






Region-Adaptive Transform with Segmentation Prior for Image Compression

Yuxi Liu^{1,2}, Wenhan Yang³, Huihui Bai^{1,2}, Yunchao Wei^{1,2,3}, and Yao Zhao^{*1,2,3}

¹ Institute of Information Science, Beijing Jiaotong University, Beijing, China
² Visual Intelligence + X International Joint Laboratory of the Ministry of Education
³ PengCheng Laboratory, Shenzhen, China
{yuxiliu, hhbai, yunchao.wei, yzhao}@bjtu.edu.cn, yangwh@pcl.ac.cn

Abstract. Learned Image Compression (LIC) has shown remarkable progress in recent years. Existing works commonly employ CNN-based or Transformer-based modules as transform methods for compression. However, there is no prior research on neural transform that focuses on specific regions. In response, we introduce the class-agnostic segmentation masks (*i.e.* semantic masks without category labels) for extracting region-adaptive contextual information. Our proposed module, Region-Adaptive Transform, applies adaptive convolutions on different regions guided by the masks. Additionally, we introduce a plug-and-play module named Scale Affine Layer to incorporate rich contexts from various regions. While there have been prior image compression efforts that involve segmentation masks as additional intermediate inputs, our approach differs significantly from them. Our advantages lie in that, to avoid extra bitrate overhead, we treat these masks as privilege information, which is accessible during the model training stage but not required during the inference phase. To the best of our knowledge, we are the first to employ class-agnostic masks as privilege information and achieve superior performance in pixel-fidelity metrics, such as Peak Signal to Noise Ratio (PSNR). The experimental results demonstrate our improvement compared to previously well-performing methods, with about 8.2% bitrate saving compared to VTM-17.0. The source code is available at <https://github.com/GityuxiLiu/SegPIC-for-Image-Compression>.

Keywords: Learned Image Compression · Segmentation Prior

1 Introduction

With the advent of the multimedia era, an ever-growing volume of image data has emerged on the internet. These vast volumes of image data demand larger transmission bandwidth and storage capacity. In this context, the importance of image compression technology has become even more pronounced, evoking greater efforts in efficient image compression techniques. Lossy image compression

* Corresponding author

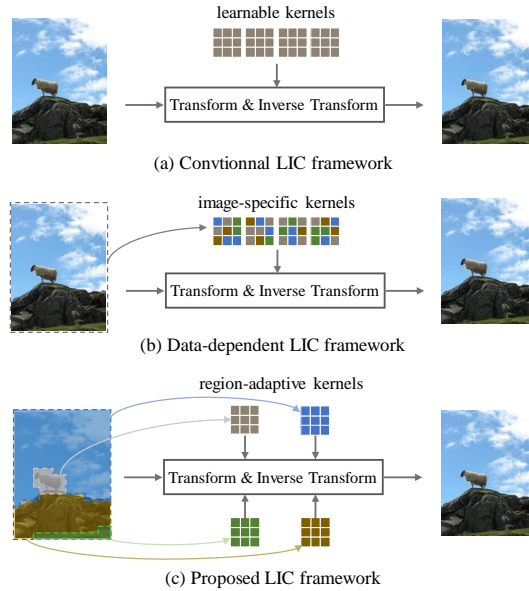


Fig. 1: Paradigms of end-to-end learned image compression (LIC) methods. (a) Conventional LIC with fixed learned kernels. (b) Data-dependent LIC with image-specific kernels [43]. (c) The proposed method is capable of generating region-adaptive kernels from different regions, providing more fine-grained transforms.

sion technology can reduce the data size to a lower level while maintaining good visual quality. In the past decades, the classical lossy image compression methods such as JPEG [42], JPEG2000 [38], BPG [7], and VVC [24], used to have excellent performance and were widely adopted. These methods include modules of transform, quantization, and entropy coding. The transform process aims to convert images into a more compact coefficient by removing pixel correlations as much as possible. The commonly employed transforms include the discrete cosine transform (DCT) [3] and wavelet transform [14]. These transformations are linear, characterized by simplicity and efficiency. However, their fixed forms are unable to adapt to the contexts of complex signals. To address this issue, latter efforts propose data-driven transformation approaches, *e.g.* the Karhunen-Loève transform (KLT) [39]. Furthermore, the subsequent works are dedicated to constructing complex patterns for the adaptive selection of local transforms, improving adaptability and overall performance.

Benefit from the advancement of neural network technology, many learned image compression (LIC) methods have been proposed, which shows a promising alternative for designing transform: learning from data instead of handcraft design. Balle *et al.* [5, 6] proposed the first well-known end-to-end framework. They utilize variational auto-encoders (VAE) [26] as a transform module and optimize the parameters of the kernels to better align with the natural data distribution. To further improve the representation capabilities of the transform

module, some works [12, 30, 43, 48, 49] design advanced modules to introduce richer contextual information, *e.g.* residual blocks [20] and self-attention mechanism [15, 32]. These works optimize learnable kernels to fit the data distribution to attain a more powerful transform. However, their kernel parameters are fixed after training (see Fig. 1 (a)). Wang *et al.* [43] introduce a neural data-dependent transform, which extracts dynamic convolution kernels based on the context of the entire image for more adaptive transform (see Fig. 1 (b)). However, even within a single image, the context could vary significantly across different regions, *e.g.* stone and sheep regions in Fig. 1. It limits the description of the signals to a more fine-grained level using such image-level transforms.

To address the issue, this paper makes the first efforts to design learnable region-adaptive transform to capture the mapping regularity in different regions. In detail, we employ class-agnostic⁴ segmentation masks to extract region-specific information for transform (see Fig. 1 (c)). Differs from previous works [4, 11, 16, 21], our approach stands out with two characteristics in mask utilization. First, our masks are “class-agnostic”, where the semantic labels are abandoned but a more effective compression-driven semantic prior will be learned by our model in a data-driven manner. Second, to avoid incurring additional costs in both bitrate and computational resources, we treat these masks as *privilege information*, which are available during training but remain inaccessible during the inference phase [29]. Specifically, we replace the masks with uniform grid partitions (see Fig. 9 (c)) in inference, and the experimental result shows approximate performance. It implies that our model has learned useful semantic knowledge from the masks in training, incorporating this knowledge into the model (*i.e.* the network’s parameters). Hence, it remains capable of mining contextual information even from relatively coarse partition regions, *i.e.*, the given uniform grid partitions. This simple operation can effectively maintain the consistency of our model’s architecture and parameters, whether or not masks are used at different phases. The more details about the difference between our method and previous semantic-mask-based works will be explained at Sec. 2.

Based on the above motivations, we propose a novel image compression framework called **Segmentation-Prior-Guided Image Compression (SegPIC)**, equipped with two key modules: Region-Adaptive Transform (RAT) and Scale Affine Layer (SAL). The RAT adopts region-adaptive kernels for transform, performing the convolution whose weights are generated from the prototypes and the context of latent features under the guidance of masks. The SAL affines the latent features for better mining of semantic contexts in the Encoder and Decoder. Experimental results show that SegPIC is superior to previously well-performing image compression methods. Our main contributions are as follows:

- We propose a novel image compression framework called Segmentation-Prior-Guided Image Compression (SegPIC) to generate more effective region-adaptive transforms in a data-driven manner.

⁴ “class-agnostic” means distinguishing different objects in an image regardless of what those objects are. Such masks can be regarded as a generalized representation of partition maps widely used in conventional codecs.

- Our SegPIC is equipped with two proposed modules: Region-Adaptive Transform (RAT) and Scale Affine Layer (SAL). The RAT adopts region-adaptive kernels for transform, while the SAL affines the latent features for better mining of semantic contexts.
- We utilize the class-agnostic masks as privilege information to assist the training of SegPIC. The “class-agnostic” property promotes the model to learn a compression-driven semantic prior, while the “privilege” way avoids additional bitrate and complexity for masks in inference.
- Experimental results demonstrate that our SegPIC is superior to previously well-performing image compression methods. We also present sufficient ablation experiments to confirm the necessity of the privilege information.

2 Related Works

Learned Image Compression. Image compression can be divided into three main components: transform, quantization, and entropy coding. In brief, the transformation module maps the image to a latent feature space. Subsequently, quantization is performed after this phase, resulting in some quantization distortions but achieving lower bitrate. The entropy encoding module estimates the data distribution of the latent features and inputs it to the Arithmetic Coding (AC) module, which generates the bitstream for transmission.

To enhance transform module, some works [30, 43, 48, 49] adopt more powerful modules like self-attention mechanism [32, 41] to explore broader context. [31] proposed a transform block that mixes CNN and Transformer to combine their advantages. As for the entropy estimation module, the studies [12, 19, 33, 34] usually focus on exploring novel auto-regressive methods to split the latent features into several portions to be decoded in order. The portions having been decoded will assist in decoding the next portion, through the spatial or channel correlation of these portions. This auto-regressive mechanism may introduce additional latency but achieve better performance than Hyper Prior [6]. Recently, [23] integrates different types of correlations present in latent representation, *e.g.* channel-wise, local spatial, and global spatial correlations. Their proposed novel auto-regressive entropy module achieve powerful performance.

Thanks to these great works, the performance of LIC has become so strong that achieving further enhancements becomes progressively challenging. Therefore, some new directions in compression have emerged, which is also meaningful on certain occasions. For example, some semantic-based compression methods [10, 11] adopt GAN [17] to compress images at extremely low bitrate. These methods maintain a good human visual perception effect but usually, poor pixel-fidelity metrics, *e.g.* Peak Signal to Noise Ratio (PSNR) and Multi-Scale Structural Similarity Index (MS-SSIM) [46]. Specifically, these methods could introduce some synthetic artifacts with low fidelity. For example, a semantic compression method could reconstruct an image of a bird with different breeds from the original image because the training category labels do not distinguish the bird’s breed. Another direction is compression for machines, which means compress-

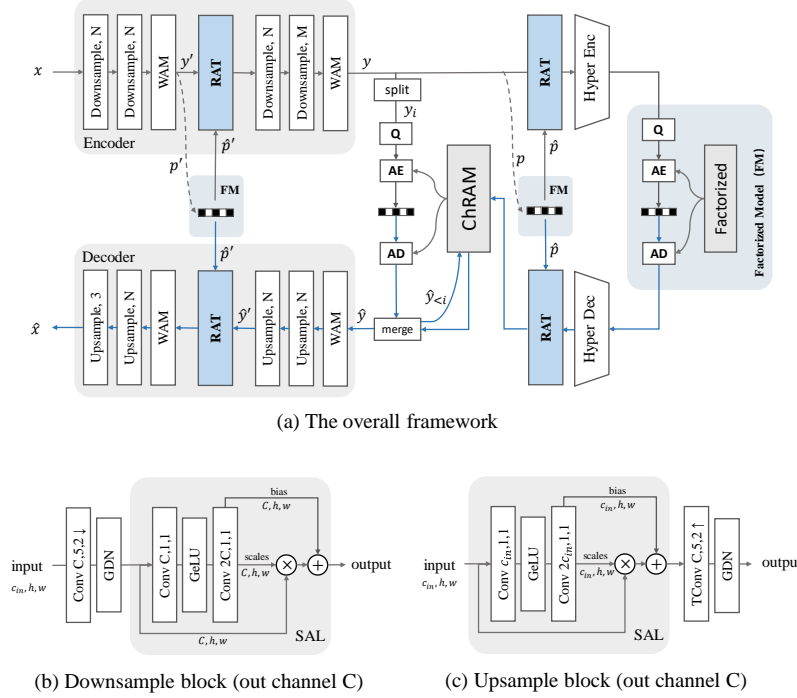


Fig. 2: (a) Overall framework of our SegPIC. The blue flow lines represent the decoding process on the decoder side. The dashed lines represent the process of extracting the prototypes p' , p (see Eq. 5). The compression of p' , p is not fully depicted for simplicity, please refer to Fig. 4 for details. N is set to 192, and M is 320 as in [49]. RAT is the proposed Region-Adaptive Transform. SAL is the proposed Scale Affine Layer. WAM is Window Attention Module [49]. ChARM is Channel-wise Auto-Regressive Model [34]. FM is a Factorized Model [6]. GDN is Generalized Divisive Normalization [5]. (b) is Downsample Block. “Downsample, N” in (a) means the block with out-channel N. “Conv C,5,2” means out-channel C, kernel size 5, stride 2. (c) is Upsample Block, and “TConv” is Transposed Convolution.

ing images for downstream tasks [16, 40]. These methods focus on balancing the bitrate and performance of downstream tasks, not just the pixel-fidelity metrics.

Although some works [10, 11, 16] about the two directions also adopt masks, our usage is significantly different. The main differences are as follows: 1) We utilize the masks as privilege information, without the mask information during the inference phase, while usually, the masks are indispensable for other methods. 2) We employ class-agnostic masks to guide the model to extract rich compression-friendly contexts among semantically related regions, regardless of the category that might degrade the fidelity as mentioned above. 3) The proposed model is optimized towards better pixel-fidelity metrics, *e.g.* PSNR and MS-SSIM.

Region-adaptive Convolution. Due to its capacity to model the degradation levels in different regions of images, region-adaptive convolution is applied to

some low-level vision tasks such as super-resolution and denoising [37, 47]. In terms of compression, the reconstruction quality varies across different regions. For instance, regions with complex textures are more challenging compared to those with simpler patterns. Therefore, specific processing can be applied to different regions based on their characteristics. The regions can be divided using segmentation masks or partitions based on certain regular patterns. To the best of our knowledge, we are the first to introduce this mechanism on image compression for optimizing pixel-fidelity metrics.

3 Proposed SegPIC

3.1 Motivation

As mentioned above, previous transforms either adopt convolutions with fixed parameters or image-specific convolutions. Our motivation is to construct a region-adaptive transform via the following three properties:

- **Region-dependent:** we propose a neural network with region-adaptive transforms guided by the segmentation masks.
- **Class-Agnostic:** the adopted mask does not include a category label, allowing the model to be free to learn compression-friendly semantic priors.
- **Privilege Information:** the masks are expected as privilege information, not being used in inference for saving bitrate and computation resources.

Based on these considerations, we propose our SegPIC in the VAE framework incorporated with two proposed modules: SAL and RAT.

- **SAL:** First of all, we extract a prototype from each region, which will be compressed and transmitted to be accessible on both the encoder and decoder sides. The prototypes are expected to extract high-level information from regions to ensure the representative ability as much as possible. However, the encoder in compression frameworks often extracts features that contain low-level information. Therefore, a module is highly desirable for communicating information across different levels. SAL is designed for extracting more abundant semantics and contexts, enriching the prototypes, and connecting features from two different levels.
- **RAT:** performing the Depth&Point-wise Separable Convolution (DPSCConv), a novel convolution whose weights are generated from the prototypes and the context of latent features under the guidance of masks. Note that, we only utilize the masks to guide RAT to learn more compression-friendly semantic priors in training, while during the inference phase, we instead adopt uniform grid partitions as coarse alternatives.

3.2 Formulation

In this section, we will introduce the problem and our method in mathematical form. Given an image x , the encoder En maps it to latent features y . After

quantization Q , \hat{y} will be compressed to the bitstream by the entropy coding, such as arithmetic coding (AC). Then the receiver converts the bitstream back into \hat{y} by entropy decoding. The above process can be formulated as follows:

$$\begin{aligned} y &= En(x), & \hat{y} &= Q(y), \\ \text{Transmitter: } & bit_{\hat{y}} &= AE(\hat{y}|p_{\hat{y}}), \\ \text{Receiver: } & \hat{y} &= AD(bit_{\hat{y}}|p_{\hat{y}}), \end{aligned} \quad (1)$$

where $p_{\hat{y}}$ is the estimated distribution of \hat{y} , $bit_{\hat{y}}$ is the binary bitstream of \hat{y} , and AE and AD are arithmetic encoding and decoding. We use a single Gaussian distribution to estimate $\hat{y} \sim \mathcal{N}(\mu, \sigma^2)$, and the mean and variance value μ, σ^2 will be yielded through an auto-regressive mechanism. The formulation is:

$$z = En_z(y), \hat{z} = Q(z), \quad \mu_i, \sigma_i = F(De_z(\hat{z}), \hat{y}_{<i}), \quad (2)$$

where En_z, De_z is hyper encoder and decoder, μ_i, σ_i are the mean and variance of $\hat{y}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, \hat{y}_i is the currently decoded portion of \hat{y} , $\hat{y}_{<i}$ is the already decoded portion to assist decoding \hat{y}_i , and F is the auto-regressive entropy model. The AC process of z in Eq. 2 is omitted for brevity. Finally, the receiver adopts decoder De to reconstruct the image \hat{x} , which is formulated as $\hat{x} = De(\hat{y})$.

The loss function \mathcal{L} to optimize this problem is:

$$\mathcal{L} = R + \lambda \cdot D = R(\hat{y}) + R(\hat{z}) + \lambda \cdot \mathbb{E}_{x \sim p_x} [d(x, \hat{x})], \quad (3)$$

where $R(\hat{y}), R(\hat{z})$ are the bitrate of \hat{y}, \hat{z} , $d(x, \hat{x})$ is the distortion between x and \hat{x} , λ denotes different rate-distortion trade-offs. Here we use Mean Square Error (MSE) or (1-MS-SSIM) as the distortion function.

In our implementation, we use class-agnostic semantic masks (denoted as m) as additional inputs of the encoding En during the training phase. Moreover, we extract a prototype from each region at the middle layer of En , then compress and transmit them (denoted as \hat{p}). In inference, we replace the masks with uniform grid partitions, so there is no additional bitstream for masks, while the bitrates of the prototypes also need to be taken into account. The En and De will transform the image with the assistance of the prototypes. The formulations requiring adjustment are as follows:

$$\begin{aligned} y &= En(x, \hat{p}|m), & \hat{x} &= De(\hat{y}, \hat{p}|m), \\ \mu_i, \sigma_i &= F(De_z(\hat{z}), \hat{y}_{<i}, \hat{p}|m), \\ R &= R(\hat{p}) + R(\hat{y}) + R(\hat{z}). \end{aligned} \quad (4)$$

3.3 Network Design

In this section, we will introduce the concrete process and our network design. We use [49] as the backbone. The overall architecture is in Fig. 2.

Scale Affine Layer. The Encoder and Decoder are responsible for image transform and reconstruction, each comprising 4 downsampling or upsampling blocks.

The previous sampling blocks consisted of a single sample convolution layer, aimed at modifying the data size. However, this may not be sufficient to extract adequate semantics and contexts. Inspired by Spatial Feature Transform (SFT) [45], which can generate visually pleasing textures in segmentation-guided low-level vision, we propose to add a Scale Affine Layer (SAL) between two adjacent convolution layers (see Fig. 2 (b) (c)). It consists of two 1×1 convolution layers with a GELU activation function in between, which is formulated as:

$$\begin{aligned} X_s &= \text{SAL}_s(X), & X_b &= \text{SAL}_b(X), \\ Y &= X_s \otimes X + X_b, \end{aligned}$$

where X and Y represent the input and output, X_s and X_b denote the scales and biases generated by SAL, \otimes denotes the element-wise product.

Extract and Transmit the Prototypes. The Encoder first converts the original image $x \in \mathbb{R}^{3 \times H \times W}$ into latent features $y' \in \mathbb{R}^{N \times H/4 \times W/4}$ at the middle layer. Then we conduct Masked Average Pooling (MAP) on y' through the class-agnostic masks $m \in \{0, 1\}^{n \times H \times W}$ (see Fig. 3 (a)), to acquire the prototypes $p' \in \mathbb{R}^{N \times n}$, where n is the number of the masks. Thanks to the proposed SAL, the Encoder becomes deeper, allowing p' to incorporate more semantics and contexts from the corresponding mask. The formulation of MAP is as follows:

$$p' = \text{MAP}(y', m) = \left\{ \frac{1}{\sum m^{(i)}} \sum y' \otimes m^{(i)}, \dots \right\}_{i \in [1, n]}, \quad (5)$$

where y' are the mid-level features (see Fig. 2 (a)), \sum denotes summation along the spatial dimension. p' will be compressed and transmitted to the receiver (see Fig. 4), and the reconstructed \hat{p}' is shared between the Encoder and Decoder. The channel $\{C_1, C_2, C_3, C_4\}$ is $\{192, 128, 96, 96\}$ for p' (on Encoder and Decoder) and $\{320, 192, 128, 96\}$ for p (on Entropy Model).

Region-Adaptive Transform. Here we propose a novel convolution called Depth&Point-wise Separable Convolution (DPSCConv), in which each element has an individual convolution kernel, and there is no channel-wise mutual communication for computational efficiency.

Before introducing it, we will start with standard convolution briefly. The input of standard convolution can be denoted as $X \in \mathbb{R}^{C \times H \times W}$, the output is $Y \in \mathbb{R}^{O \times H \times W}$, and the kernel is $W \in \mathbb{R}^{O \times C \times k \times k}$, where k is kernel size. The formulation of standard convolution is as follows:

$$Y_{o,h,w} = \sum_{c=1}^C X_{c,h,w} * W_{o,c} = \sum_{c=1}^C \sum_{i,j} X_{c,h+i,w+j} \cdot W_{o,c,i,j},$$

where $*$ is 2D convolution, $i, j \in [-\lfloor k/2 \rfloor, \lfloor k/2 \rfloor]$, k is kernel size, and $\lfloor \cdot \rfloor$ denotes floor rounding.

DPSCConv can be seen as a combination of Depthwise Separable Convolution (DWConv) [13] and Dynamic Convolution [22]. The formulation is as follows:

$$Y_{c,h,w} = X_{c,h,w} * W_{c,h,w} = \sum_{i,j} X_{c,h+i,w+j} \cdot W_{c,h,w,i,j}. \quad (6)$$

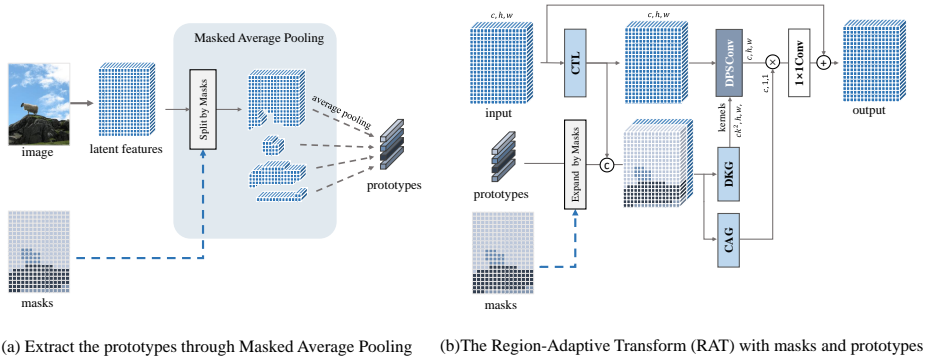


Fig. 3: The diagram of extracting the prototypes and the proposed RAT module. DPSConv is proposed Depth&Point-wise Separable Convolution (see Eq. 6). CTL is Channel Transform Layers, DKG is the DPSConv Kernel Generator, and CAG is the Channel Attention Generator. DPSConv is Depth&Point-wise Separable Convolution. The detailed modules are represented in Fig. 5.

The computational complexity of standard convolution is $O(O \times C \times H \times W \times k^2)$, while DPSConv is $O(C \times H \times W \times k^2)$, same as DWConv.

The overall design of RAT is depicted in Fig. 3 (b). We expand the prototypes based on the positional information of the masks, filling in the corresponding copies of the prototypes at each pixel position. Then, we feed-forward the input through Channel Transform Layers (CTL) and concatenate it with the expanded features. The resulting variable, which contains both semantics and contexts, is used for subsequent filter generation. The DPSConv Kernels Generator (DKG) is designed for generating the filters. In the first layer of DKG, we employ a 1x1 convolution to attempt the fusion of the prototype and the pixel’s intrinsic attribute. We opt for grouped convolution in the final layer of the DKG to reduce the computation and parameters. Then Channel Attention Generator (CAG) generates the channel attention to be multiplied with the output of DPSConv along the channel dimension. Finally, the output passes through a 1x1 convolution to merge channel information. We apply the RAT in both the Encoder and Decoder, as well as the entropy encoding module.

Entropy Encoding Module. We use the Channel-wise Auto-Regressive Model (ChARM) [34] as our entropy estimation module, and the specific hyperparameter configuration follows [49]. In this model, the latent variable y is partitioned into 10 groups along the channel dimension. The Gaussian prior distribution $p(y_i | \hat{z}, \hat{y}_{<i})$ is estimated in an auto-regressive manner, where the mean and scale of y_i depend on the quantized latent in the preceding groups $\hat{y}_{<i}$ (see Eq. 2). To further enhance the estimation ability, we also deploy RAT in it. The RAT here can transmit key region prototypes through a separate bitstream, assisting in more efficient encoding and decoding. The prototypes p could be seen as segmentation-prior, a supplement of the hyper-prior information z .

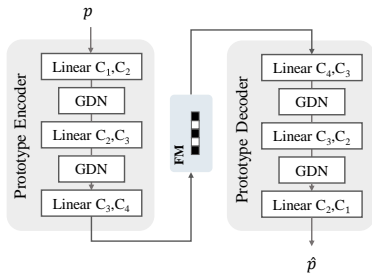


Fig. 4: The architecture of the Prototype Encoder and Decoder for compressing and transmitting the prototypes. “Linear C_1, C_2 ” means linear layer with in-channel C_1 and out-channel C_2 .

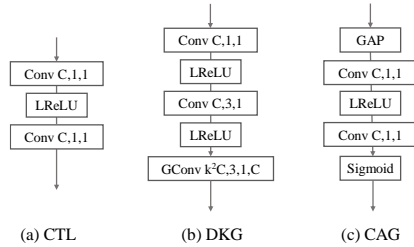


Fig. 5: The detailed modules in RAT, including Channel Transform layers (CTL), DPSCConv Kernels Generator (DKG), and Channel Attention Generator (CAG). LReLU is Leaky ReLU. GAP is Global Average Pooling. “GConv $k^2C, 3, 1, C$ ” means Grouped Convolution with out-channel k^2C , kernel size 3, stride 1 and group C .

Table 1: BD-rate(%) performance on Kodak and CLIC Professional Valid (CLIC for short). The anchor is VTM-17.0.

Model	<i>Kodak</i>		<i>CLIC</i>		
	BD-rate↓	BD-rate↓	BD-rate↓	BD-rate↓	
BPG [7]	+31.5	+48.1	EVC [44]	+0.27	–
Cheng2020 [12]	+7.46	+11.7	Feng2023 [16]	-2.21	–
Agustsson2023 [2]	+6.15	–	ELIC [18]	-3.08	-3.58
ChARM [34]	+3.58	–	Contextformer [28]	-5.38	–
Wang2022 [43]	+1.10	-0.99	Pan2022 [35]	-5.42	–
BL-S	+15.5	+16.6	Zou2022 (BL) [49]	-2.89	+0.14
Ours-S	+3.87	+5.26	Ours	-8.18	-8.26

4 Experiments

4.1 Experimental Setup

Training. We use COCO-Stuff [9] as the training dataset, which is composed of 118k images with panoptic semantic masks. We just use the class-agnostic masks without the category labels. We randomly crop the images with the size of 256×256 . The models are trained for 270 epochs (about 1M iterations) using the Adam optimizer [25] with a batch size of 32. The initial learning rate is set to 1×10^{-4} for 170 epochs, and reduced by half every 30 epochs. Our models are optimized by the rate-distortion trade-off loss function (see Eq. 3 and Eq. 4). The λ belongs to $\{0.0018, 0.0035, 0.0067, 0.0130, 0.0250, 0.0483\}$ for MSE as the distortion, and $\{2.4, 4.58, 8.73, 16.64, 31.73, 60.50\}$ for MS-SSIM [46].

Evaluation. We evaluate our model on two commonly used datasets, Kodak [27] with the size of 768×512 , and CLIC Professional Validation [1] with the size of 2K. Performance is evaluated using both bitrate and distortions. Distortion

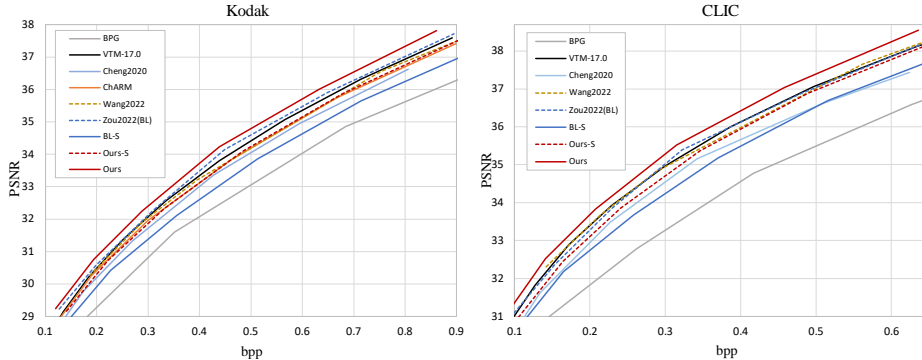


Fig. 6: Rate-Distortion Performance on Kodak [27] and CLIC Professional Valid dataset [1]. Our SegPIC introduces proposed RAT and SAL based on Zou2022 [49]. We also implement a simple baseline model without self-attention and auto-regressive mechanism (BL-S), and incorporate our proposed modules on it (Ours-S).

measurement involves PSNR and MS-SSIM. And bitrates are assessed in bits per pixel (bpp). In this phase, we replace the masks with 4×4 grid partitions. Thus, there is no bitrate cost related to the masks, while the bitrate of prototypes $R(p)$ and $R(p')$ is still taken into account.

4.2 Rate-Distortion Performance

We compare our SegPIC with previously well-performing methods, including discretized Gaussian mixture likelihoods and attention modules [12] (denoted as Cheng2020), ChARM [34] (denoted as Minnen2022), ChARM with WAM [49] (denoted as Zou2022), image-specific data-dependent transform model [43] (denoted as Wang2022), the best-performing traditional method VTM-17.0 [24] and influential traditional method BPG [7]. Furthermore, we introduce two simplified models without self-attention and auto-regressive mechanism, denoted as baseline-small (BL-S) and the one incorporating our proposed SAL and RAT (Ours-S), to demonstrate the generality of our modules. Details about the architecture of these two models will be provided in the appendix.

As illustrated in Fig. 6 and Tab. 1, compared to VTM-17.0, our SegPIC could save 8.18% BD-rate [8] on Kodak and 8.26% on CLIC, which outperforms all the LIC and traditional methods mentioned above. To further verify the efficiency of the proposed RAT and SAL, We also compare SegPIC with the baseline Zou2022, which can save 5.45% BD-rate on Kodak, and 8.50% on CLIC (see Tab. 2). Moreover, Ours-S even achieves closely matched performance compared to ChARM. The MS-SSIM results are provided in the appendix.

4.3 Ablation Study

To assess the contributions of our RAT and SAL modules individually, we independently incorporate these two modules into the baseline Zou2022. Conse-



Fig. 7: Visualization of the reconstructed images *kodim04* and *kodim24* in Kodak. The metrics are (PNSR \uparrow /bpp \downarrow). It shows that our SegPIC can distinguish the objects’ contours more accurately, making the edges sharper with less bitrate.

Table 2: Ablation Study on Kodak and CLIC. The Baseline (BL) is Zou2022 [49]. The speed is measured on Kodak with one Nvidia RTX 3090 GPU.

Model	<i>Kodak</i>		<i>CLIC</i>		Enc Time	Dec Time
	Params	BD-rate \downarrow	BD-rate \downarrow	BD-rate \downarrow		
Zou2022 (BL)	75.2 M	0.0	0.0	120 ms	124 ms	
BL w/ SAL	76.7 M	-2.12	-2.93	–	–	
BL w/ RAT	82.0 M	-2.75	-4.35	–	–	
Ours	83.5 M	-5.45	-8.50	142 ms	130 ms	
ChARM	48.8 M	+6.52	–	107 ms	108 ms	
BL-S	21.7 M	+19.0	+16.3	–	–	
Ours-S	29.2 M	+6.96	+5.11	99 ms	46 ms	

quently, we construct two models: “baseline w/ SAL” and “baseline w/ RAT.” As illustrated in Tab. 2, the parameter counts introduced by RAT and SAL separately are 1.5M and 5.8M, which occupy only a small fraction of the total parameter count, about 1.7% and 6.9%. And they can contribute to BD-rate saving by 2.93% and 4.35% on CLIC Professional Valid, respectively. It shows that the proposed RAT and SAL are lightweight and efficient. When we combine these two modules, SAL will provide stronger prototypes for RAT, leading to better results, *i.e.*, a saving of 8.50% BD-rate.

With our proposed modules, Ours-S achieves high performance with a significantly smaller parameter count. Compared to ChARM [34], Ours-S saves nearly 40% parameters and has faster speed while achieving comparable performance. Ours-S is a CNN-based model without any self-attention module. And the entropy estimation is just the hyperprior-style [6], without the efficient but time-consuming auto-regressive architecture. It indicates that the proposed modules RAT and SAL are highly effective and lightweight.

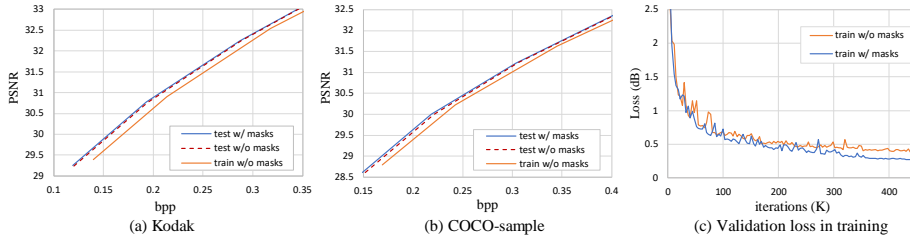


Fig. 8: We measure the impact of the masks separately in both the training and testing phases. “test w/ masks” means we adopt the masks in both training and testing, “test w/o masks” means we adopt the masks only in training, while “train w/o masks” means we do not use masks all the time. Naturally, if the masks are not used, 4×4 grid partitions will be used as alternatives.

4.4 Subjective Results

Fig. 7 shows the subjective results compared with VTM and Zou2022. As we can see, with the help of the segmentation masks in training, our SegPIC can distinguish the objects’ contours more accurately, making the edges sharper with less bitrate. In *kodim04*, we can see the wool reconstructed by SegPIC is more consistent. In *kodim24*, the bird patterns decoded by SegPIC are more clear, and the blurrings are less than in other methods. These results are based on grid partitions used as alternatives to masks. It demonstrates that SegPIC has acquired a more robust representative capability for various objects. Consequently, SegPIC can discern objects in complex textures more effectively.

4.5 Analysis of the Role of Segmentation Masks

Quantitative Results. To validate the role of the masks, we make experiments on two datasets, Kodak and COCO-sample. Due to Kodak not being a segmentation dataset, the corresponding masks are generated using the segmentation method [36]. Otherwise, we randomly sample 20 images and their masks from COCO Valid 2017 [9] to create the COCO-sample dataset.

As illustrated in Fig. 8 (a) (b), during the testing (inference) phase, the differences caused by masks are not so evident. However, for training, the situation is entirely different. With the assistance of the masks, the model achieves significantly better performance, especially at low bitrates. It could imply that, due to bitrate loss constraints during training, the information in the prototypes is implicitly transferred to the learnable parameters in RAT, allowing limited contextual information to still be effective during the inference phase. At low bitrates, since the lack of sufficient high-frequency information, pixels in the region exhibit stronger similarity, enhancing the representational ability of features extracted by prototypes. Fig. 8 (c) shows the early stage of model training.

As shown in Tab. 3, the model training w/o masks (the last column) leads to noticeable degradation. However, training w/ masks has a relatively minor

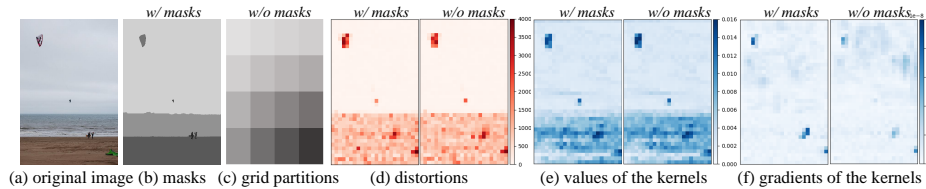


Fig. 9: Visualization of the kernels generated by DKG in RAT at the Encoder (see Fig. 3), in the cases with and without the masks. Fig. (d)(e)(f) are generated by the same pre-trained network weights. During testing, they go through processes with or without masks, respectively. The distortions in (d) can be formulated as $(x - \hat{x})^2$. We take the maximum absolute values of the tensors along the channels and apply 2D max pooling on them for better visualization.

Table 3: The effect of the masks and the grid number on BD-rate on COCO-sample. “n×” denotes $n \times n$ grid partitions to replace masks.

Train	train w/ masks				4×
Test	test w/ masks	2×	4×	8×	4×
BD-rate ↓	-0.11	+0.21	0(<i>default</i>)	+0.46	+2.10

impact on the pattern and number in testing. So we can replace the masks in testing to further save bitrate but should take the masks as privileged information during training. The hyperparameter 4× in testing is an empirical trade-off between the effect of RAT (*i.e.* image reconstruction precision) and the quantity of the prototypes (*i.e.* the additional bitrate they introduced).

Qualitative Analysis. Fig. 9 shows the visualization of the kernels generated by DKG in RAT at the Encoder (see Fig. 2). Although there is not a noticeable difference in Fig. 9 (e), it is obvious that the gradients of *w/ mask* are more focused on the semantic objects (see Fig. 9 (f)), which is complex and prone to causing distortions. This attribute will assist in optimizing the models, enabling it to better reconstruct various objects. These observations indicate that the masks do play a necessary role during the training phase.

5 Conclusion

We propose a novel image compression framework called Segmentation-Prior-Guided Image Compression (SegPIC), equipped with two proposed modules: Region-Adaptive Transform (RAT) and Scale Affine Layer (SAL). We adopt the class-agnostic masks as privilege information to assist the training of SegPIC. Experimental results demonstrate that our SegPIC is superior to the previous works. Otherwise, we analyze the role of the masks. Experiments and visualization results show that the masks can guide the model to focus more on semantic objects in training, leading to higher pixel-fidelity performance.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (No.2021ZD0112100), National NSF of China (No.62120106009), Guangdong Basic and Applied Basic Research Foundation (2024A1515010454), the Basic and Frontier Research Project of PCL, and the Major Key Project of PCL.

References

1. Workshop and challenge on learned image compression (clic2020) (2020), <http://www.compression.cc> 10, 11
2. Agustsson, E., Minnen, D., Toderici, G., Mentzer, F.: Multi-realism image compression with a conditional generator. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 22324–22333 (2023) 10
3. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. *IEEE Transactions on Computers* **100**(1), 90–93 (1974) 2
4. Akbari, M., Liang, J., Han, J.: Dsslic: Deep semantic segmentation-based layered image compression. In: ICASSP. pp. 2042–2046. IEEE (2019) 3
5. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: Int. Conf. Learn. Represent. (2017) 2, 5
6. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: Int. Conf. Learn. Represent. (2018) 2, 4, 5, 12
7. Bellard, F.: Bpg image format (2014), <https://bellard.org/bpg/> 2, 10, 11
8. Bjontegaard, G.: Calculation of average psnr differences between rd-curves. ITU SG16 Doc. VCEG-M33 (2001) 11
9. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and stuff classes in context. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1209–1218 (2018) 10, 13
10. Chang, J., Mao, Q., Zhao, Z., Wang, S., Wang, S., Zhu, H., Ma, S.: Layered conceptual image compression via deep semantic synthesis. In: IEEE Int. Conf. Image Process. pp. 694–698 (2019) 4, 5
11. Chang, J., Zhao, Z., Yang, L., Jia, C., Zhang, J., Ma, S.: Thousand to one: Semantic prior modeling for conceptual coding. In: Int. Conf. Multimedia and Expo. pp. 1–6. IEEE (2021) 3, 4, 5
12. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7939–7948 (2020) 3, 4, 10, 11
13. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1251–1258 (2017) 8
14. Daubechies, I.: The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory* **36**(5), 961–1005 (1990) 2
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Hounsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. Learn. Represent. (2021) 3
16. Feng, R., Gao, Y., Jin, X., Feng, R., Chen, Z.: Semantically structured image compression via irregular group-based decoupling. In: Int. Conf. Comput. Vis. pp. 17237–17247 (2023) 3, 5, 10

17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) *Adv. Neural Inform. Process. Syst.* vol. 27. Curran Associates, Inc. (2014) [4](#)
18. He, D., Yang, Z., Peng, W., Ma, R., Qin, H., Wang, Y.: Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 5718–5727 (2022) [10](#)
19. He, D., Zheng, Y., Sun, B., Wang, Y., Qin, H.: Checkerboard context model for efficient learned image compression. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 14771–14780 (2021) [4](#)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 770–778 (2016) [3](#)
21. Hoang, T.M., Zhou, J., Fan, Y.: Image compression with encoder-decoder matched semantic segmentation. In: *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.* pp. 160–161 (2020) [3](#)
22. Jia, X., De Brabandere, B., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. *Adv. Neural Inform. Process. Syst.* **29** (2016) [8](#)
23. Jiang, W., Yang, J., Zhai, Y., Ning, P., Gao, F., Wang, R.: Mlic: Multi-reference entropy model for learned image compression. In: *ACM Int. Conf. Multimedia.* pp. 7618–7627 (2023) [4](#)
24. (JVET), J.V.E.T.: Versatile video coding (2021), <https://jvet.hhi.fraunhofer.de/> [2](#), [11](#)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) [10](#)
26. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: *Int. Conf. Learn. Represent.* (2014) [2](#)
27. Kodak, E.: Kodak lossless true color image suite (photocd pcd0992) (1993), <http://r0k.us/graphics/kodak/> [10](#), [11](#)
28. Koyuncu, A.B., Gao, H., Boev, A., Gaikov, G., Alshina, E., Steinbach, E.: Contextformer: A transformer with spatio-channel attention for context modeling in learned image compression. In: *Eur. Conf. Comput. Vis.* pp. 447–463. Springer (2022) [10](#)
29. Li, F., Zhang, L., Liu, Z., Lei, J., Li, Z.: Multi-frequency representation enhancement with privilege information for video super-resolution. In: *Int. Conf. Comput. Vis.* pp. 12814–12825 (2023) [3](#)
30. Liu, J., Lu, G., Hu, Z., Xu, D.: A unified end-to-end framework for efficient deep image compression. arXiv preprint arXiv:2002.03370 (2020) [3](#), [4](#)
31. Liu, J., Sun, H., Katto, J.: Learned image compression with mixed transformer-cnn architectures. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 14388–14397 (2023) [4](#)
32. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: *Int. Conf. Comput. Vis.* pp. 10012–10022 (2021) [3](#), [4](#)
33. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. *Adv. Neural Inform. Process. Syst.* **31** (2018) [4](#)
34. Minnen, D., Singh, S.: Channel-wise autoregressive entropy models for learned image compression. In: *IEEE Int. Conf. Image Process.* pp. 3339–3343. IEEE (2020) [4](#), [5](#), [9](#), [10](#), [11](#), [12](#)
35. Pan, G., Lu, G., Hu, Z., Xu, D.: Content adaptive latents and decoder for neural image compression. In: *Eur. Conf. Comput. Vis.* pp. 556–573. Springer (2022) [10](#)

36. Qi, L., Kuen, J., Wang, Y., Gu, J., Zhao, H., Torr, P., Lin, Z., Jia, J.: Open world entity segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022) [13](#)
37. Shen, H., Zhao, Z.Q., Zhang, W.: Adaptive dynamic filtering network for image denoising. In: *AAAI*. vol. 37, pp. 2227–2235 (2023) [6](#)
38. Skodras, A., Christopoulos, C., Ebrahimi, T.: The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine* **18**(5), 36–58 (2001) [2](#)
39. Stark, H., Woods, J.W.: Probability, random processes, and estimation theory for engineers. Prentice-Hall, Inc. (1986) [2](#)
40. Sun, S., He, T., Chen, Z.: Semantic structured image coding framework for multiple intelligent applications. *IEEE Trans. Circuit Syst. Video Technol.* **31**(9), 3631–3642 (2020) [5](#)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Adv. Neural Inform. Process. Syst.* **30** (2017) [4](#)
42. Wallace, G.K.: The JPEG still picture compression standard. *Communications of the ACM* **34**(4), 30–44 (1991) [2](#)
43. Wang, D., Yang, W., Hu, Y., Liu, J.: Neural data-dependent transform for learned image compression. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 17379–17388 (2022) [2](#), [3](#), [4](#), [10](#), [11](#)
44. Wang, G.H., Li, J., Li, B., Lu, Y.: Evc: Towards real-time neural image compression with mask decay. In: *Int. Conf. Learn. Represent.* (2023) [10](#)
45. Wang, X., Yu, K., Dong, C., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 606–615 (2018) [8](#)
46. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. vol. 2, pp. 1398–1402. IEEE (2003) [4](#), [10](#)
47. Xu, Y.S., Tseng, S.Y.R., Tseng, Y., Kuo, H.K., Tsai, Y.M.: Unified dynamic convolutional network for super-resolution with variational degradations. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 12496–12505 (2020) [6](#)
48. Zhu, Y., Yang, Y., Cohen, T.: Transformer-based transform coding. In: *Int. Conf. Learn. Represent.* (2021) [3](#), [4](#)
49. Zou, R., Song, C., Zhang, Z.: The devil is in the details: Window-based attention for image compression. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 17492–17501 (2022) [3](#), [4](#), [5](#), [7](#), [9](#), [10](#), [11](#), [12](#)