is chosen to be the perceptual loss. Specifically, the perceptual loss is evaluated based on a pre-trained ResNet50 [2], Faster R-CNN [6] and Mask R-CNN [1] for classification, object detection and instance segmentation, respectively. Fig. A1 illustrates a ResNet50-based Feature Pyramid Network (FPN), which serves as the feature extractor in Faster R-CNN and Mask R-CNN. For the classification task, the perceptual loss is evaluated in the feature space of F1, F2, F3, and F4:

$$d(x,\hat{x}) = \frac{1}{4}\cdot\sum_{l=1}^{4} MSE(F_l(x), F_l(\hat{x})), \qquad (1)$$

where $x$ and $\hat{x}$ are the input and decoded images, respectively. For the tasks of object detection and instance segmentation, the perceptual loss is evaluated in the feature space of P2, P3, P4, P5, and P6:

$$d(x,\hat{x}) = \frac{1}{5}\cdot\sum_{l=2}^{6} MSE(P_l(x), P_l(\hat{x})). \qquad (2)$$

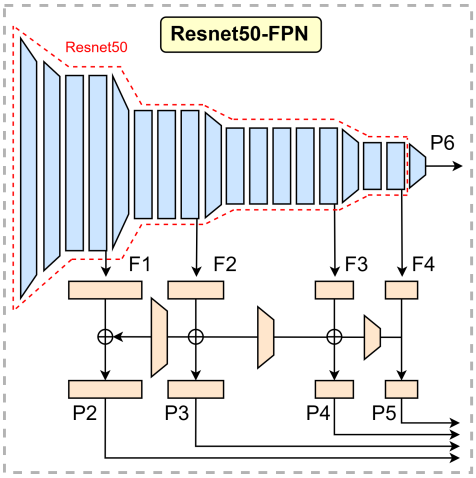



Figure A1. Architecture of Resnet50-based FPN, which shows the features selected for evaluating the perceptual loss.

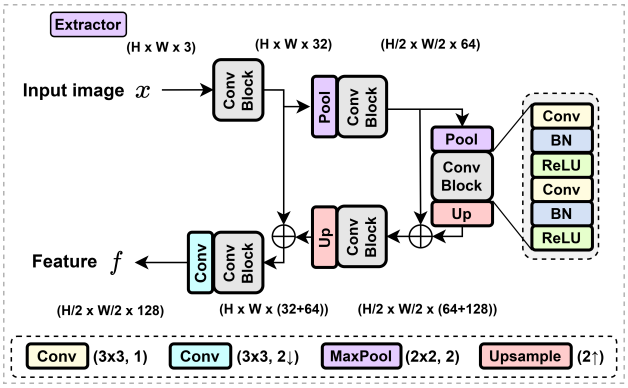



Figure A2. Architecture of the extractor in our prompt generator $g_p$.

In Fig. A1, the network weights are initialized using a separate pre-trained model, depending on the downstream task.

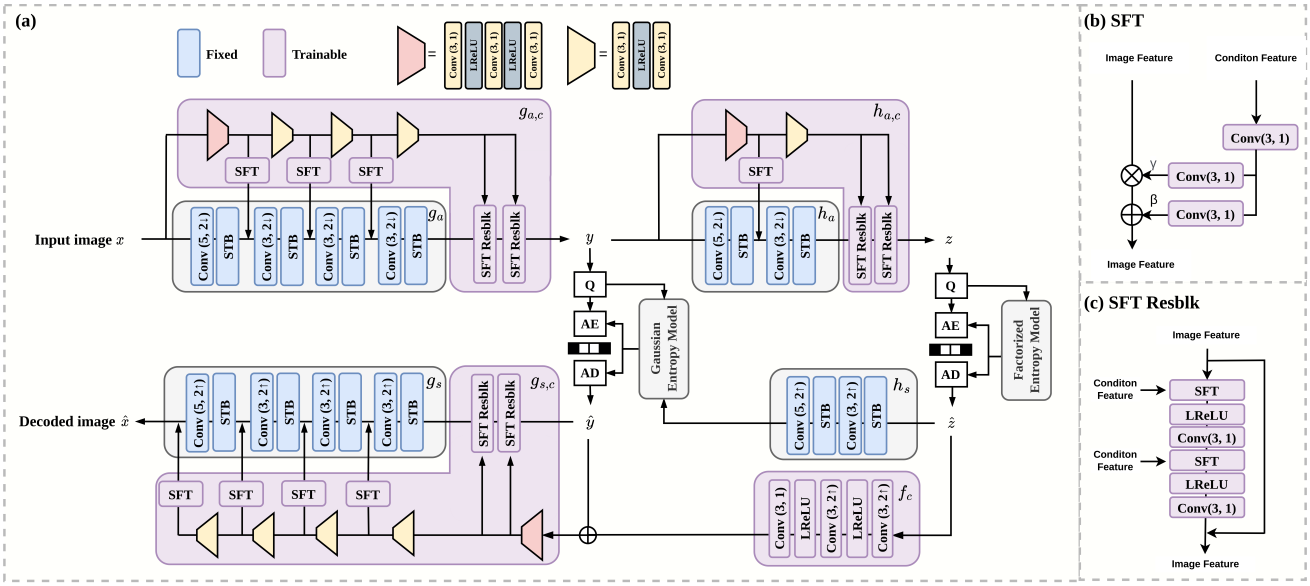



Figure A3. Architecture of *TIC+SFT*.

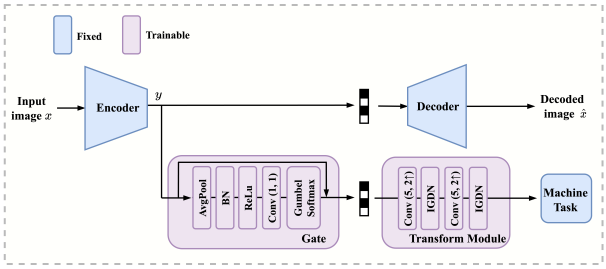



Figure A4. Architecture of *TIC+channel selection*.

### A1.2. Extractor in Prompt Generator

Fig. A2 details the network architecture of the extractor in our task-specific prompt generator $g_p$ (see Fig. 2(a) in the main paper). It has a U-Net [7]-like structure.

### A1.3. *TIC+SFT*

Fig. A3 depicts the network architecture of the baseline method *TIC+SFT* [9], which shares the same fixed pre-trained base codec (the parts in blue color) as our *TransTIC*. *TIC+SFT* utilizes spatial feature transform (SFT) layers to perform element-wise affine transformation of the feature maps in $g_a, g_s$, and $h_a$ for transferring the base codec from human perception to downstream machine tasks. It follows [8] in using convolutional neural networks to produce the element-wise affine parameters $\gamma, \beta$ for each SFT layer.

### A1.4. *TIC+channel selection*

Fig. A4 shows the architecture of *TIC+channel selection* [5]. Based on a pre-trained codec for human perception, *TIC+channel selection* introduces two additional task-specific modules for machine perception. As shown, a gate module first performs adaptive channel selection on the image latent $y$ through multiplying each of its channels by a binary value. Then, a transform module converts the masked image latent into a set of feature maps suitable for the downstream recognition network.

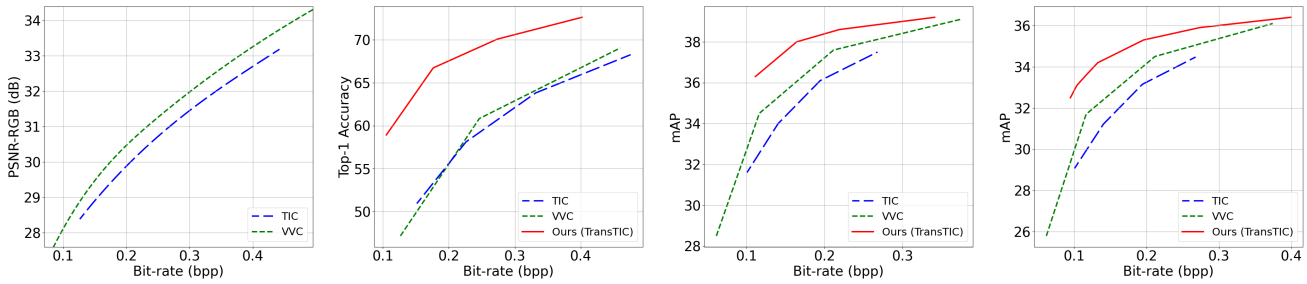
## A2. Comparison with VVC

Fig. A5 (a) compares our base codec, *TIC*, with the state-of-the-art traditional codec VVC (VTM 16.0 intra coding) on the standard image compression task (i.e. for human perception). The dataset is Kodak [4]. As shown, *TIC* shows worse PSNR results than VVC on the standard reconstruction task. It is thus not surprising to see that *TIC* performs worse than VVC on the remaining recognition tasks. However, based on *TIC*, our *TransTIC* achieves much better rate-accuracy performance than VVC (Fig. A5 (b)(c)(d)). This result confirms the effectiveness of our prompting technique.

## A3. More Ablation Experiments

### A3.1. Prompt Injection: Deep vs. Shallow

This ablation experiment tests another variant of prompt injection. Our *TransTIC* injects prompts to every Swin-Transformer layer in an IP-type or TP-type STB, which is similar to VPT-Deep in [3]. Another possible way of injecting prompts is to insert them only at the first Swin-Transformer layer of a STB. These prompts are also updated in the multi-head self-attention step. This setting is analogous to VPT-shallow in [3]. The architectural difference between *Deep* and *Shallow* is shown in Fig. A6. From Fig. A7, *Deep* performs comparably to *Shallow* on the classification




(a) Human Perception (b) Classification (c) Object Detection (d) Instance Segmentation

Figure A5. Performance comparison between our *TransTIC* and VVC under various tasks.

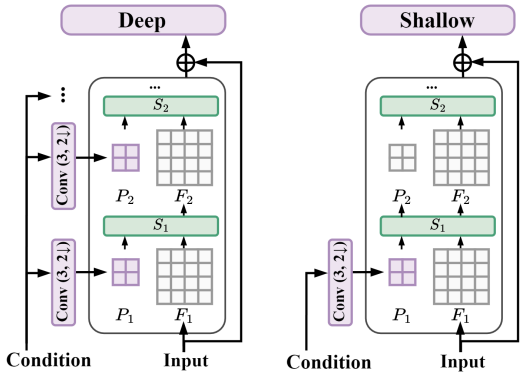
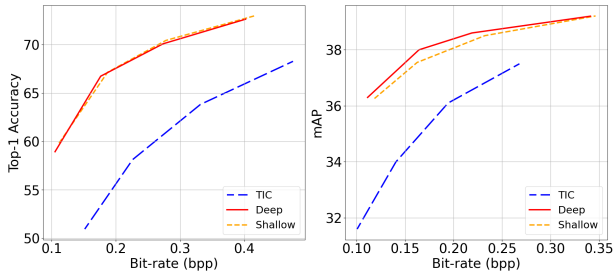



Figure A6. Architecture comparison of *Deep* and *Shallow* IP-type STB.




(a) Classification (b) Object Detection

Figure A7. Ablation on prompt injection: *Deep* vs. *Shallow*.

task, and performs slightly better than *Shallow* on the detection task. In Table A1, *Deep* has comparable kMAC/pixel and model size to *Shallow*. We thus choose *Deep* in our *TransTIC* for its better rate-accuracy performance.

## A3.2. IP-type STBs in the Decoder

This ablation study introduces IP-type STBs to the decoder. Currently, our *TransTIC* uses only TP-type STBs in the decoder because the input image is not accessible on the decoder side. One alternative to constructing IP-type STBs on the decoder side is to utilize the decoded latent $\hat{y}$ to generate instance-specific prompts (Fig. A8)). From Fig. A9, we see that introducing such IP-type STBs to the decoder

Table A1. Comparison of the kMACs/pixel and model size. **Bold** indicates our final design choices.

| Section | Method | kMACs/pixel Encoder | kMACs/pixel Decoder | Params (M) Encoder | Params (M) Decoder |
|---|---|---|---|---|---|
| | *TIC* | 142.54 | 188.52 | 3.65 | 3.86 |
| A3.1 | *Shallow* | 322.80 | 209.51 | 4.65 | 3.88 |
| | ***Deep*** | 332.03 | 202.60 | 5.24 | 3.89 |
| A3.2 | Enc: IP, Dec: IP | 332.03 | 276.39 | 5.24 | 5.06 |
| | **Enc: IP, Dec: TP** | 332.03 | 202.60 | 5.24 | 3.89 |
| A3.3 | 4 prompts | 302.06 | 192.04 | 5.24 | 3.87 |
| | **16 prompts** | 332.03 | 202.60 | 5.24 | 3.89 |
| | 64 prompts | 451.91 | 244.87 | 5.24 | 3.98 |
| A3.4 | **STB-1234** | 332.03 | 202.60 | 5.24 | 3.89 |
| | STB-12 | 332.03 | 200.80 | 5.24 | 3.87 |
| | STB-34 | 332.03 | 190.32 | 5.24 | 3.88 |
| A3.5 | Enc: IP, Dec: - | 332.03 | 188.52 | 5.24 | 3.86 |
| | Enc: - , Dec: TP | 142.54 | 202.60 | 3.65 | 3.89 |
| | **Enc: IP, Dec: TP** | 332.03 | 202.60 | 5.24 | 3.89 |

improves the rate-accuracy performance on the classification task, but performs comparably to TP-type STBs on the object detection task. From Table A1, as compared to TP-type STBs, IP-type STBs lead to a 36% increase in the decoder's kMACs/pixel and a 30% increase in the decoder's model size. Because low decoding complexity and small decoder size are of importance, we choose to use TP-type STBs in the decoder.

## A3.3. Prompt Numbers

Fig. A10 ablates the effect of the number of prompts used in a Swin-Transformer window. When the number of prompts decreases from 64 to 4, the rate-accuracy performance drops marginally on the more complicated detection task. According to Table A1, the kMACs/pixel and model size of the model with 16 prompts is close to those of the model with 4 prompts. We thus choose 16 prompts to strike a balance between the rate-accuracy performance and model complexity.

## A3.4. Prompt Depth of the Decoder

Fig. A11 analyzes which and how many STBs to inject prompts on the decoder side. As shown, injecting task-

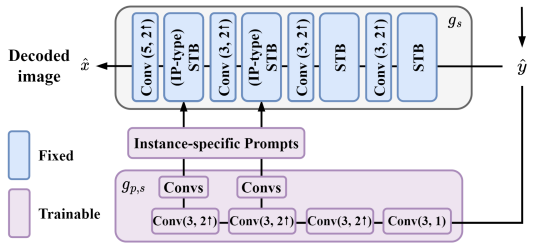
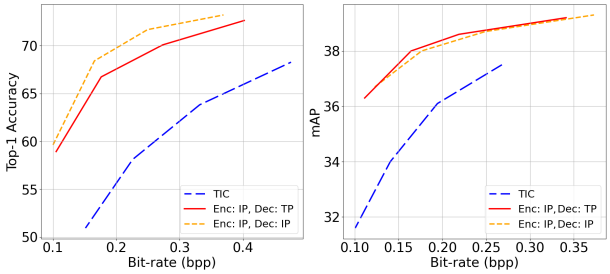
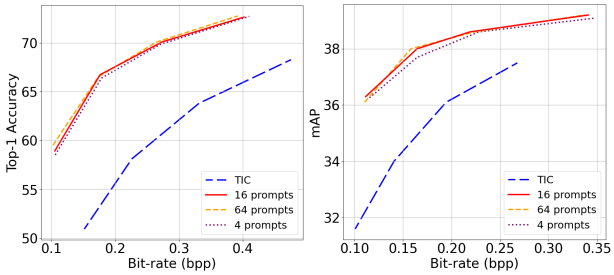



Figure A8. Architecture of IP-type STBs in the decoder.




(a) Classification (b) Object Detection

Figure A9. Ablation on decoder side STB.



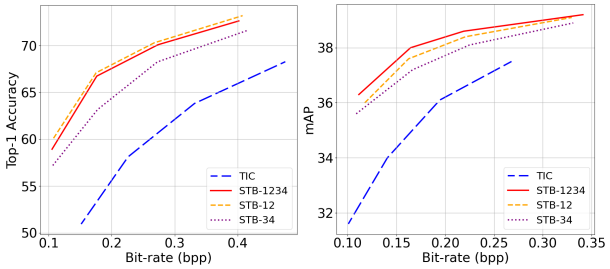

(a) Classification (b) Object Detection

Figure A10. Ablation on the number of prompts.

specific prompts to all STBs (STB-1234) appears to be a better choice than the other variants, namely, STB-12 and STB-34, in terms of the rate-accuracy performance. STB-12 refers to injecting prompts to the two STBs closer to the decoded image $\hat{x}$ while STB-34 refers to injecting them to STBs closer to the image latent. From Fig. A11, STB-12 performs better than STB-34. Because STB-1234 has only slightly higher kMAC/pixel and model size than STB-12 (Table A1), we choose STB-1234 as our final design.

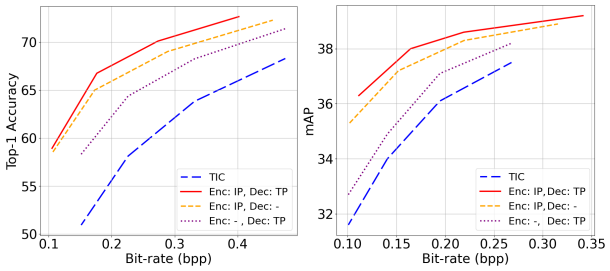
### A3.5. Prompting Encoder vs. Decoder

Fig. A12 compares the effectiveness of introducing IP-type STBs to the encoder and TP-type STBs to the decoder. As shown, introducing prompts to both the encoder and decoder achieves the best rate-accuracy performance. We also see that prompting on the encoder side is more effective than prompting on the decoder side. This result is intuitively agreeable because prompting on the encoder side allows the compressed bitstream to be tailored for the downstream




(a) Classification (b) Object Detection

Figure A11. Ablation on the prompt depth of the decoder.




(a) Classification (b) Object Detection

Figure A12. Ablation on effectiveness of prompt on encoder and decoder sides.

task. The complexity characteristics of these variants are provided in Table A1.

## A4. More Qualitative Results

Fig. A13, Fig. A14, and Fig. A15 provide more qualitative results, comparing the decoded images and the bit allocation maps produced by the competing methods. As shown, *TIC*, the codec optimized for human perception, tends to allocate more bits to complex regions, even if those regions are less relevant (e.g. background) to the downstream recognition tasks. In contrast, the other methods, which target machine perception, attempt to shift coding bits from the background regions to the foreground objects.

## References

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022.

[4] E. Kodak. Kodak lossless true color image suite (photocd pcd0992). http://r0k.us/graphics/kodak/.

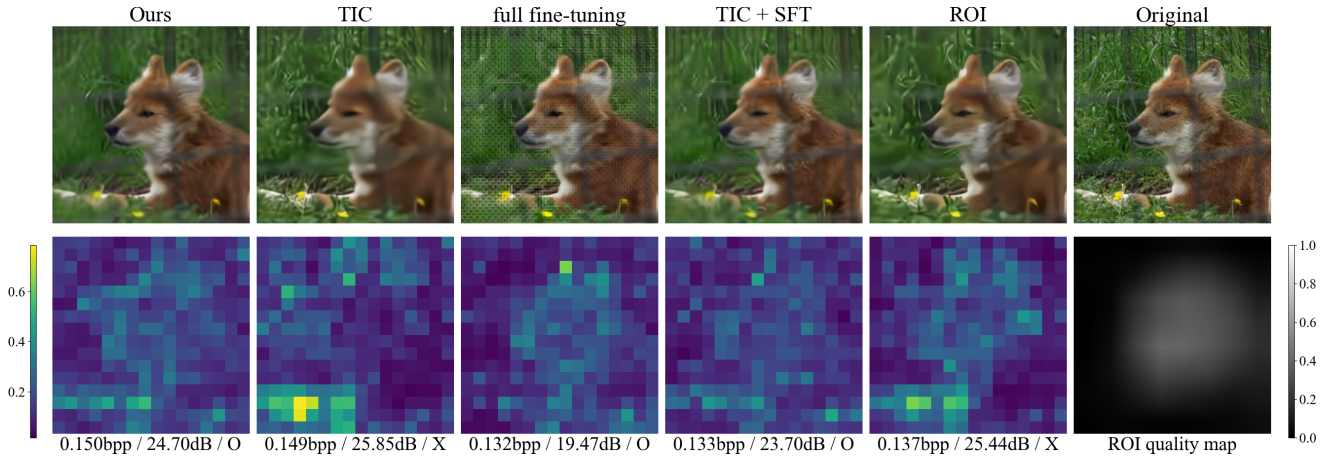
[5] Jinming Liu, Heming Sun, and Jiro Katto. Improving multiple machine vision tasks in the compressed domain. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 331–337. IEEE, 2022.

[6] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

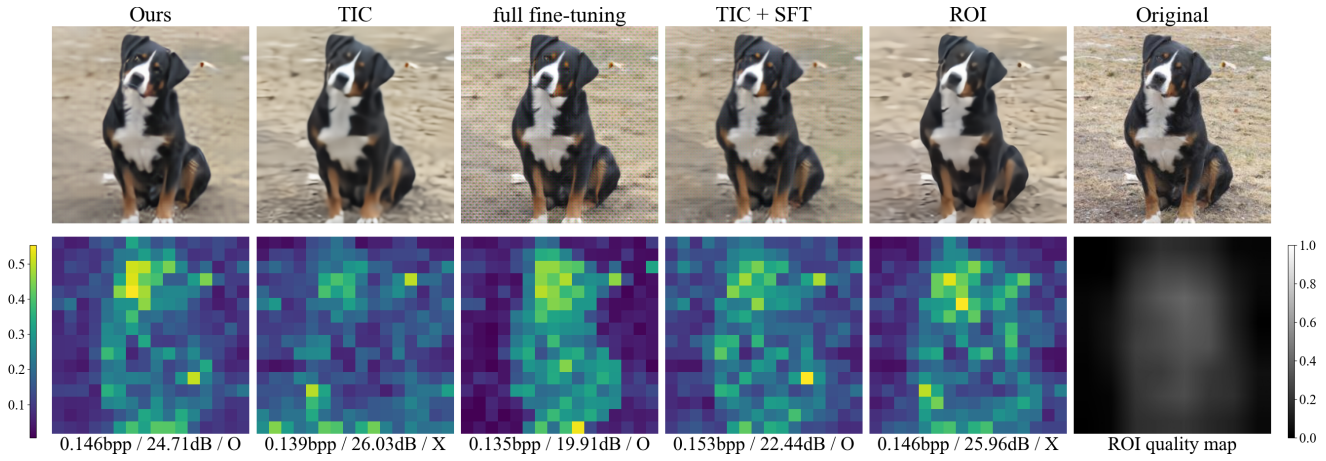
[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[8] Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2380–2389, 2021.

[9] Chao Dong Xintao Wang, Ke Yu and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

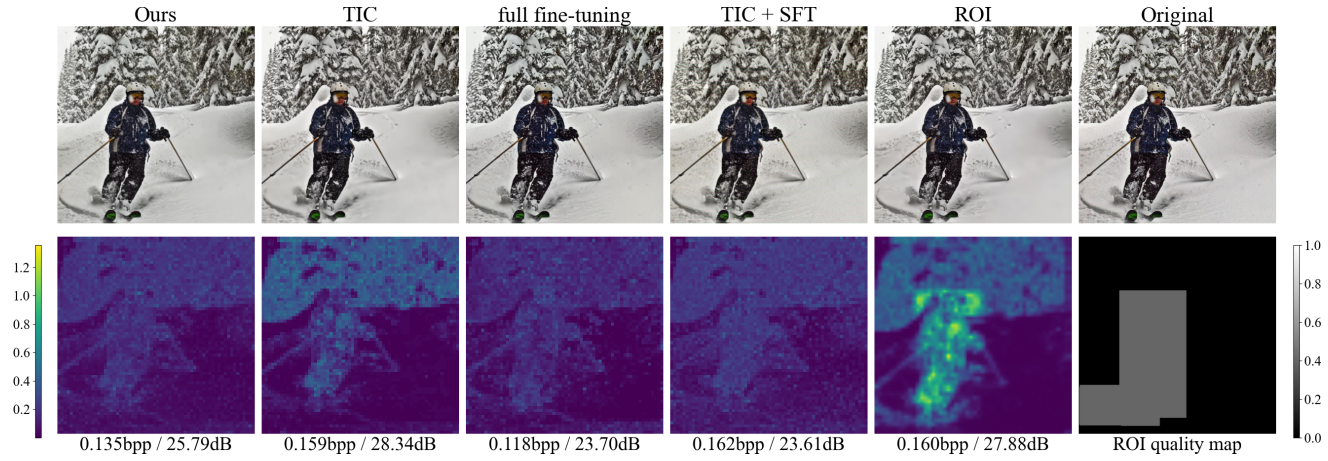



(a)

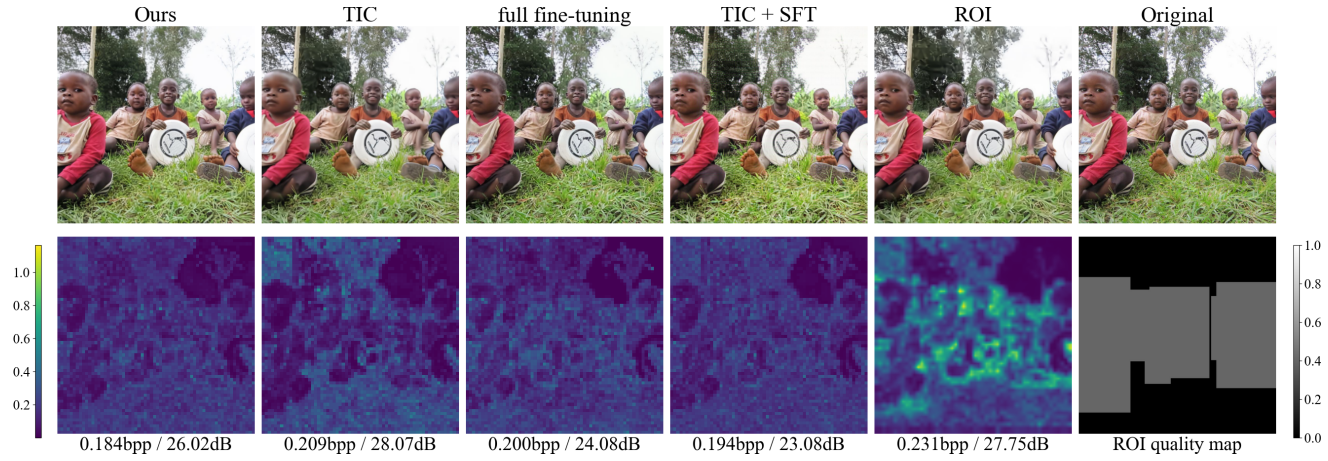



(b)

Figure A13. Visualization of the decoded images (the first row) and the bit allocation maps (the second row) of the image latent $\hat{y}$ for the classification task. The rightmost image of the second row shows the quality map used for the ROI method. The text below each map denotes the corresponding bit rate / PSNR / prediction result, with O and X indicating correct and false classification, respectively.

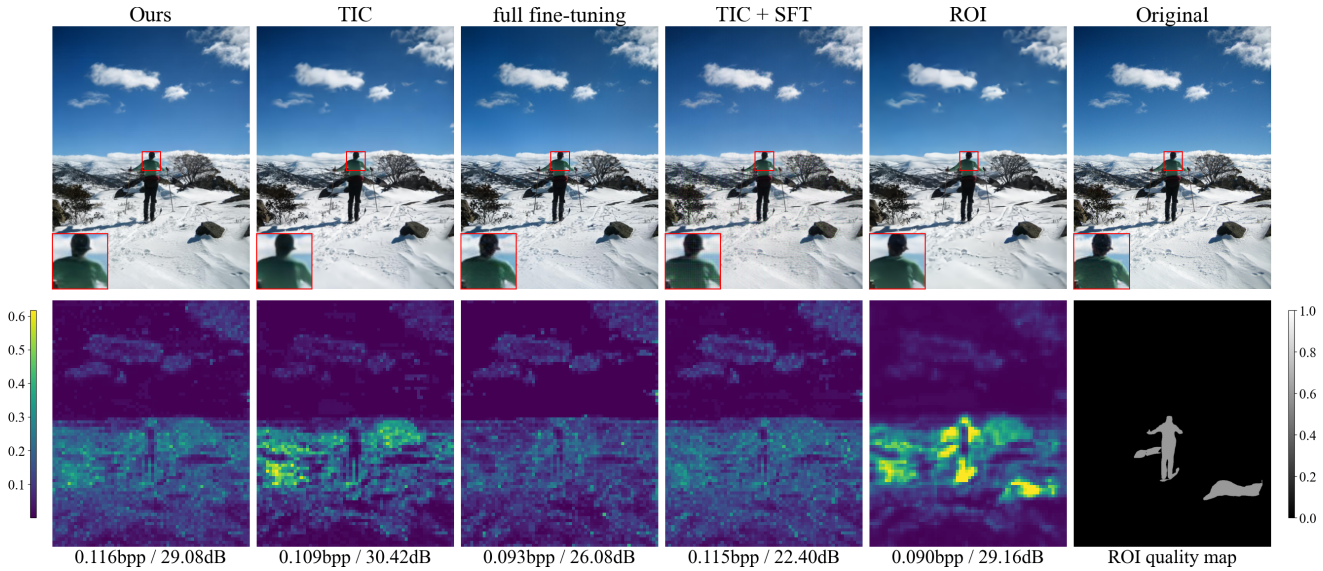



(a)

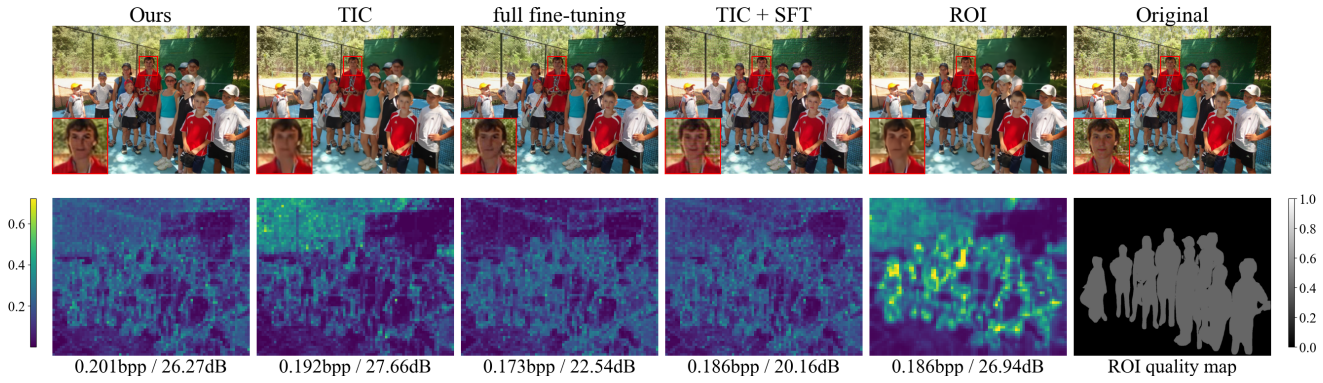



(b)

Figure A14. Visualization of the decoded images (the first row) and the bit allocation maps (the second row) of the image latent $\hat{y}$ for the object detection task. The rightmost image of the second row shows the quality map used for the ROI method. The text below each map denotes the corresponding bit rate / PSNR.




(a)




(b)

Figure A15. Visualization of the decoded images (the first row) and the bit allocation maps (the second row) of the image latent $\hat{y}$ for the instance segmentation task. The rightmost image of the second row shows the quality map used for the ROI method. The text below each map denotes the corresponding bit rate / PSNR.