TransTIC: Transferring Transformer-based Image Compression from Human Perception to Machine Perception

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

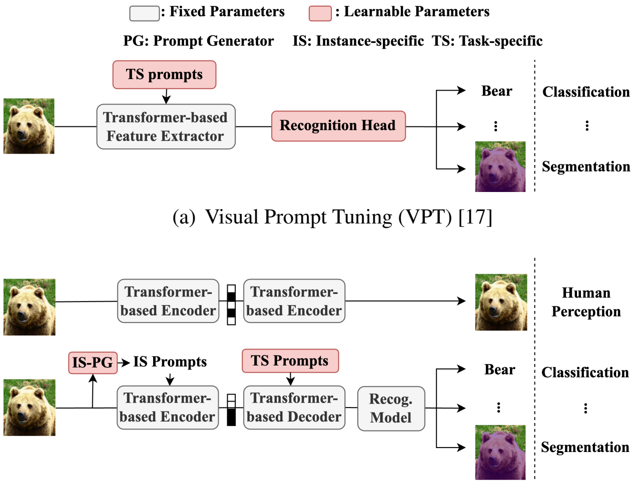
Abstract

This work aims for transferring a Transformer-based image compression codec from human perception to machine perception without fine-tuning the codec. We propose a transferable Transformer-based image compression framework, termed TransTIC. Inspired by visual prompt tuning, TransTIC adopts an instance-specific prompt generator to inject instance-specific prompts to the encoder and task-specific prompts to the decoder. Extensive experiments show that our proposed method is capable of transferring the base codec to various machine tasks and outperforms the competing methods significantly. To our best knowledge, this work is the first attempt to utilize prompting on the low-level image compression task.

1. Introduction

End-to-end learned image compression systems [8, 12, 44] have recently attracted lots of attention due to their competitive compression performance to traditional image coding methods, such as intra coding in VVC [4] and HEVC [41]. Among them, transformer-based autoencoders [47, 32, 31, 42] emerge as attractive alternatives to convolutional neural networks (CNN)-based solutions because of their high content adaptivity. Some even feature lower computational cost than CNN-based autoencoders. In common, most learned image compression systems are designed primarily for human perception.

Recently, image coding for machine perception becomes an active research area due to the rising demands for transmitting visual data across devices for high-level recognition tasks. Coding techniques in this area mainly include approaches that produce multi-task or single-task bitstreams. The methods with the multi-task bitstream feature one single compressed bitstream that is able to serve multiple downstream tasks, such as human perception and machine perception. Most of them [6, 11, 9, 28, 46] aim to learn a robust, general image representation via multi-task or contrastive learning. However, a general bitstream can hardly be rate-distortion optimal from the perspective of each individual task.



(a) Visual Prompt Tuning (VPT) [17]

(b) Ours (*TransTIC*)

Figure 1. Comparison of VPT [17] and our proposed TransTIC.

The methods with the single-task bitstream allow the image codec to be tailored for individual downstream tasks, thereby generating multiple task-specific bitstreams. One straightforward approach is to optimize a codec end-to-end for each task [5]. However, given the sheer amount of machine tasks and their recognition networks, along with the ongoing developments of new machine tasks and models, customizing a neural codec, particularly hardware-based, for every one-off machine application would be prohibitively expensive even if not impossible. Region-of-interest (ROI)-based [40] and transferring-based methods [27] are two preferred solutions. The former performs spatially adaptive coding of images according to an importance map, which can be optimized for different downstream tasks. The latter aims to transfer a pre-trained base codec to a new task without changing the base codec. However, how to transfer efficiently a given codec without retraining is a largely under-explored topic.

In this work, we aim to transfer a well-trained Transformer-based image codec from human perception to machine perception *without* fine-tuning the codec. Inspired by Visual Prompt Tuning (VPT) [17], we propose a plug-in mechanism, which injects additional learnable inputs, known as prompts, to the fixed base codec. As shown in Fig. 1 (a), VPT [17] targets re-using a large-scale, pre-trained Transformer-based feature extractor on different recognition tasks. This is achieved by injecting a small amount of task-specific learnable parameters, prompts, to the Transformer-based feature extractor and learning a task-specific recognition head. Different from VPT [17], which considers only the performance of the downstream recognition task, our task focuses on image compression, which needs to strike a balance between the downstream task performance and the transmission cost (i.e. the bitrate needed to signal the bistream). Fig. 1 (b) sketches the high-level design of our proposed method. As shown, the Transformer-based encoder and decoder are initially optimized for human perception while the recognition model is an off-the-shelf recognition network. To transfer the codec from human perception to machine perception, we inject prompts to both the encoder and decoder. On the encoder side, we introduce an instance-specific prompt generator to generate instance-specific prompts by observing the input image. On the decoder side, the input image is not accessible. We thus introduce task-specific prompts to the decoder.

Our main contributions are four-fold:

- Without fine-tuning the codec, we transfer a well-trained Transformer-based image codec from human perception to machine perception by injecting instance-specific prompts to the encoder and task-specific prompts to the decoder.
- To the best of our knowledge, this work is the first attempt to utilize prompting techniques on the low-level image compression task.
- The plug-in nature of our method makes it easy to integrate any other Transformer-based image codec.
- Our proposed method is capable of transferring the codec to various machine tasks. Extensive experiments show that our method achieves better rate-accuracy performance than the other transferring-based methods on complex machine tasks.

2. Related Work

2.1. Learned Image Compression

End-to-end learned image compression systems have recently attracted lots of attention due to their competitive compression performance to traditional codecs, such as VVC [4] and HEVC [41]. Most of them adopt CNN-based autoencoders [2, 7, 8, 16, 35] with a hyperprior entropy model. Due to the great success of Transformers in high-level vision tasks, some studies [47, 32] start to explore their application to low-level vision tasks, such as image compression. In [47], Zhu *et al.* [47] construct an autoencoder using Swin-Transformers [30], achieving comparable compression performance to the state-of-the-art CNN-based solutions with lower computational complexity. Lu *et al.* [32] improves on [47] by replacing patch merging/splitting in Swin-Transformers [30] with the more general convolutional layers. Lu *et al.* [31] further adopt Neighborhood Attention Transformers [13] in place of Swin-Transformers for their more efficient sliding-window attention mechanism. Transformers also find applications in building efficient entropy coding models [19, 43, 34].

2.2. Compression for Machine Perception

The success of neural networks in both high-level and low-level vision tasks opens up a new research area known as compression for machine perception. That is, the compressed image features or the decoded image should be suitable for the downstream recognition tasks. We divide recent works in this emerging research area into three categories according to the characteristics of their coded bitstreams, namely, mutli-task bitstreams, scalable bitstreams and single-task bitstreams.

Mutli-task bitstreams. The methods [6, 11, 21] in this category generate one single compressed bitstream to serve the needs of multiple downstream tasks, such as human perception and machine perception. A straightforward approach is to train an image codec through multi-task learning [21, 6]. More recently, Feng *et al.* [11] utilize contrastive learning to learn a general image representation for various downstream vision tasks. An obvious disadvantage of these methods is that a single, multi-purpose bitstream can hardly be rate-distortion optimal for individual downstream tasks. Whether such a bitstream can generalize to unseen tasks is an open issue.

Scalable bitsreams. There are also approaches [9, 28, 46] that aim to generate a scalable bitstream, which can be partially decoded for simpler machine tasks or fully decoded to reconstruct the input image for human perception. However, how to arrange potentially multiple image representations for various tasks in a layered manner without introducing redundancy is challenging.

Single-task bitstreams. Different from the previous two approaches, the methods that produce single-task bitstreams allow the codec to be tailored for each individual task. A common approach is to fine-tune a pre-trained codec for a specific downstream task [26, 33]. However, each task requires a separate model and a machine recognition task normally has a variety of recognition networks to choose from. Customizing a neural codec for each possible choice can be prohibitively expensive, particularly when the neural codec is implemented on specific hardware accelerators. In comparison, region-of-interest (ROI) coding presents a versatile coding solution. For example, Song *et al.* [40] propose an image codec capable of encoding an input image in a spatially adaptive way according to an importance map. The importance map can be determined at inference time to optimize the decoded image for different uses, e.g. spatial bit allocation or machine perception. Another interesting approach is presented in [27], which introduces a learnable task-specific gate module to channel-wisely selects image latents produced by a pre-trained codec for compression. Along a similar line of thinking, we propose a novel idea of transferring a pre-trained base codec to different tasks. This is achieved by introducing additional task-specific modules to adapt the base codec without changing its network weights. Specifically, we utilize prompting techniques to transfer a Transformer-based image codec from human perception to machine perception.

2.3. Prompt Tuning

The idea of prompting [22, 23, 29] was first brought up in the field of neural language processing. While transformer-based language models have been a huge success in many language tasks, fine-tuning a pre-trained, large-scale transformer model for a specific downstream task requires huge effort. Prompt tuning offers an attractive alternative, which modifies the input of text encoders while keeping their backbones untouched. Jia *et al.* [17] are the first to extend this approach to computer vision tasks via inserting learnable task-specific prompts to the input of the vision transformer layers. With only a small number of trainable parameters, it manages to achieve comparable or even superior performance to full fine-tuning in downstream recognition tasks.

3. Proposed Method

Given a Transformer-based image codec well-trained for human perception, i.e., the image reconstruction task, this work aims for transferring the codec to achieve image compression for machine perception (e.g., object detection) *without* fine-tuning the codec. We draw inspiration from [17] to propose a plug-in mechanism that utilizes prompting techniques to transfer the codec. However, unlike [17], which addresses how to adapt a pre-trained large-scale Transformer-based recognition model to different recognition tasks, our work focuses on transferring a pre-trained Transformer-based image compression model to tailor image compression for recognition tasks, an application also known as image compression for machines. As such, our performance metric involves not only the recognition accuracy but also the bit rate (in terms of bits-per-pixel) needed to signal the compressed image. In our case, the downstream recognition model can be Transformer-based or convolutional neural network-based.

In Section 3.1, we first address a classic paradigm of end-to-end learned image compression. In Section 3.2, we outline our proposed framework. This is followed by details of our transferring mechanism in Section 3.3 and the training objective in Section 3.4.

3.1. Preliminaries

An end-to-end learned image compression system usually has two major components: a main autoencoder and a hyperprior autoencoder. The main autoencoder consists of an analysis transform (g_a in Fig. 2) and a synthesis transform (g_s in Fig. 2). The analysis transform g_a encodes an RGB image $x \in \mathbb{R}^{H \times W \times 3}$ of height H and width W into its latent representation $y \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 192}$ by an encoding distribution $q_{g_a}(y|x)$. The latent y is then uniformly quantized as $\hat{y}$ and entropy encoded into a bitstream by a learned prior distribution $p(\hat{y})$. On the decoder side, $\hat{y}$ is entropy decoded and reconstructed as $\hat{x} \in \mathbb{R}^{H \times W \times 3}$ through a decoding distribution $q_{g_s}(\hat{x}|\hat{y})$ realized by the synthesis transform g_s. In the process, the prior distribution $p(\hat{y})$ crucially affects the number of bits needed to be signal the quantized latent $\hat{y}$. It is thus modelled in a content-adaptive manner by a hyperprior autoencoder [2], which comprises a hyperprior analysis transform (h_a in Fig. 2) and a hyperprior synthesis transform (h_s in Fig. 2). As illustrated, h_a converts the image latent y into the side information $z \in \mathbb{R}^{\frac{H}{64} \times \frac{W}{64} \times 128}$, representing typically a tiny portion of the compressed bitstream. Its quantized version is decoded from the bitstream through h_s to arrive at $p(\hat{y})$. Notably, this work considers a particular implementation of the main and hyperprior autoencoders, the backbones of which are Transformer-based.

3.2. System Overview

Fig. 2 illustrates our transferable Transformer-based image compression framework, termed *TransTIC*. It is built upon [32], except that the context prior model is replaced with a simpler Gaussian prior for entropy coding. As shown, the main autoencoder g_a, g_s and the hyperprior autoencoder h_a, h_s include Swin-Transformer blocks (STB) as the basic building blocks. These STB are interwoven with convolutional layers to adapt feature resolution in the data pipeline. In this work, the main and hyperprior autoencoders are pre-trained for human perception (i.e. the image reconstruction task) and their network weights are fixed during the transferring process.

To transfer g_a, g_s such that the decoded image $\hat{x}$ is suitable for machine perception, we inject (1) instance-specific

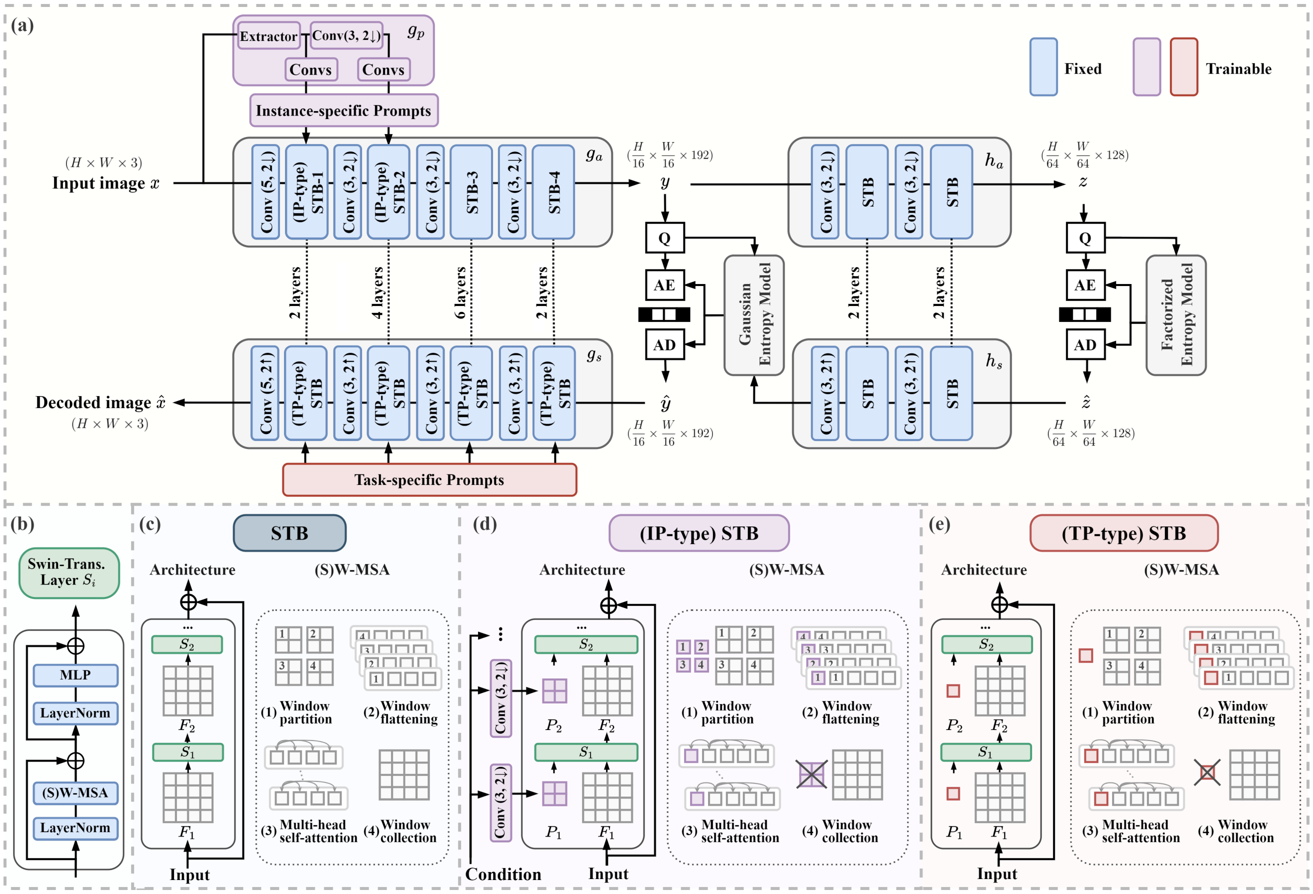


Figure 2. Overall architecture of our proposed method *TransTIC* and the detailed design of STBs.

prompts produced by g_p into the first two STBs in g_a and (2) task-specific prompts into all the STBs in g_s. Section 3.3 details how these additional prompts are input to these STBs. We note that the prompt generator g_p and the task-specific prompts input to the decoder are learnable and updated according to the machine perception task. That is, the network weights of g_p are task-specific. However, the prompts produced by g_p are instance-specific because they are dependent on the input image. Section 4.4 presents ablation experiments to justify our design choices.

3.3. Prompting Swin-Transformer Blocks

Swin-Transformer blocks (STB). STB is at the very core of our design. Fig. 2 (c) details its data processing pipeline. It consists of multiple (e.g. M) Swin-Transformer layers (Fig. 2 (b)), with the odd-numbered layers implementing window-based multi-head self-attention (W-MSA) and the even-numbered layers realizing shifted W-MSA (SW-MSA) to facilitate cross-window information exchange. In a Swin-Transformer layer, the input is a set of tokens, each representing a feature vector at a specific spatial location. As shown, the operation of W-MSA (or SW-MSA) has four sequential steps: (1) *window partition* that divides evenly an input feature map $F_i \in \mathbb{R}^{h_i \times w_i \times c_i}, i = 1, 2, \ldots, M$, in the i-th Swin-Transformer layer into non-overlapping windows, (2) *window flattening* that flattens the feature map along the token dimension in each window, (3) *multi-head self-attention* that adaptively updates tokens in each window through self-attending to tokens of the same window, and (4) *window collection* that unflattens the updated tokens in each window and collect tokens from all the windows to form an updated feature map of dimension the same as F_i. In symbols, the self-attention in a window for a specific head is given by

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

where $Q = FW_Q$, $K = FW_K$, $V = FW_V$, and $F \in \mathbb{R}^{N \times d}$ is the flattened feature map with N denoting the number of tokens in the window and d the dimension of each token. $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are learnable matrices projecting the input F into query $Q \in \mathbb{R}^{N \times d}$, key $K \in \mathbb{R}^{N \times d}$ and value $V \in \mathbb{R}^{N \times d}$. $B \in \mathbb{R}^{N \times N}$ is a learnable relative position bias matrix.

Transferring Encoding STBs via Instance-specific Prompting. To transfer g_a, g_s to machine perception, we propose to inject additional learnable tokens, known as prompts, into the STBs. They interact with the corresponding input tokens in the pre-trained STBs to adapt the encoding process and the decoded image, in order to achieve better rate-accuracy performance for machine perception. There are two types of prompts: instance-specific prompts and task-specific prompts. Instance-specific prompts are dependent on the input image, while task-specific prompts are task dependent yet invariant to the input image. As shown in Fig. 2 (a), on the encoder side, we introduce an instance-specific prompt generator g_p that generates instance-specific prompts for the first two STBs (which are referred to as IP-type STBs) based on the input image. g_p itself is task-specific because its network weights are trained for a specific downstream machine task. The network details of g_p are provided in the supplementary document. Fig. 2 (d) depicts the inner workings of IP-type STBs. They operate similarly to the ordinary STB without prompting, except that additional and separate prompts, denoted collectively as $P_i \in \mathbb{R}^{\frac{h_i}{4} \times \frac{w_i}{4} \times c_i}$, are introduced in the i-th Swin-Transformer layer. In particular, P_i for a specific layer has a spatial resolution that is a quarter of that of F_i. This design choice is meant to strike a balance between compression performance and complexity. Morover, P_i is partitioned and flattened in the same way as the image feature F_i. In the self-attention step for a specific window and head, the prompts of the same window update the image tokens with the query Q, key K and value V matrices in Eq. (1) augmented as follows:

$$\begin{aligned} Q &= FW_Q, \\ K &= [F; P]W_K, \\ V &= [F; P]W_V, \end{aligned} \tag{2}$$

where $P \in \mathbb{R}^{\frac{N}{4} \times d}$ refers collectively to the prompts in the same window as the image tokens F and $[\cdot;\cdot]$ indicates concatenation along the token dimension. With the same W_Q, W_K, W_V as for Eq. (1), we have $Q \in \mathbb{R}^{N \times d}$, $K \in \mathbb{R}^{(N+\frac{N}{4}) \times d}$ and $V \in \mathbb{R}^{(N+\frac{N}{4}) \times d}$. In the window collection step, only image tokens are collected while prompt tokens are discarded.

Transferring Decoding STBs via Task-specific Prompting. Similarly, we introduce prompts to the STBs in the decoder. Unlike the encoder, the decoder adopts task-specific prompts because the input image is unavailable on the decoder side and communicating instance-specific prompts to the decoder incurs extra overhead. Specifically, these task-specific prompts are input to every STB (referred to as TP-type STB) in the decoder. Fig. 2 (e) illustrates the operation of TP-type STBs. Similar to the IP-type STBs in the encoder, the TP-type STBs are prompted with separate tokens $P_i \in \mathbb{R}^{\frac{N}{4} \times c}$ in different Swin-Transformer layers. However, within a Swin-Transformer layer, the same prompts are shared across fixed-size windows for window-based multi-head self-attention (see window flattening and multi-head self-attention). In other words, in Eq. (2) when applied to the decoder, P is set to P_i for different windows. This is limited by the fact that the number of fixed-size windows is variable depending on the image size. Training window- and task-specific prompts requires learning a variable number of prompts, which is infeasible.

3.4. Training Objective

In Fig. 2, the prompt generator network g_p and the task-specific prompts on the decoder side are learnable while the base codec (i.e. g_a, g_s, h_a, h_s) is fixed. We construct the training objective in the same way as learning an image compressor. That is, the training involves minimizing a rate-distortion cost

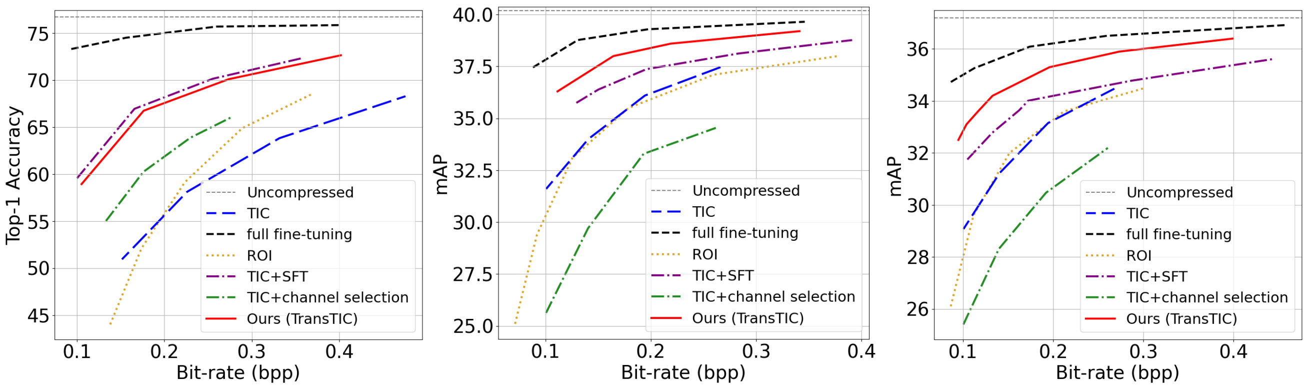
$$\mathcal{L} = \underbrace{-\log p(\hat{z}) - \log p(\hat{y}|\hat{z})}_{R} + \lambda \underbrace{d(x, \hat{x})}_{D}, \tag{3}$$

where R is the estimated rate needed to signal the quantized latent $\hat{y}$ and side information $\hat{z}$, D characterizes the distortion between the input image x and its reconstruction $\hat{x}$, and λ is a hyper-parameter. For the sake of machine perception, the distortion measure is perceptual loss, which is evaluated with a recognition network depending on the downstream task. More details can be found in the supplementary document.

4. Experimental Results

Training Details and Datasets. We evaluate our method on three machine tasks: classification, object detection, and instance segmentation. We first follow [32] to train the base codec composed of g_a, g_s, h_a, h_s on Flicker2W [25] for human perception. In this case, the distortion measure $d(\cdot,\cdot)$ in Eq. (3) is mean-squared error, and λ is chosen to be 0.0018, 0.0035, 0.0067, and 0.013 to arrive at four separate codecs for variable-rate compression. By freezing these pre-trained base codecs, we then train the prompt generator g_p together with the task-specific decoder-side prompts on ImageNet-*train* [10] for classification and on COCO2017-*train* [24] for object detection and instance segmentation. In the process, $d(\cdot,\cdot)$ in Eq. (3) is evaluated based on the perceptual loss using a pre-trained ResNet50 [15], Faster R-CNN [36] and Mask R-CNN [14] for classification, object detection and instance segmentation, respectively.

Evaluation. For classification, we use ImageNet-*val* [10] as the test set and a pre-trained ResNet50 [15] as the downstream recognition network. For object detection and instance segmentation, we test the competing methods on



(a) Classification (b) Object Detection (c) Instance Segmentation

Figure 3. Rate-accuracy performance comparison under different machine tasks.

Table 1. BD-Rate, BD-accuracy, and BD-mAP comparison under different machine tasks with *TIC* as anchor.

	Classification		**Detection**		**Segmentation**	
	BD-Rate (%) ↓	BD-accuracy ↑	BD-Rate (%) ↓	BD-mAP ↑	BD-Rate (%) ↓	BD-mAP ↑
full fine-tuning	-	16.24	-	4.11	-	3.86
ROI [40]	-8.67	1.14	2.43	-0.2	-1.45	0.04
TIC+SFT [45]	-62.10	11.07	-26.56	1.37	-26.47	1.4
TIC+channel selection [27]	-31.13	5.88	53.66	-3.8	54.37	-2.8
Ours (*TransTIC*)	-58.77	10.02	-46.07	2.72	-45.83	2.66

COCO2017-*val* [24] using a pre-trained Faster R-CNN [36] and Mask R-CNN [14] as the downstream recognition networks, respectively. Note that these recognition networks are the same as those utilized for learning g_p and the decoder-side prompts. We adopt top-1 accuracy as the quality metric for classification and mean average precision (mAP) for both detection and instance segmentation.

Baseline Methods. The baseline methods include: (1) using the base codec (g_a, g_s, h_a, h_s) trained for human perception without prompting, known as *TIC*, (2) fine-tuning *TIC* end-to-end for the downstream machine tasks, i.e., *full fine-tuning*, (3) transferring *TIC* by introducing spatial feature transform (SFT) layers, termed *TIC+SFT*, (4) selecting partial channels of the image latent y produced by *TIC* to perform coding for machine perception, termed *TIC+channel selection*, and (5) adopting region-of-interest (ROI) coding proposed in [40], which modulates a CNN-based compression backbone with SFT layers [45] according to a ROI map.

For *TIC+SFT*, we introduce a SFT layer after every Swin-Transformer block in the encoder and decoder. In particular, the affine parameters for SFT are produced by a task-specific network that takes the coding image as input. In a sense, *TIC+SFT* presents an alternative to our prompting technique. For *TIC+channel selection*, we follow [27] in implementing a task-specific channel selection module, which observes the image latent for channel selection, and a task-specific transform module, which converts the selected latent channels into feature maps suitable for the downstream recognition network. Both *TIC+SFT* and *TIC+channel selection* use the same base codec (i.e., *TIC*) and training protocol as our *TransTIC* to train additional task-specific networks for spatial feature transform or channel selection. More details about their implementation are provided in the supplementary document.

For ROI, we follow [40] to extract the Grad-CAM [39] output based on the pre-trained ResNet50 [15] as the ROI map for classification. For object detection and instance segmentation, we generate binary ROI maps according to the foreground predictions of Faster R-CNN [36] and Mask R-CNN [14], respectively.

4.1. Rate-Accuracy Comparison

Fig. 3 visualizes the rate-accuracy plots for the competing methods. Table 1 summarizes the average bit-rate savings and accuracy improvements using the Bjontegaard metrics [3]. BD-accuracy/mAP are computed similarly to BD-PSNR with PSNR replaced with the top-1 accuracy or mAP. Negative BD-rate numbers suggest rate saving at the same quality/accuracy level, while positive BD-PSNR/mAP/accuracy numbers suggest quality/accuracy improvements at the same bit rate. From Fig. 3 and Table 1, we make the following observations. First, (1) our *TransTIC* and *TIC+SFT* outperform *TIC+channel selection* across all the recognition tasks. This is attributed to the fact that both *TransTIC* and *TIC+SFT* are able to

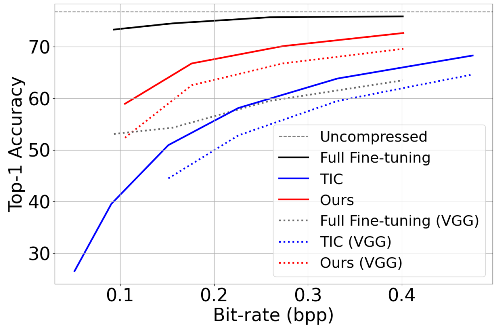
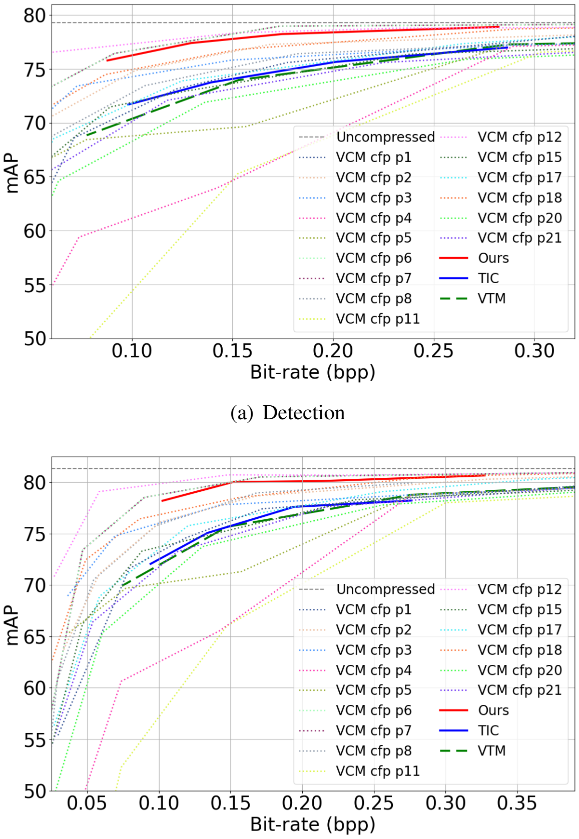


Figure 4. Rate-accuracy comparison on classification with two different recognition networks (ResNet50 and VGG19).

achieve spatially adaptive coding (see Section 4.2 for their decoded images). Second, (2) our *TransTIC* performs comparably to *TIC+SFT* on the classification task and outperforms *TIC+SFT* on more complicated tasks, such as object detection and instance segmentation. This suggests that our prompting technique works more effectively than spatial feature transform [45] in terms of transferring our transform-based codec. Third, (3) ROI [40] performs the worst in between methods with spatially adaptive ability (i.e., *TransTIC*, *TIC+SFT* and ROI). It relies heavily on the quality of the ROI mask, which can later be seen in Fig. 6. Lastly, (4) *full fine-tuning* achieves the best performance as expected. However, this comes at the expense of having to fully fine-tune the codec for the downstream recognition model. With a wide variety of machine tasks and their recognition networks, customizing neural codecs, particularly hardware-based, for each task is impractical. One feasible approach would be to re-purpose existing neural codecs already in mass production for new machine tasks. Fig. 4 shows that the image codec trained with *full fine-tuning* generalizes poorly to other unseen recognition models. Taking the classification task as an example, we replace the downstream classification model ResNet50, which is used for training under the settings of *full fine-tuning* and our *TransTIC*, with VGG19 at test time. As shown, the accuracy of *full fine-tuning* drops severely by 15%-20%. In contrast, our *TransTIC* has a less sharp decline of 3%-7%. Part of their accuracy loss comes from the smaller model capacity of VGG19 than that of ResNet50. This is evidenced by the 4%-8% accuracy loss on *TIC* when VGG19 is used in place of ResNet50.

To further validate the performance of our proposed method, we also compare our TransTIC with the methods recently submitted to the call-for-proposals (CFP) competition of the MPEG VCM standard based on their test protocol[1]. The results of these competing methods are from the CFP test report [38]. As shown in Fig. 5, our TransTIC performs comparably to the top performers in terms of rate-accuracy performance. However, our base codec has the additional constraint that it is optimized for human perception, while the top performers (e.g. p12, p6, p7) optimize the entire codec end-to-end for machine tasks. This shows the potential of our TransTIC.



(a) Detection

(b) Segmentation

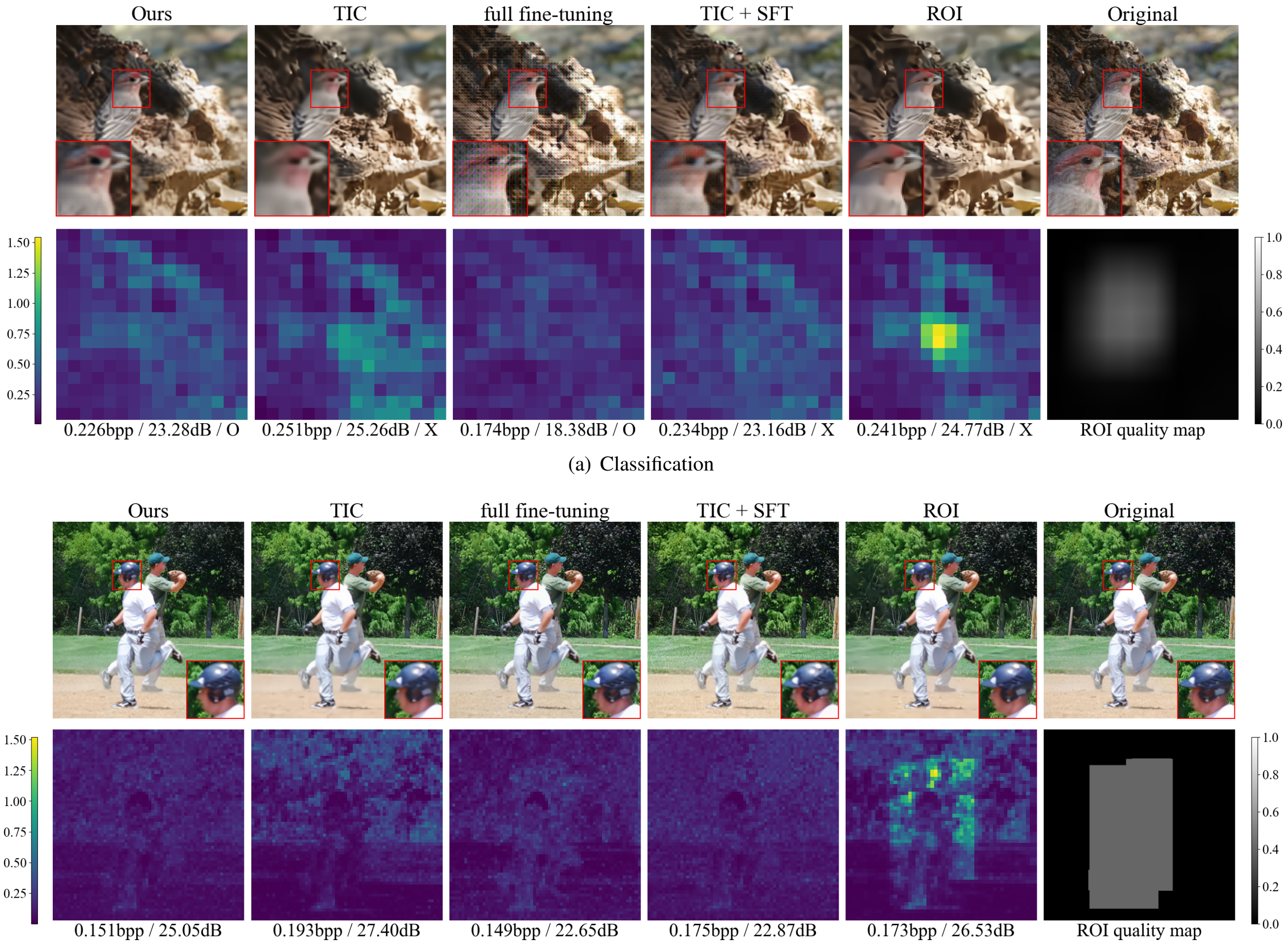
Figure 5. Rate-mAP comparison against MPEG VCM proposals on OpenImages v6 dataset [20].

4.2. Qualitative Results

Fig. 6 presents the decoded images and the corresponding bit allocation maps produced by the competing methods. As shown, the base codec *TIC*, which is optimized for human perception, tends to spend more bits on coding complex regions (e.g. the rocky surface in Fig. 6 (a) and the background forest in Fig. 6 (b)), which may be less relevant to the downstream recognition tasks. In contrast, our TransTIC and the other methods optimized for machine tasks shift more bits from the background to the foreground, resulting in generally more sharper foreground objects.

4.3. Complexity Comparison

Table 2 compares the complexity of the competing methods in terms of the kilo-multiply-accumulate-operations per pixel (kMACs/pixel) and model size. Through transferring the pre-trained *TIC*, our *TransTIC* only needs to learn for each task a prompt generation network on the encoder side



(a) Classification

(b) Object Detection

Figure 6. Visualization of decoded images (first row) and bit allocation map of latent $\hat{y}$ (second row). The rightmost image of the second row shows the quality map used for ROI method. The text below each map denotes the corresponding rate / PSNR / prediction result (classification only). The values of bit allocation maps denote the average negative log likelihood of each element in $\hat{y}$ across all channels.

and several task-specific prompts on the decoder side. The number of these additional parameters is about one fifth of that of a complete *TIC* (7M). As compared with *TIC*, the increase in kMACs/pixel (i.e. from 188 to 202) on the decoder side is modest. In contrast, our *TransTIC* has a much more complex encoder than *TIC* because of incorporating a prompt generation network, which must adapt the encoder-side prompts to the downstream task and the current input image. Nevertheless, *TransTIC* offers significantly lower kMACs/pixel and parameters than ROI [40] and *TIC+SFT* [45], both of which utilize more computationally heavy networks for SFT layers. Furthermore, ROI [40] currently uses a pre-trained recognition network to generate the ROI mask. This results in much higher kMACs/pixel on the encoder side. Table 2 provides the kMACs/pixel needed to generate the ROI masks for different tasks. Last but not least, *TIC+channel selection* [27] has a low-complexity decoder (which is composed of only the hyperprior decoder and the transform module) because it need not reconstruct the input image. However, it performs much worse than our *TransTIC* (see Fig. 3).

Table 2. Comparison of the kMACs/pixel and model size.

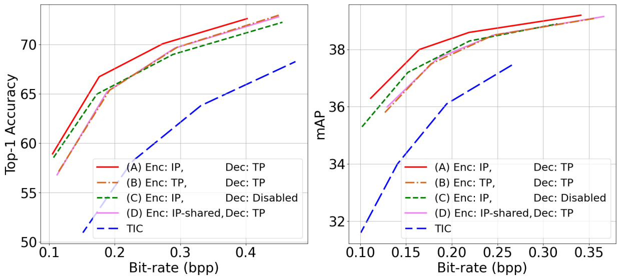
		kMACs/pixel		**Params (M)**	
		Encoder	Decoder	Encoder	Decoder
TIC		142.54	188.52	3.65	3.86
ROI [40]	Autoencoder only	800.36	679.80	21.91	5.65
	Classification	882.46	679.80	47.47	5.65
	Detection	991.13	679.80	63.39	5.65
	Segmentation	≈997.85	679.80	66.03	5.65
TIC+SFT [45]		686.39	462.23	12.08	9.62
TIC+channel selection [27]		142.54	25.13	3.76	1.96
Ours (*TransTIC*)		332.03	202.60	5.24	3.89

4.4. Ablation Experiments

IP-type vs. TP-type STBs. This ablation experiment investigates how the prompting type, instance-specific (IP-

Table 3. Ablation variants of IP-type vs. TP-type STBs.

Variants	Encoder Prompting	Decoder Prompting
A	IP-type	TP-type
B	TP-type	TP-type
C	IP-type	Disabled
D	IP-shared-type	TP-type



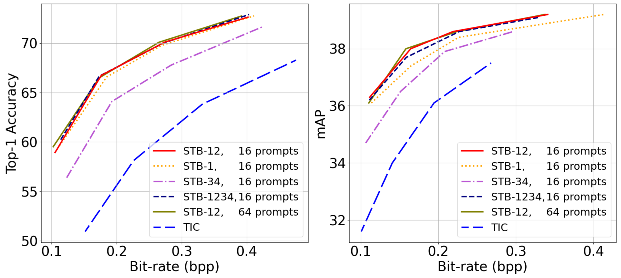
(a) Classification (b) Object Detection

Figure 7. Ablation on IP-type and TP-type STBs.

type) or task-specific (TP-type), in the encoder and decoder STBs may impact the rate-accuracy performance. We explore four variants of the proposed method, as summarized in Table 3, where the IP-shared-type refers to instance-specific prompting with the same instance-specific prompts shared across windows in a Swin-Transformer layer. When included in the encoder STBs, these shared prompts are learned by introducing a spatially adaptive pooling layer after the prompt generation network. This ensures that the number of prompts is invariant to the image size.

In Fig. 7, we make the following observations. First, a comparison of our full model (variant A) and variant B shows that IP-type prompting works more effectively than TP-type prompting on the encoder side. Intuitively, adapting the encoding process according to the input image helps generate a bitstream better suited for the downstream recognition task in the rate-accuracy sense. Second, compared to the full model, disabling the decoder-side prompting (variants A vs. C) results in only a moderate accuracy loss, suggesting that the encoder-side prompting is more critical. Third, on the encoder side, IP-shared prompting (variant D) performs worse than IP-type prompting (variant A). That is, spatially adaptive prompting is indispensable. Last but not least, all the variants outshine *TIC* significantly, implying that prompting is effective in transferring *TIC* from human perception to machine perception.

Prompt Depth. Fig. 8 analyzes which and how many STBs to inject prompts on the encoder side. As shown, injecting prompts to early STBs closer to the input image (e.g. STB-1 with 16 prompts, and STB-1,2 with 16 prompts) is more effective than injecting them to later STBs (e.g. STB-3,4 with 16 prompts). Comparing the two early prompting variants (STB-1 with 16 prompts vs. STB-1,2 with 16 prompts), it is clear that the performance gap between them is rather limited on the easier classification task, but becomes more significant on object detection. Also, the full injection variant (STB-1,2,3,4 with 16 prompts) performs comparably to the early prompting variant (STB-1,2 with 16 prompts). We thus choose to inject prompts to STB-1,2 only.



(a) Classification (b) Object Detection

Figure 8. Ablation on the depth and number of prompts.

Prompt Numbers. Fig. 8 also ablates different design choices on prompt numbers. In our design, the number of prompts in a window is chosen empirically to be 16, which is one quarter of that (i.e. 64) of the image tokens. As shown in Fig. 8, increasing the number of prompts from 16 to 64 (cp. STB-1,2 with 64 prompts vs. STB-1,2 with 16 prompts) brings little benefit on recognition accuracy. We thus stick to 16 prompts to save storage and computational cost.

5. Conclusion

This paper utilizes prompting techniques to transfer a well-trained Transformer-based image codec from human perception to machine perception. Instead of retraining the codec, we introduce additional instance-specific prompts to the Swin-Transformer layers in the encoder and task-specific prompts to the decoder. Experimental results show that our TransTIC achieves comparable or better rate-accuracy performance than the other transferring methods on various machine tasks.

Acknowledgement

This work is supported by National Science and Technology Council, Taiwan under Grants NSTC 111-2634-F-A49-010- and MOST 110-2221-E-A49- 065-MY3, MediaTek, and National Center for High-performance Computing.

References

[1] Common test conditions and evaluation methodology for video coding for machines. ISO/IEC JTC 1/SC 29/WG 2 138th meeting, N192, April 2022.

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior, 2018.

[3] Gisle Bjøntegaard. Calculation of average psnr differences between rd-curves. In *Technical Report VCEG-M33, ITU-T SG16/Q6*, 2001.

[4] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

[5] Lahiru D Chamain, Fabien Racapé, Jean Bégaint, Akshay Pushparaja, and Simon Feltman. End-to-end optimized image compression for machines, a study. In *2021 Data Compression Conference (DCC)*, pages 163–172. IEEE, 2021.

[6] Lahiru D. Chamain, Fabien Racapé, Jean Bégaint, Akshay Pushparaja, and Simon Feltman. End-to-end optimized image compression for multiple machine tasks. *CoRR*, abs/2103.04178, 2021.

[7] Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30:3179–3191, 2021.

[8] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules, 2020.

[9] Hyomin Choi and Ivan V. Bajic. Scalable image coding for humans and machines. *IEEE Transactions on Image Processing*, 31:2739–2754, 2022.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Ruoyu Feng, Xin Jin, Zongyu Guo, Runsen Feng, Yixin Gao, Tianyu He, Zhizheng Zhang, Simeng Sun, and Zhibo Chen. Image coding for machines with omnipotent feature learning, 2022.

[12] Zongyu Guo, Zhizheng Zhang, Runsen Feng, and Zhibo Chen. Causal contextual prediction for learned image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(4):2329–2341, 2021.

[13] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv preprint arXiv:2204.07143*, 2022.

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

[15] 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.

[16] Yueyu Hu, Wenhan Yang, and Jiaying Liu. Coarse-to-fine hyper-prior modeling for learned image compression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11013–11020, Apr. 2020.

[17] 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.

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

[19] A Burakhan Koyuncu, Han Gao, Atanas Boev, Georgii Gaikov, Elena Alshina, and Eckehard Steinbach. Contextformer: A transformer with spatio-channel attention for context modeling in learned image compression. In *European Conference on Computer Vision (ECCV)*, pages 447–463. Springer, 2022.

[20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[21] Nam Le, Honglei Zhang, Francesco Cricri, Ramin Ghaznavi-Youvalari, and Esa Rahtu. Image coding for machines: an end-to-end learned approach. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1590–1594, 2021.

[22] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[23] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190, 2021.

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[25] Jiaheng Liu, Guo Lu, Zhihao Hu, and Dong Xu. A unified end-to-end framework for efficient deep image compression. *arXiv preprint arXiv:2002.03370*, 2020.

[26] Jinming Liu, Heming Sun, and Jiro Katto. Learning in compressed domain for faster machine vision tasks. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 01–05, 2021.

[27] 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.

[28] Kang Liu, Dong Liu, Li Li, Ning Yan, and Houqiang Li. Semantics-to-signal scalable image compression with learned revertible representations. *International Journal of Computer Vision*, 129:1–17, 09 2021.

[29] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586, 2021.

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In

Proceedings of the IEEE/CVF international conference on computer vision, pages 10012–10022, 2021.

[31] Ming Lu, Fangdong Chen, Shiliang Pu, and Zhan Ma. High-efficiency lossy image coding through adaptive neighborhood information aggregation, 2022.

[32] Ming Lu, Peiyao Guo, Huiqing Shi, Chuntong Cao, and Zhan Ma. Transformer-based image compression. In *Data Compression Conference*, 2022.

[33] Yixin Mei, Fan Li, Li Li, and Zhu Li. Learn a compression for objection detection - vae with a bridge. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5, 2021.

[34] Fabian Mentzer, George Toderici, David Minnen, Sung-Jin Hwang, Sergi Caelles, Mario Lucic, and Eirikur Agustsson. Vct: A video compression transformer. In *Advances in Neural Information Processing Systems*, 2022.

[35] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. *CoRR*, abs/1809.02736, 2018.

[36] 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.

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

[38] C. Rosewarne. [VCM Track 2] CfP test report. ISO/IEC JTC 1/SC 29/WG 2 140th meeting, m61010, online, October 2022.

[39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[40] 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.

[41] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[42] Xining Wang, Ming Lu, and Zhan Ma. Block-level rate control for learnt image coding. In *2022 Picture Coding Symposium*, pages 157–161. IEEE, 2022.

[43] Jinxi Xiang, Kuan Tian, and Jun Zhang. Mimt: Masked image modeling transformer for video compression. In *International Conference on Learning Representations*, 2023.

[44] 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.

[45] 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.

[46] Ning Yan, Changsheng Gao, Dong Liu, Houqiang Li, Li Li, and Feng Wu. Sssic: Semantics-to-signal scalable image coding with learned structural representations. *IEEE Transactions on Image Processing*, 30:8939–8954, 2021.

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

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 [15], Faster R-CNN [36] and Mask R-CNN [14] 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:

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

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:

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

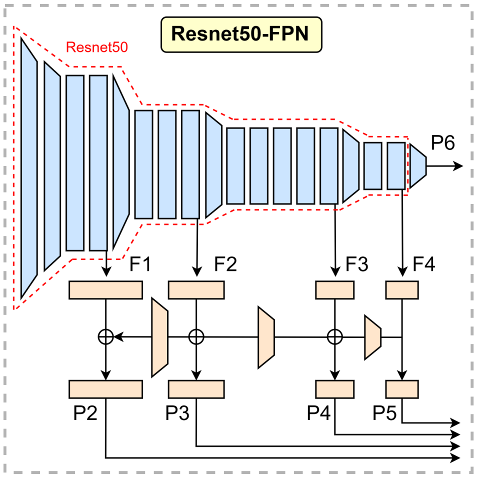


Figure A1. Architecture of Resnet50-based FPN, which shows the features selected for evaluating the perceptual loss.

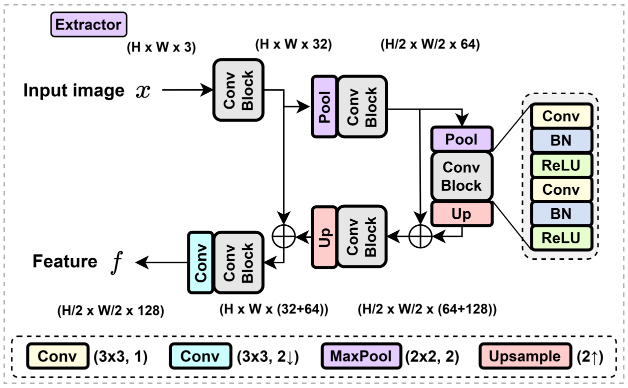


Figure A2. Architecture of the extractor in our prompt generator g_p.

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

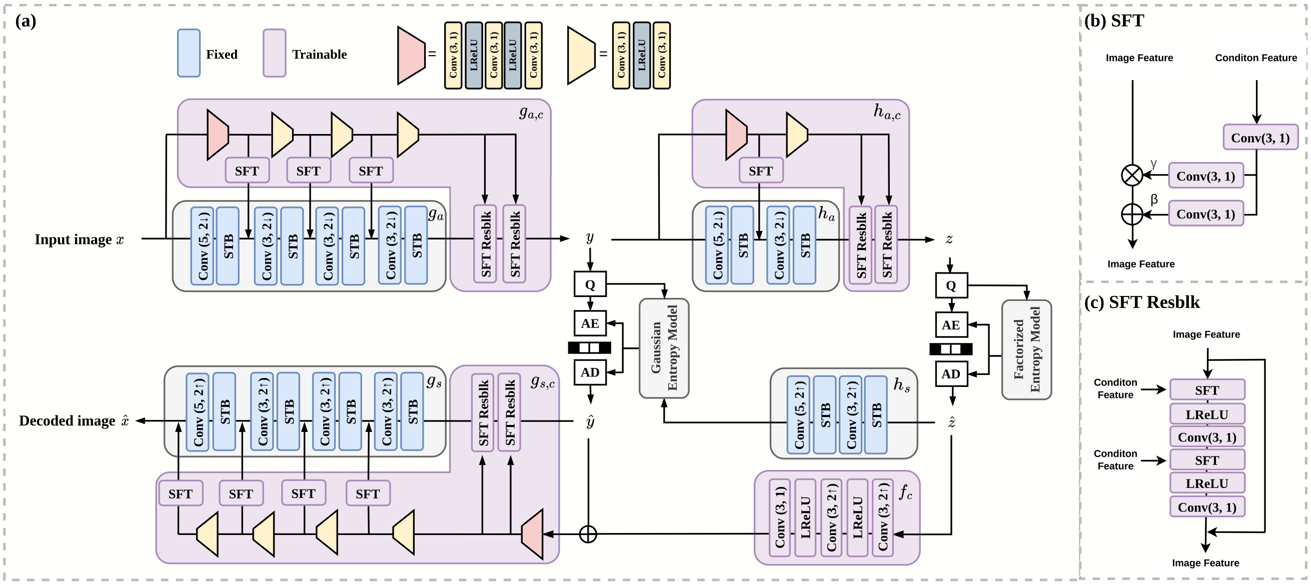


Figure A3. Architecture of *TIC+SFT*.

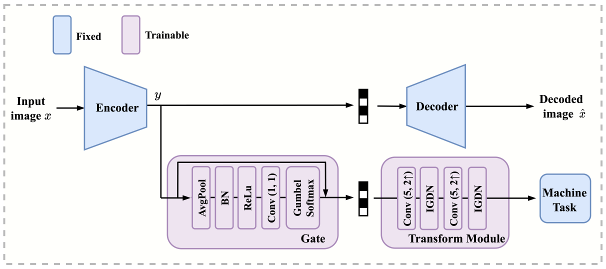


Figure A4. Architecture of *TIC+channel selection*.

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 [37]-like structure.

A1.3. *TIC+SFT*

Fig. A3 depicts the network architecture of the baseline method *TIC+SFT* [45], 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 [40] in using convolutional neural networks to produce the element-wise affine parameters γ, β for each SFT layer.

A1.4. *TIC+channel selection*

Fig. A4 shows the architecture of *TIC+channel selection* [27]. 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.

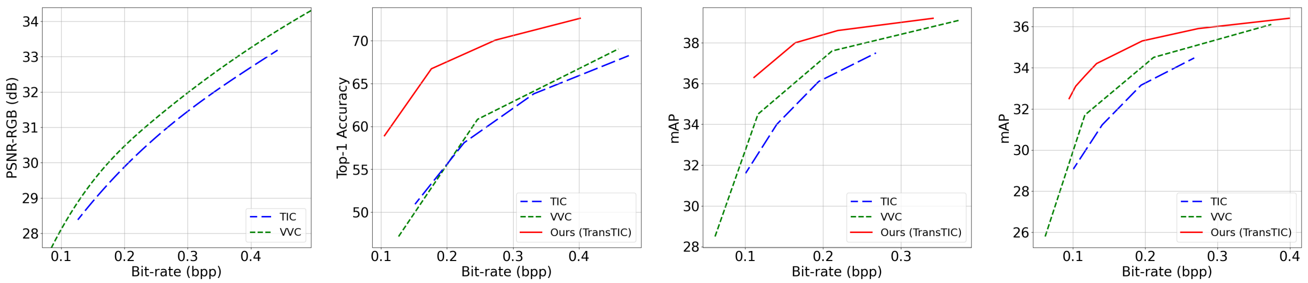
A2. Comparison with VVC

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 [18]. 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 [17]. 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 [17]. The architectural difference between *Deep* and *Shallow* is shown in Fig. A6. From Fig. A7, *Deep* performs comparably to *Shallow* on the clas-



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

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

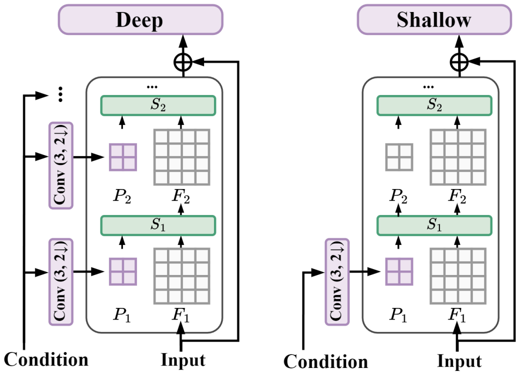
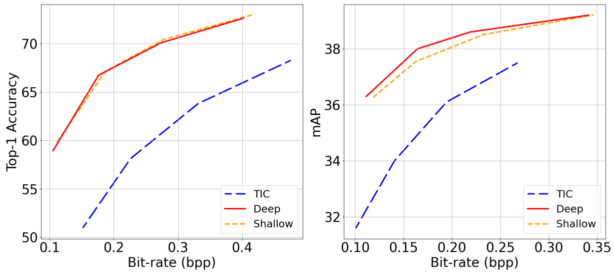


Figure A6. Architecture comparison of *Deep* and *Shallow* IP-type STB.



(a) Classification (b) Object Detection

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

sification 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 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.

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

Section	Method	kMACs/pixel		Params (M)	
		Encoder	Decoder	Encoder	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

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.

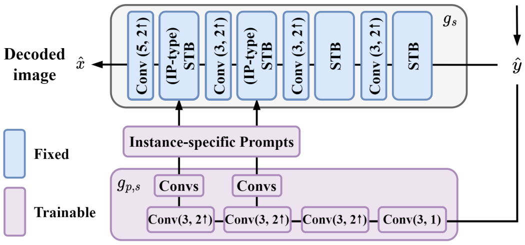
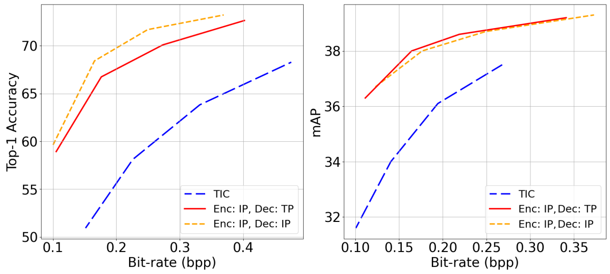
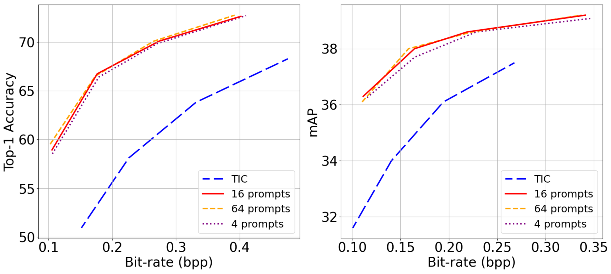


Figure A8. Architecture of IP-type STBs in the decoder.



(a) Classification (b) Object Detection

Figure A9. Ablation on decoder side STB.



(a) Classification (b) Object Detection

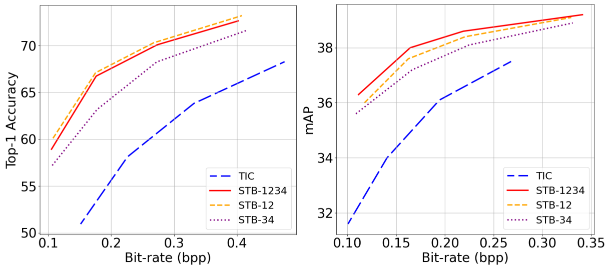
Figure A10. Ablation on the number of prompts.

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-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.

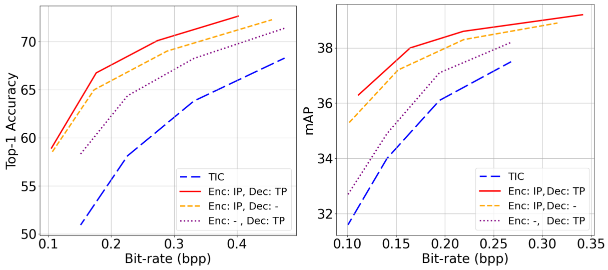
A3.5. Prompting Encoder vs. Decoder

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



(a) Classification (b) Object Detection

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

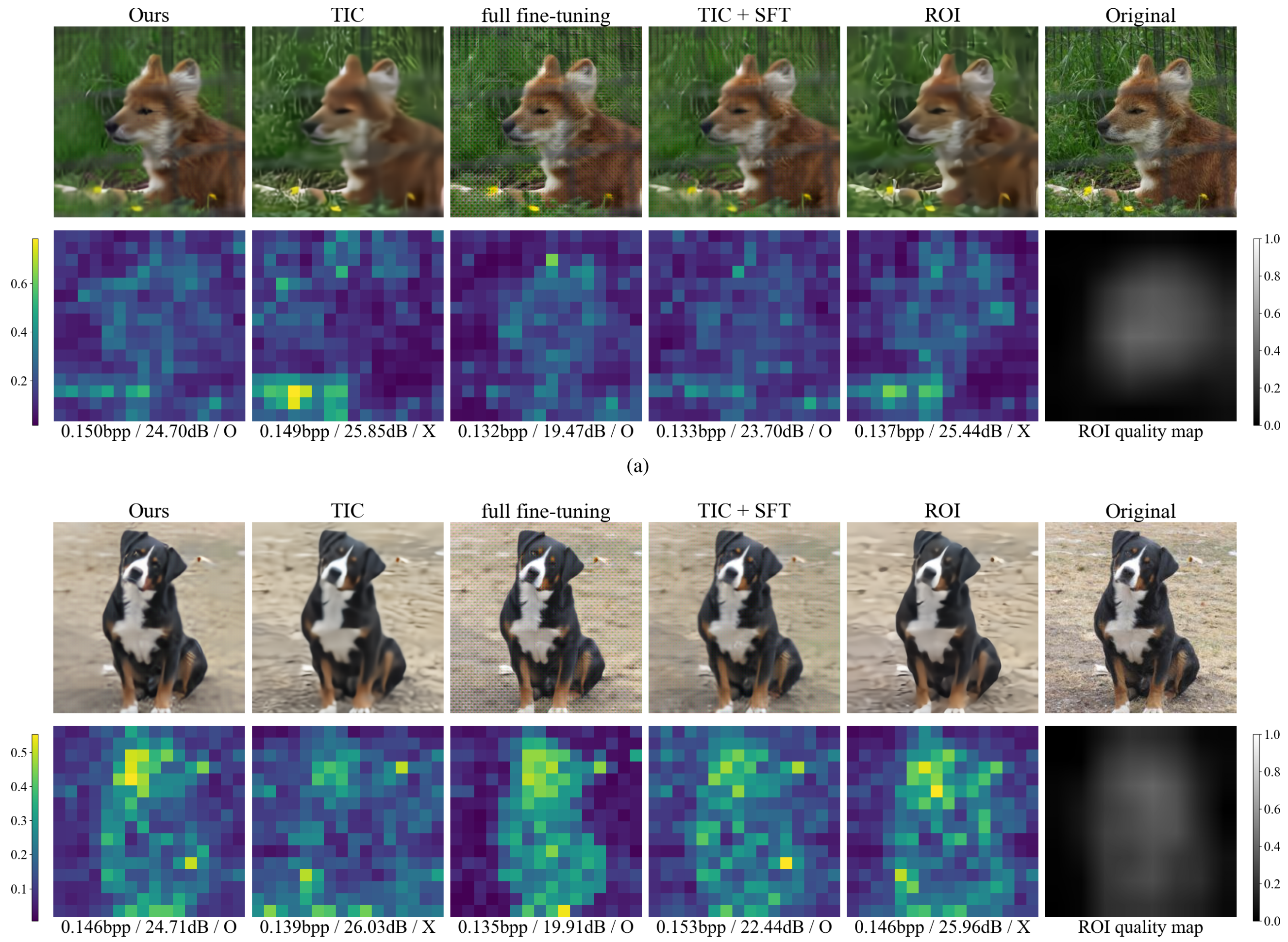


(a) Classification (b) Object Detection

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

A4. More Qualitative Results

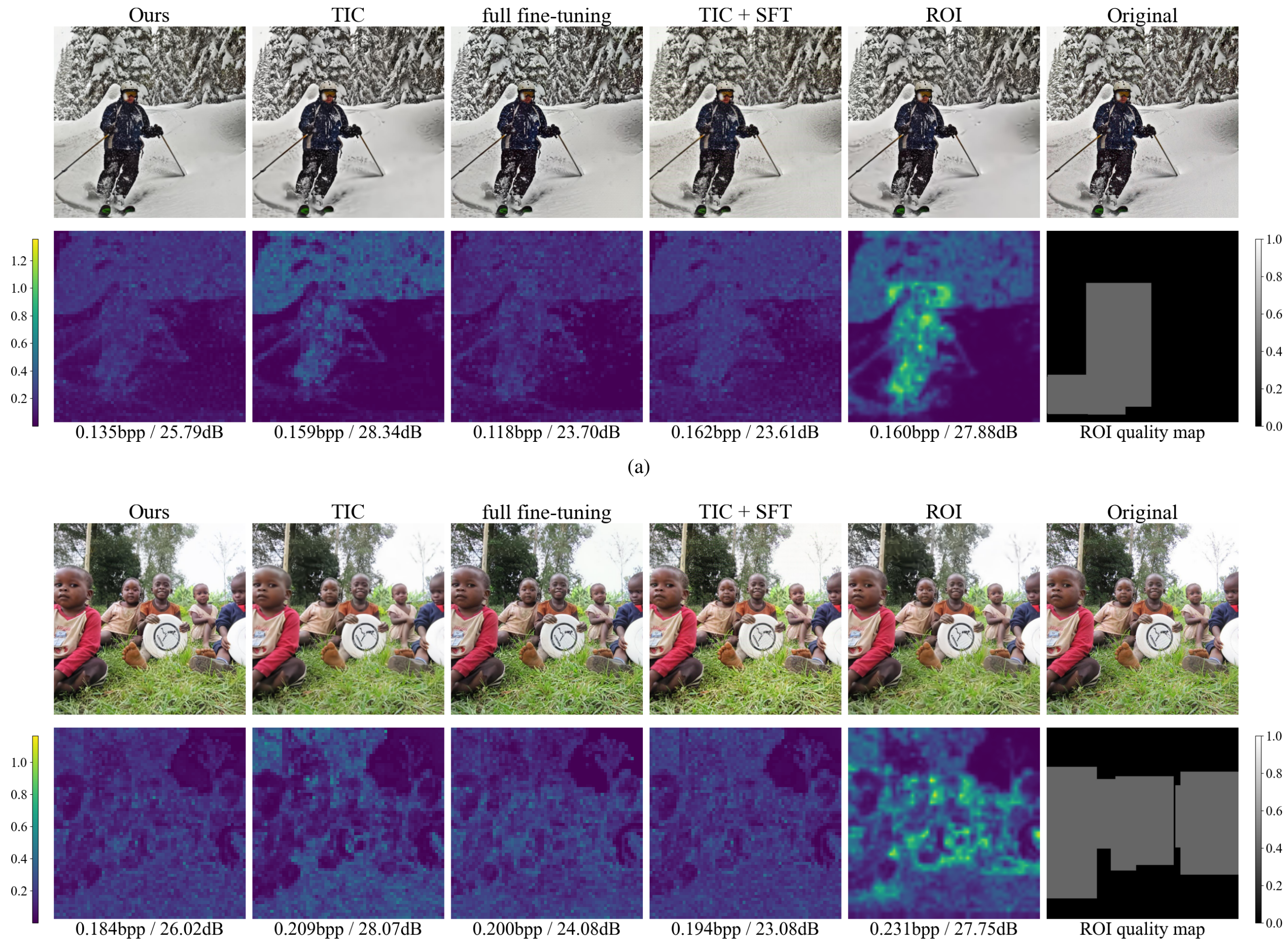
Fig. A13, Fig. A14, and Fig. A15 provide more qualitative results, comparing the decoded images and the bit allocation maps produced by the competing methods. As shown, *TIC*, the codec optimized for human perception, tends to allocate more bits to complex regions, even if those regions are less relevant (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.



(a)

(b)

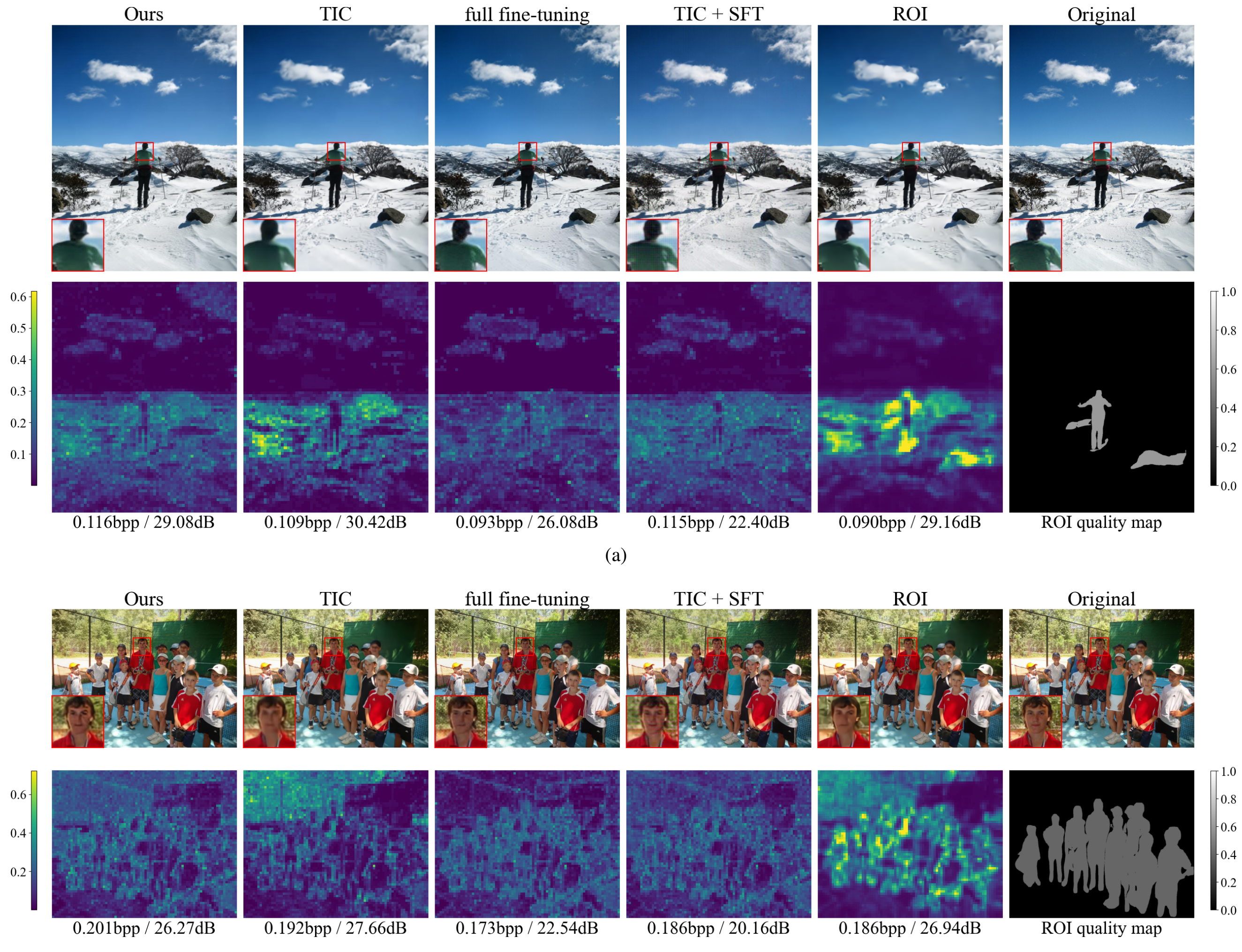
Figure A13. Visualization of the decoded images (the first row) and the bit allocation maps (the second row) of the image latent $\hat{y}$ for the classification task. The rightmost image of the second row shows the quality map used for the ROI method. The text below each map denotes the corresponding 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.