IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 June 2024, accepted 10 July 2024, date of publication 18 July 2024, date of current version 29 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3430322

RESEARCH ARTICLE

Semantically-Guided Image Compression for Enhanced Perceptual Quality at Extremely Low Bitrates

SHOMA IWAI, (Graduate Student Member, IEEE), TOMO MIYAZAKI, (Member, IEEE), AND SHINICHIRO OMACHI, (Senior Member, IEEE)

Graduate School of Engineering, Tohoku University, Miyagi 980-8579, Japan

Corresponding author: Shoma Iwai (shoma.iwai.s4@dc.tohoku.ac.jp)

This work was supported in part by Japan Society for the Promotion of Science under Grant JP24KJ0343.

ABSTRACT Image compression methods based on machine learning have achieved high rate-distortion performance. However, the reconstructions they produce suffer from blurring at extremely low bitrates (below 0.1 bpp), resulting in low perceptual quality. Although some methods attempt to reconstruct sharp images using Generative Adversarial Networks (GANs), reconstructing natural textures at low bitrates remains challenging. In this paper, we propose a novel image compression method that explicitly utilizes semantic information. Specifically, we send a semantic label map to the decoder, which takes it as input. This semantic information enables the decoder to reconstruct appropriate textures consistent with the corresponding semantic classes. Although semantic label maps can be compressed into relatively small data sizes using common methods (e.g., PNG), the data size is not negligible in an extremely low-rate setting. To address this problem, we propose simple yet effective label map compression strategies, including an autoregressive label map compressor. Our strategies significantly reduce the data size of the label map while maintaining the critical semantic information that allows the decoder to reconstruct realistic and suitable textures. By utilizing this data-efficient semantic information, our method can reconstruct realistic images even at an extremely low bitrate. As a result, the proposed method outperformed existing models, including a GAN-based model designed for low-rate settings and a state-of-the-art semantically guided method, in both quantitative evaluation and user studies. Furthermore, we analyzed the effect of semantic information by switching the input label map, confirming that the model synthesized textures appropriate to the given semantic labels.

INDEX TERMS Image compression, semantic information, perceptual image compression, GANs, neural image compression.

I. INTRODUCTION

Lossy image compression is essential for efficient image storage and transmission. Many image compression methods have been developed over the past several decades, such as JPEG, BPG [1], and VVC [2], which are based on handcrafted algorithms. With the recent advancements in deep learning, neural-network-based lossy image compression methods have been developed [3], [4], [5]. Most such methods adopt an encoder-decoder architecture, and their parameters are optimized using large image datasets. State-of-the-art methods outperform traditional codecs, such as BPG and VVC, in terms of rate-distortion performance.

The challenge lies in extremely low-bitrate compression, which involves compressing images to bitstreams below 0.1 bpp (bits per pixel). Even state-of-the-art compression models [6] suffer from blurring at low bitrates, leading to poor perceptual quality, as shown in Fig. 1 (b). To address this, Generative Adversarial Networks (GANs [7]) have been employed to enhance sharpness in compressed images. However, achieving sharpness alone is inadequate for the reconstruction of realistic images, suggesting the need for

The associate editor coordinating the review of this manuscript and approving it for publication was Miaohui Wang.

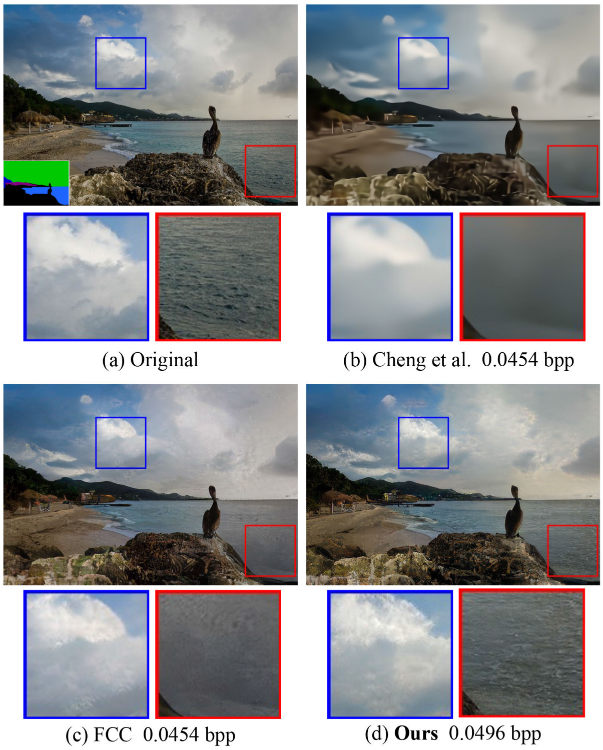


FIGURE 1. Comparison of compressed images using Cheng et al. [6], FCC [8], and our proposed method. While other methods suffer from blur or artifacts, our method reconstructs realistic textures by leveraging a semantic label map. Specifically, it appropriately renders textures of clouds and the sea, which are highlighted in blue and red squares.

more advanced solutions For example, as shown in Fig. 1 (c), the output of a representative GAN-based method, fidelity-controllable compression (FCC) [8] is sharper than that of Cheng et al. [6]. However, it looks unnatural owing to compression artifacts in a part of the image. This example shows that even sharp reconstructions do not look realistic unless their textures are semantically correct.

To synthesize appropriate textures even at low bitrates, we propose a semantically guided image compression model that leverages semantic information. We employ a semantic label map, enabling the model to reconstruct textures that are consistent with their semantic classes. Fig. 1 (d) shows the image compressed by our proposed method. As can be observed, it clearly retains the characteristic textures of the sea and clouds, free from blur or noise, unlike those produced by other methods.

Our proposed method achieves perceptually pleasing images at extremely low bitrates by compressing input images and semantic information. Existing methods [9], [10], [11], [12], [13] require a full-resolution semantic label map. Thus, they are unsuitable for the low bitrates. To address this problem, we introduce three strategies: (1) downscaling the label map, (2) reducing the number of classes in the label maps, and (3) using an autoregressive compression model to compress label maps losslessly. These strategies reduce the average bitrate of the label maps to 0.001 bpp on the COCO dataset [14].

In our experiments, we conducted quantitative evaluations and user studies on both a general dataset (COCO) and a domain-specific dataset (Cityscapes [15]). In both datasets, our method surpassed existing methods, including a state-of-the-art semantically guided compression method [11]. The results showed that even with its small data size, semantic information significantly enhanced perceptual quality. Futhermore, our analysis demonstrated that the model generates textures aligned with the input semantic classes.

The contributions of this study are summarized as follows.

- We propose a novel GAN-based image compression method that utilizes semantic information for extremely low bitrate compression. The semantic information enables the decoder to synthesize textures aligned with their semantic classes.
- We introduce three label map compression strategies. These strategies reduce overhead data size drastically and realize extremely low bitrate compression.
- We analyzed the effect of semantic information in our method by switching semantic classes. The results demonstrate that the proposed method synthesizes appropriate textures corresponding to the input semantic class.

II. RELATED WORKS

A. LEARNED IMAGE COMPRESSION

Over the last several years, image compression methods based on machine learning have been developed [3], [4], [5], [6], [16], [17], [18], [19]. Balle et al. [3] developed an end-to-end compression method for the first time. Some works have investigated powerful entropy models for effective compression. Balle et al. [4] developed hyperprior networks designed to utilize side information. Minnen et al. [5] and Lee et al. [18] adopted an autoregressive context model to utilize information from known subsets. Cheng et al. [6] introduced a Gaussian mixture model to parameterize the distributions of latent codes, improving entropy estimation performance. Other approaches have focused on the architecture of the encoder and decoder. Some methods have used RNN-based architectures [16], [17], [20] instead of CNN. Attention modules have also been used. Chen et al. [21] introduced a non-local attention module. Some works [22], [23], [24] adopt Swin-Transformer [25]-based architectures, improving performance further. Thanks to these advancement, state-of-the-art methods [24], [26], [27] outperform Versatile Video Coding (VVC) [2], the latest coding standard. However, these methods are trained to optimize the rate-distortion trade-off. In extremely low-bitrate settings, the outputs tend to be blurry, resulting in low perceptual quality. By contrast, our proposed method is trained to improve perceptual quality by using GAN-based training and semantic information.

B. GAN-BASED IMAGE COMPRESSION

Some works have utilized GAN models to reconstruct visually pleasing images [8], [10], [28], [29], [30], [31], [32], [33], [34], [35]. Rippel and Bourdev [28] introduced adversarial training for image compression. Agustsson et al. [10] used a GAN to reconstruct realistic images at an extremely low bitrate. Iwai et al. [8] utilized a two-stage training method to avoid unstable training, and applied network interpolation to control the effects of a GAN. Mentzer et al. [29] developed a high-fidelity GAN-based compression method. Inspired by Vector-Quantized GANs (VQGANs) [36] vector-quantization-based methods have also been proposed [33], [34], [35]. Following the recent success of the diffusion model [37], some diffusion-based image compression methods [38], [39], [40], [41], [42] have been proposed. While they achieve high perceptual quality, they require computationally expensive iterative process to decode an image.

Although these methods can reconstruct sharp images, they do not explicitly consider the semantic structure of an input image. This leads to unnatural textures, as shown in Fig. 1 (c). By contrast, we introduce a semantically guided decoder to reconstruct natural and semantically accurate textures.

C. SEMANTIC GUIDED IMAGE COMPRESSION

Some existing image compression schemes do consider semantic information. These methods can be divided into two groups, including those that focus on the performance of downstream tasks (e.g., classification) after compression and those that focus on the perceptual quality of the reconstructed images. The present work belongs to the latter group.

1) METHODS THAT FOCUS ON PERFORMANCE ON DOWNSTREAM TASKS

Patwa et al. [43] utilized classification loss to develop an image compression method designed to preserve semantics. Le et al. [44] proposed training strategies to balance three loss functions: rate loss, distortion loss, and task-specific loss. Sun et al. [45] proposed a semantically structured image-coding framework. In this framework, intelligent tasks, such as classification and pose estimation, can be performed without decoding an entire image. Yan et al. [46] proposed semantics-to-signal scalable image coding (SSSIC). This method stores deep features of an image to perform downstream tasks and reconstruct images. TransTIC [47] incorporates a visual-prompt tuning [48] technique to improve performance on different downstream tasks while keeping the original network weights fixed. Although these methods are designed to achieve high performance on downstream tasks, the proposed approach aims to reconstruct visually pleasing images.

2) METHODS THAT FOCUS ON PERCEPTUAL QUALITY

Several methods [9], [10], [11], [12], [13], [49], [50] have been proposed to improve the perceptual quality of reconstructions by using semantic information. Wang et al. [9] utilized Grad-CAM [51] to locate semantically important regions and compensate for details. Chang et al. [49] used edge detection to extract a structural map, helping the decoder reconstruct an image. However, these methods do not explicitly use the semantic classes of images. Agustsson et al. [10] have proposed a GAN-based image compression method that reconstructs images from latent code and feature maps extracted from a semantic label map. However, their approach used semantic label maps only for controlling the bit allocation. Akbari et al. [13] developed a compression framework that utilized a semantic label map to enhance the quality of the decoded images. Duan et al. [11] designed a semantically guided compression framework. It can be applied to any image codec because it enhances the quality of already decoded images using the corresponding semantic label map. However, it applies post-processing using a Pix2PixHD network [52] for decoded images, which makes the entire pipeline complex and computationally inefficient. Chang et al. [12] proposed a coding scheme based on a semantic prior. It stores one representative vector for each semantic class of the input image, enabling extreme compression. However, it is limited to simple and low-resolution image applications due to the lack of expressive ability of the representative vectors. Their follow-up work [50] has introduced a consistency-contrast regularization to improve textural consistency between the original and reconstructed images.

The main difference between our method and existing works is the label map compression strategies. Specifically, we introduce three label map compression strategies: downscaling the label map, reducing the number of classes, and an autoregressive label map compressor. These strategies reduce the data size of the label maps to around 0.001 bpp on the COCO dataset. Although JPD-SE [11] compresses the semantic maps by converting them into polygons, the average bitrate of compressed semantic maps is 0.03 bpp, which is too large in extremely low-bitrate settings (below 0.1 bpp). Our experimental results demonstrate that the semantic label map helps our model reconstruct natural textures despite its small data size. Furthermore, we analyzed the effect of semantic information by switching semantic classes, confirming that our model synthesized texture aligned with the input semantic classes.

D. GAN-BASED IMAGE SYNTHESIS FROM SEMANTIC LABEL MAPS

Several works have studied GAN-based image synthesis from a semantic label map [52], [53], [54], [55], [56]. Isola et al. [53] developed a model based on U-Net to generate realistic images. Wang et al. [52] used an autoencoder-based model to synthesize high-resolution images. Park et al. [54] proposed spatially adaptive-normalization (SPADE), in which semantic label maps are injected into each SPADE layer. Zhu et al. [55] developed

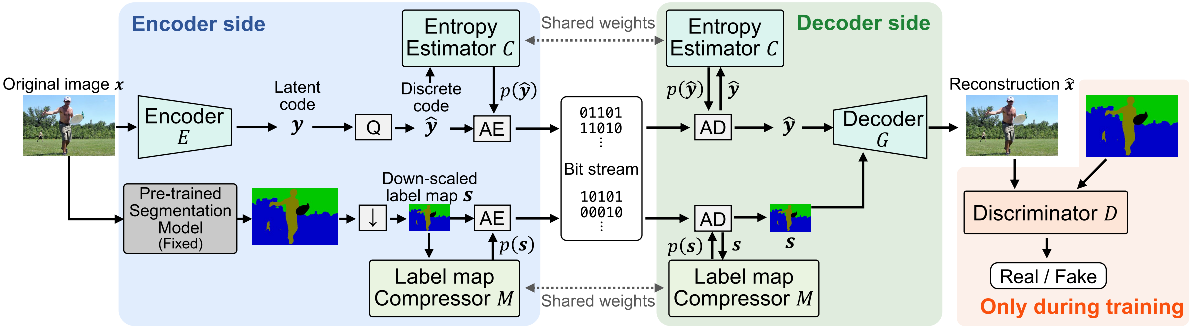


FIGURE 2. Overview of the proposed method. On the encoder side, the original image and down-scaled label map are transformed into a bitstream. To reduce the data size of the label map, we introduce a label map compressor M. On the decoder side, the bit stream is decoded into quantized latent code and label map, and then the decoder reconstructs an image. During training, the discriminator distinguishes the original image from reconstructions. “Q”, “AE”, and “‘AD’ denote quantization, arithmetic encoder, and arithmetic decoder, respectively.

a method referred to as SEAN-normalization layers, similar to SPADE, which utilizes style and semantic information to generate high-fidelity images. Schönfeld et al. [56] adopted a U-Net [57]-like discriminator for adversarial training. The discriminator was trained to predict semantic classes. This allows the generator to be trained by only an adversarial loss, which improves the quality of generated images. Semantic label maps are also utilized for super-resolution tasks. Wang et al. [58] developed an SFT layer designed to renormalize a feature map using semantic information. They also proposed a method using only eight classes and segmentation probability maps instead of a one-hot segmentation map. Using segmentation probability maps as prior, the model learns class-specific features and textures.

III. PROPOSED METHOD

In this section, we present the pipeline of our semantically-guided image compression method. First, we provide an overview and outline the procedure of the proposed method in Sec. III-A. Next, we elaborate on the details of each component of the model in Sec. III-B through Sec. III-F. Finally, we describe the training strategy in Sec. III-G.

A. MODEL OVERVIEW

As shown in Fig. 2, our method consists of five components: an encoder E, an entropy estimator C, a label map compressor M, a decoder G, and a discriminator D. The encoding and decoding processes are as follows: On the encoder side, the encoder extracts a latent code $\boldsymbol{y}$ from the original image $\boldsymbol{x}$, which is then quantized into a discrete code $\hat{\boldsymbol{y}}$. Using a pre-trained semantic segmentation model, a label map of the original image is obtained and down-scaled into $\boldsymbol{s}$. We use DeepLab v3 [59] for segmentation, and its weights are fixed. The discrete code $\hat{\boldsymbol{y}}$ and down-scaled label map $\boldsymbol{s}$ are transformed into a bitstream through entropy coding, with the entropy estimator C and the label map compressor M used to estimate their probability distributions $p(\hat{\boldsymbol{y}})$ and $p(\boldsymbol{s})$, respectively. The bitstream is then transmitted to the decoder side. On the decoder side, the same entropy estimator C and label map compressor M are used to entropy-decode the bitstream, recovering $\hat{\boldsymbol{y}}$ and $\boldsymbol{s}$. The decoder G reconstructs the image $\hat{\boldsymbol{x}}$ from $\hat{\boldsymbol{y}}$ and $\boldsymbol{s}$. The discriminator D is employed during GAN-based training to improve the perceptual quality of the reconstructed images; however, it is not used during inference.

B. ENCODER

The encoder transforms the original image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$ into a latent code $\boldsymbol{y} \in \mathbb{R}^{H/16\times W/16\times C_y}$, where H, W are the height and width of the image, and C_y denotes the number of channels of the latent code. We employ the encoder used in [6], which consists of six residual blocks, two attention modules, and two convolutional layers. These blocks and layers extract the deep feature from the input image. The latent code $\boldsymbol{y}$ is then quantized, obtaining a discrete code $\hat{\boldsymbol{y}}$. Since the gradient of the quantization function (*i.e.*, ROUND$(\cdot)$) is zero almost everywhere, actual quantization cannot be applied during training. Thus, we adopt additive uniform noise during training and use the actual quantization during inference, as used in prior arts [3], [4]:

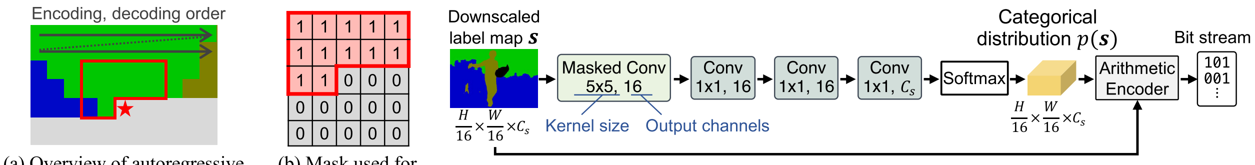
$$\hat{\boldsymbol{y}} = \begin{cases} \boldsymbol{y} + \Delta\boldsymbol{y} & \text{(training)} \\ \mathrm{ROUND}(\boldsymbol{y}) & \text{(inference)} \end{cases} \tag{1}$$

$$\Delta\boldsymbol{y} \sim \mathcal{U}\left(-0.5, 0.5\right), \tag{2}$$

where $\Delta\boldsymbol{y}$ represents uniform noise and $\mathcal{U}(-0.5, 0.5)$ denotes a uniform distribution over the interval $[-0.5, 0.5]$.

C. ENTROPY ESTIMATOR

The entropy estimator predicts the probability distribution $p(\hat{\boldsymbol{y}})$ of the quantized code $\hat{\boldsymbol{y}}$. This probability distribution is used to approximate the bitrate during training and for entropy coding during inference. We adopt an entropy estimator used in [6], which models each element of $\hat{\boldsymbol{y}}$ using a Gaussian Mixture Model (GMM) with K mixtures. Specifically, the entropy estimator predicts the weight $\boldsymbol{w} \in \mathbb{R}^{n\times K}$, mean $\mu \in \mathbb{R}^{n\times K}$, and standard deviation $\sigma \in \mathbb{R}^{n\times K}$, where $n = \frac{H}{16} \times \frac{W}{16} \times C_y$ denotes the number of elements in $\hat{\boldsymbol{y}}$.



(a) Overview of autoregressive encoding and decoding

(b) Mask used for 5 × 5 Masked Conv

(c) Architecture of our label map compressor

FIGURE 3. (a) Overview of the autoregressive label map encoding and decoding. The label map is processed in a pixel-by-pixel manner from top left to bottom right. ★ indicates the current position, whose entropy is estimated based on the pixels enclosed by the red outline. (b) A mask used to filter a 5 × 5 Masked Convolution layer. The mask forces the convolution kernels to extract features only from previously encoded or decoded pixels. (c) The architecture of our semantic label map compressor. Given a down-scaled label map *s*, it predicts categorical distribution *p*(*s*) in an autoregressive manner. Finally, it is transformed into a bit stream using an arithmetic encoder based on *p*(*s*).

These parameters are predicted using a hyperprior [4] and a context model [5]. For the detailed process of this parameter prediction, please refer to [6]. Given the estimated parameters $\boldsymbol{w}$, μ, σ, the probability $p(\hat{\boldsymbol{y}})$ can be calculated as follows:

$$p(\hat{\boldsymbol{y}}\,|\,\boldsymbol{w},\mu,\sigma)=\prod_{i=1}^{n}p(\hat{y}_i\,|\boldsymbol{w}_i,\mu_i,\sigma_i) \tag{3}$$

$$p(\hat{y}_i\,|\,\boldsymbol{w}_i,\mu_i,\sigma_i)=\left(\sum_{k=1}^{K}w_{i,k}\mathcal{N}\left(\mu_{i,k},\sigma_{i,k}^2\right)*\mathcal{U}\left(-\frac{1}{2},\frac{1}{2}\right)\right)(\hat{y}_i). \tag{4}$$

The approximated bitrate R (*i.e.*, the number of bits per pixel) is the sum of the bitrate of $\hat{\boldsymbol{y}}$ and the bitrate of the additional discrete latent variable $\hat{z}$, which is used in the hyperprior [4]. We use a learned factorized prior distribution [4] to model $p(\hat{z})$. As a result, R can be represented as follows:

$$R(\hat{\boldsymbol{y}},\hat{z})=-\frac{1}{HW}(\log_2 p(\hat{\boldsymbol{y}}\,|\,\boldsymbol{w},\mu,\sigma))+\log_2 p(\hat{z})). \tag{5}$$

We use R as a loss function to reduce the bitrate. During inference, the probability distributions $p(\hat{\boldsymbol{y}})$ and $p(\hat{z})$ are used for entropy coding to translate $\hat{\boldsymbol{y}}$ and $\hat{z}$ into bitstream.

D. SEMANTIC LABEL MAP COMPRESSOR

To transmit the semantic label map $\boldsymbol{s}$ to the decoder side, we employ a lossless compression using our semantic label map compressor M. We introduce three strategies to effectively reduce the data size.

First, we downscale the semantic label map to $\frac{H}{16}\times\frac{W}{16}$. Although it causes information loss, we will demonstrate that this downscaling does not affect compression performance in our experiment in Sec. IV-G.

Second, we reduce the number of classes in the label maps. This strategy is inspired by a semantically-guided image super-resolution method [58], where eight classes are chosen: *sky*, *plant*, *water*, *animal*, *building*, *mountain*, *grass*, and *others*. The *others* represent pixels not fitting within the first seven classes. In addition to the eight classes, we have also added "person" and "road" categories. Furthermore, we have merged "plant" and "grass" into a single "plant" category for simplification. As a result, we obtain nine classes: *sky*, *plant*, *water*, *animal*, *building*, *mountain*, *person*, *road*, and *others* (representing pixels not fitting within the first eight classes). This strategy has two advantages. Firstly, it reduces the data size of the label maps. Secondly, it makes training simpler by removing rare and fine-grained classes.

Third, we introduce an autoregressive label map compressor M to effectively reduce spatial redundancy, as shown in Fig. 3. Fig. 3(a) provides an overview of the autoregressive encoding and decoding processes. The label map is entropy-encoded and -decoded from top left to bottom right in a pixel-by-pixel manner. For entropy coding, the entropy of each pixel is predicted based on previously encoded or decoded pixels. To achieve this, we employ the masked convolution layer [5], [60]. In the masked convolution, the kernels are masked using a matrix, as shown in Fig. 3(b), which forces the convolution layer to refer only to already encoded or decoded pixels. The overall architecture of our label map compressor is depicted in Fig. 3 (c). Given a down-scaled C_s-channel one-hot semantic label map $\boldsymbol{s}$, the 5 × 5 masked convolution first extracts features. Then, three 1 × 1 convolution layers predict the categorical distribution of the semantic class for each pixel as $p(\boldsymbol{s})\in\mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C_s}$. During inference, this $p(\boldsymbol{s})$ is used to compress and decompress the label map losslessly with entropy coding. During training, the label map compressor is optimized to minimize the cross-entropy loss R_{seg}:

$$R_{seg}=-\frac{1}{HW}\sum_{i=1}^{H/16}\sum_{j=1}^{W/16}\sum_{c=1}^{C_s}s_{i,j,c}\log_2(p(\boldsymbol{s})_{i,j,c}), \tag{6}$$

which corresponds to the approximated bitrate of the label map.

It is worth noting that autoregressive models are typically slow due to their pixel-by-pixel processing. However, our downscaling strategy mitigates this problem by reducing the number of forward processes from $H\times W$ to $\frac{H}{16}\times\frac{W}{16}$, resulting in a 99.6% reduction in computational cost.

With our three strategies, the proposed approach can compress the semantic label maps to ∼ 0.001 bpp on average in the COCO-Stuff validation dataset, which is approximately 160 times smaller than that of without our strategies. We discuss the effect of each strategy in Sec. IV-F.

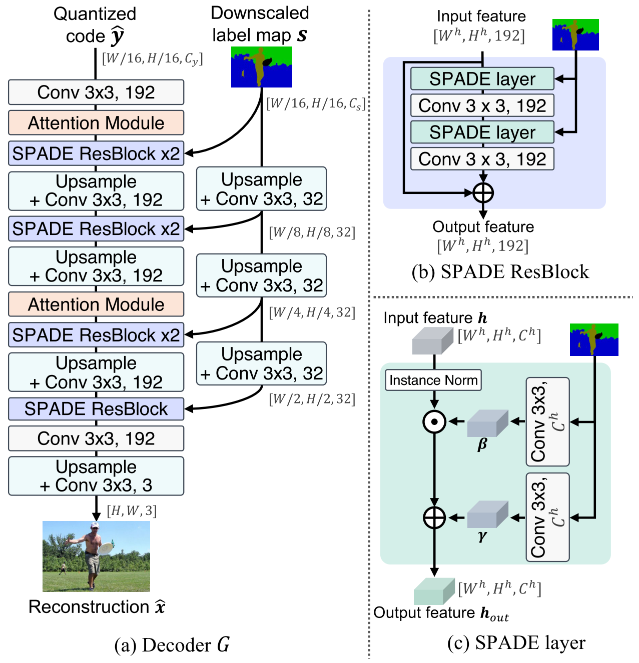


FIGURE 4. (a) Architecture of our decoder. Given a quantized code $\hat{y}$ and label map s, the decoder reconstructs the image $\hat{x}$. (b) The structure of the SPADE ResBlock. (c) SPADE layer. The semantic feature is integrated into the input feature through this layer. ⊙ and ⊕ indicate element-wise product and sum, respectively.

E. DECODER

Given the transmitted quantized latent code $\hat{y}$ and the downscaled semantic label map s, the decoder reconstructs an image $\hat{x}$. The architecture of the decoder is illustrated in Fig. 4 (a). Our decoder's foundational architecture is based on [6], which incorporates simplified attention modules [6], upsampling layers, and residual blocks, all configured with 192 channels. However, we replace the regular residual blocks with SPADE residual blocks (Fig. 4 (b)) [54] to inject semantic information.

Each SPADE residual block consists of two convolution layers and two SPADE layers (Fig. 4 (c)) [54]. The SPADE layer transforms the input intermediate feature map $h \in \mathbb{R}^{H^h \times W^h \times C^h}$ into the modulated feature map $h^{\text{out}} \in \mathbb{R}^{H^h \times W^h \times C^h}$ using the input semantic label map s, where H^h, W^h, C^h are the height, width, and the number of the channels, respectively. Initially, channel-wise normalization (Instance Normalization [61]) is applied to h. Then, two 3×3 convolutional layers g_γ and g_β extract two feature maps $\gamma \in \mathbb{R}^{H^h \times W^h \times C^h}$ and $\beta \in \mathbb{R}^{H^h \times W^h \times C^h}$ from s, respectively. Finally, a pixel-wise affine transformation is applied to the normalized h using γ and β, resulting in h^{out}. Formally, each pixel in h^{out} is formulated as follows:

$$h^{\text{out}}_{i,j,c} = \beta_{i,j,c} \left(\frac{h_{i,j,c} - \mu_c}{\sigma_c} \right) + \gamma_{i,j,c} \tag{7}$$

$$\mu_c = \frac{1}{H^h W^h} \sum_{i,j} h_{i,j,c} \tag{8}$$

$$\sigma_c = \sqrt{\frac{1}{H^h W^h} \sum_{i,j} \left((h_{i,j,c})^2 - (\mu_c)^2 \right)} \tag{9}$$

$$\gamma = g_\gamma(s) \tag{10}$$

$$\beta = g_\beta(s), \tag{11}$$

where i, j, c denote the spatial and channel indices of the feature map, and μ_c, σ_c represent the mean and standard deviation of h at c-th channel, respectively. In this way, the SPADE layer reflects semantic information in the intermediate feature map in the decoder.

In contrast to SPADE [54] and other semantic-guided image compression methods [10], [11], we use a $\frac{1}{16}$ downscaled semantic label map. To seamlessly integrate this downscaled semantic information into the decoding process, we incorporate three additional up-sampling blocks within our decoder architecture, as depicted in Fig. 4(a). These up-sampling blocks scale the semantic features to match the spatial resolution of the intermediate feature map h.

By applying the SPADE layer multiple times, the decoder enriches the intermediate features according to the input label map s at different scales, leading to semantically correct textures. Finally, the reconstruction $\hat{x}$ is obtained as an output of the decoder.

F. DISCRIMINATOR

We adopt GAN [7] to improve the perceptual quality of reconstructions. In GAN-based training, a discriminator D learns to distinguish real images x from reconstructions $\hat{x}$, while the compression model learns to output images that are indistinguishable from D. In our method, D takes a corresponding semantic label map s as well as the real or fake image as inputs, as shown in Fig. 2. Feeding a semantic label map into the discriminator enables D to evaluate the alignment between the input image and the semantic label map, encouraging the compression model to reconstruct an image with semantically appropriate texture.

Moreover, we adopt a multi-scale patch discriminator, which has three sub-discriminators D_1, D_2, D_3 as in other GAN-based methods [52], [54]. The sub-discriminators have the same simple CNN-based architecture [52] but take different scales of images as input. Specifically, while D_1 takes an image and its label map with a full resolution, D_2 and D_3 take $\frac{1}{2}$ and $\frac{1}{4}$ down-scaled images and label maps, respectively. Note that, since the discriminator is used only in the training, we use full-resolution label maps instead of the down-scaled ones employed in the decoder. This approach enables a more nuanced discrimination process, facilitating the generation of semantically coherent reconstructions.

G. TRAINING AND LOSS FUNCTIONS.

In this section, we explain how to train our image compression model and label map compressor. Since the label maps are compressed in a lossless manner, this label map

TABLE 1. Comparison of the baseline methods and our method, highlighting the key features and architectural differences between each. *Since the original implementation of polygon-based label map compression is unavailable, an autoregressive model was used in our experiments.

Method	Deep-learning-based	GAN-based	Semantically-guided	Label map compression	Decoder architecture
BPG [1]					
Cheng-CVPR [6]	✓				ResBlocks + Attention
FCC [8]	✓	✓			ResBlocks + Attention
JPD-SE-p2, p3 [11]	✓	✓	✓	*Polygon-based	Pix2PixHD [52]
Ours	✓	✓	✓	Autoregressive model + Down-scaling + Class reduction	SPADE-ResBlocks [54] + Attention

compression does not affect the optimization process of the image compression model. Consequently, the label map compressor is not employed during the image compression model's training phase. Instead, we train the label map compressor independently to minimize R_{seg} in (6).

For the training of the image compression model, we adopt a two-stage training strategy as in [8] and [62]. In the first stage, we train the encoder E, the entropy estimator C, and the decoder G without GAN loss. Specifically, the model is optimized to minimize the following rate-distortion loss function:

$$\min_{E,C,G} \mathcal{L}_{1st} = \mathbb{E}[R(\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) + \lambda_{\text{mse}}^{(1)} \text{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}})], \tag{12}$$

where $\lambda_{\text{mse}}^{(1)}$ is a hyperparameter, MSE denotes a Mean Squared Error, and R is the approximate bitrate defined in (5).

The second stage is GAN-based training, where the compression model and the discriminator D are trained adversarially. For the training of the compression model, we fix the weight of E and C and finetune only the decoder G as in [8] and [62]. We use VGG perceptual loss $\mathcal{L}_{\text{vgg}}$ [63], MSE loss, and adversarial loss $\mathcal{L}_{adv}^{G}$. Since the encoder side of the compression model is fixed, the rate term R is omitted in this stage. This omission makes the optimization simpler, leading to stable training. For the adversarial loss $\mathcal{L}_{adv}^{G}$, we use Least-square GAN (LSGAN) [64]. These loss functions for the compression model are defined as follows:

$$\begin{aligned}\min_{G} \mathcal{L}_{2nd} &= \mathbb{E}[\mathcal{L}_{\text{vgg}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_{\text{mse}}^{(2)} \text{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) \\ &\quad + \lambda_{adv} \mathcal{L}_{adv}^{G}(\hat{\boldsymbol{x}}, \boldsymbol{s})] \qquad (13)\\ \mathcal{L}_{adv}^{G}(\hat{\boldsymbol{x}}, \boldsymbol{s}) &= (D(\hat{\boldsymbol{x}}, \boldsymbol{s}) - 1)^2, \qquad (14)\end{aligned}$$

where $\lambda_{\text{mse}}^{(2)}$ and λ_{adv} are hyperparameters. The discriminator D is trained to minimize an adversarial loss $\mathcal{L}_{adv}^{D}$:

$$\min_{D} \mathcal{L}_{adv}^{D}(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{s}) = \frac{1}{2}\mathbb{E}[(D(\hat{\boldsymbol{x}}, \boldsymbol{s}) - 0)^2 + (D(\boldsymbol{x}, \boldsymbol{s}) - 1)^2]. \tag{15}$$

In this way, D learns to predict 1 for the original images and 0 for the reconstructed images.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) DATASET

We trained and evaluated our model using COCO [14] and Cityscapes [15] datasets. The COCO dataset, contains various images, including indoor and outdoor scenes, and comprises 183 classes. From these, we selected nine classes: *animal, building, mountain, person, plant, road, sky, water*, and *others*, as described in Sec. III-D. The Cityscapes dataset is an urban scene dataset containing object categories like roads, cars, and pedestrians. We used all 34 classes in this dataset. During training, we randomly extracted 256 × 256 patches from both the original images and their ground truth label maps. For evaluation, label maps predicted by a pre-trained DeepLabV3 model were used unless specified otherwise.

2) IMPLEMENTATION DETAILS

Following [6], we set the number of Gaussian mixtures $K = 3$. We set the number of channels of the latent code $C_y = 192$. For the first-stage training, we used different $\lambda_{\text{mse}}^{(1)}$ values to obtain three models with varying bitrates. Specifically, we used $\lambda_{\text{mse}}^{(1)} = \{0.4, 0.75, 1.5\}$ and $\lambda_{\text{mse}}^{(1)} = \{0.75, 1.5, 5.0\}$ for COCO and Cityscapes datasets, respectively. For the second-stage, we set $\lambda_{\text{mse}}^{(2)} = 0.5$ and $\lambda_{adv} = 0.05$ for both datasets. Consistent settings for the optimizer, learning rate, training steps, and batch size were maintained across both stages. Specifically, the model was trained for 500,000 steps using the Adam optimizer, with a batch size of 8. The initial learning rate was 1×10^{-4}, reduced to 1×10^{-5} for the final 100,000 iterations.

3) BASELINE METHODS

We conducted a performance comparison of our method against several baseline methods, summarized in Table 1. The detailed descriptions are as follows:

- **BPG** [1] is a non-learning-based codec.
- **Cheng-CVPR** [6] is a state-of-the-art PSNR-oriented compression method that does not incorporate GAN.
- **FCC** [8] is a GAN-based compression method designed for extremely low-bit-rate compression. The fundamental architecture, including the encoder and entropy

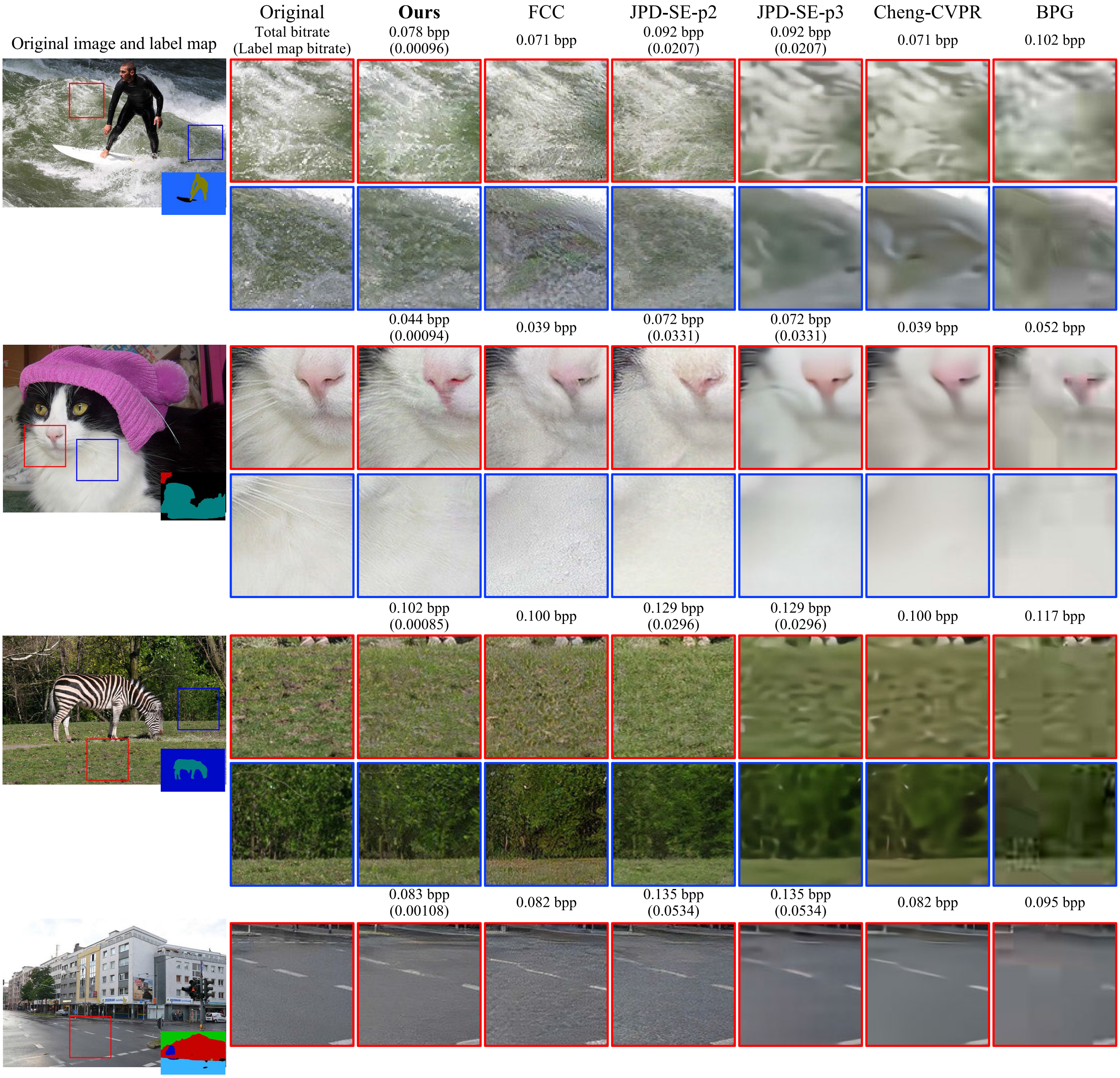


FIGURE 5. Original image and zoomed patches of the reconstructions of our method and existing methods on the COCO dataset. For each reconstruction, the number represents the bitrate (bits per pixel) after compression. For *Ours* and *JPD-SE-p2,3*, we also report the bitrates of the label maps.

model, is the same as that of our model. The key distinctions lie in our use of semantic information through SPADE ResBlocks and our label map compression strategies, which are central to our approach.

- **JPD-SE-p2 and -p3** [11] is the state-of-the-art semantically-guided image compression method. It employs semantically-guided post-processing to improve the quality of decoded images. Since it requires an external compression model, we used *Cheng-CVPR* for this role. Consistent with the original implementation, we use full-resolution label maps. As the original polygon-based label map compression algorithm is not publically available, we used an autoregressive model to compress label maps losslessly. The training of JPD-SE consists of three phases, with the initial two phases leveraging GAN-based training and the final phase minimizing only distortion loss to mitigate artifacts. The final JPD-SE model is denoted as *JPD-SE-p3*. However, the third phase tended to remove desirable details at extremely low bitrates, leading us

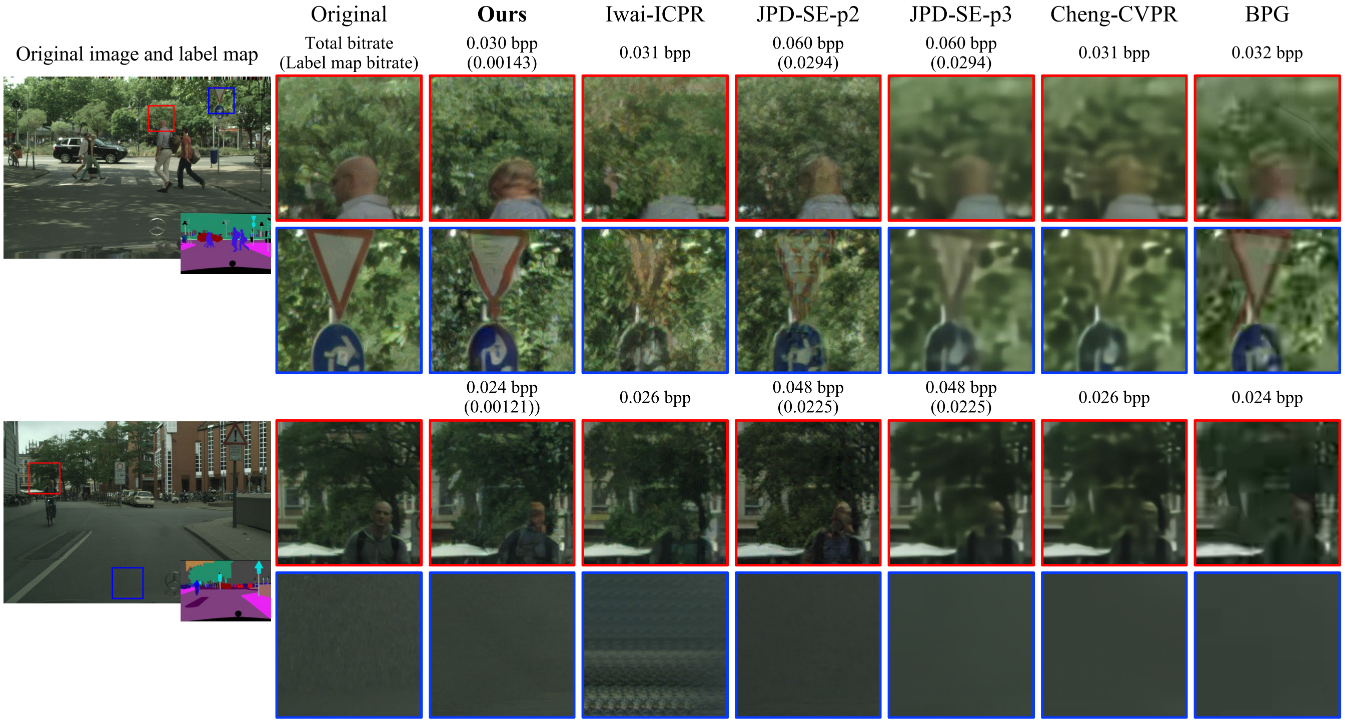
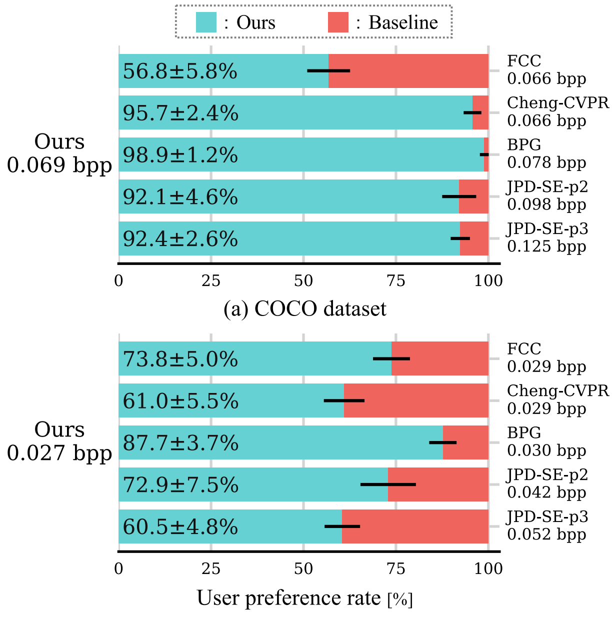


FIGURE 6. Original image and zoomed patches of the reconstructions of our method and existing methods on the Cityscapes dataset. The numbers represent the bitrate. For *Ours* and *JPD-SE-p2,3*, we also report the bitrates of the label maps.



(a) COCO dataset

(b) Cityscapes dataset

FIGURE 7. Results of the user study on (a) COCO and (b) Cityscapes dataset, comparing our method with existing methods. Values in the bars represent the user preference percentage for our method along with the 95% confidence interval.

to include results from the *JPD-SE-p2* model, which retains more detail.

For each method, including the baselines and ours, we trained three distinct models targeting different bitrates.

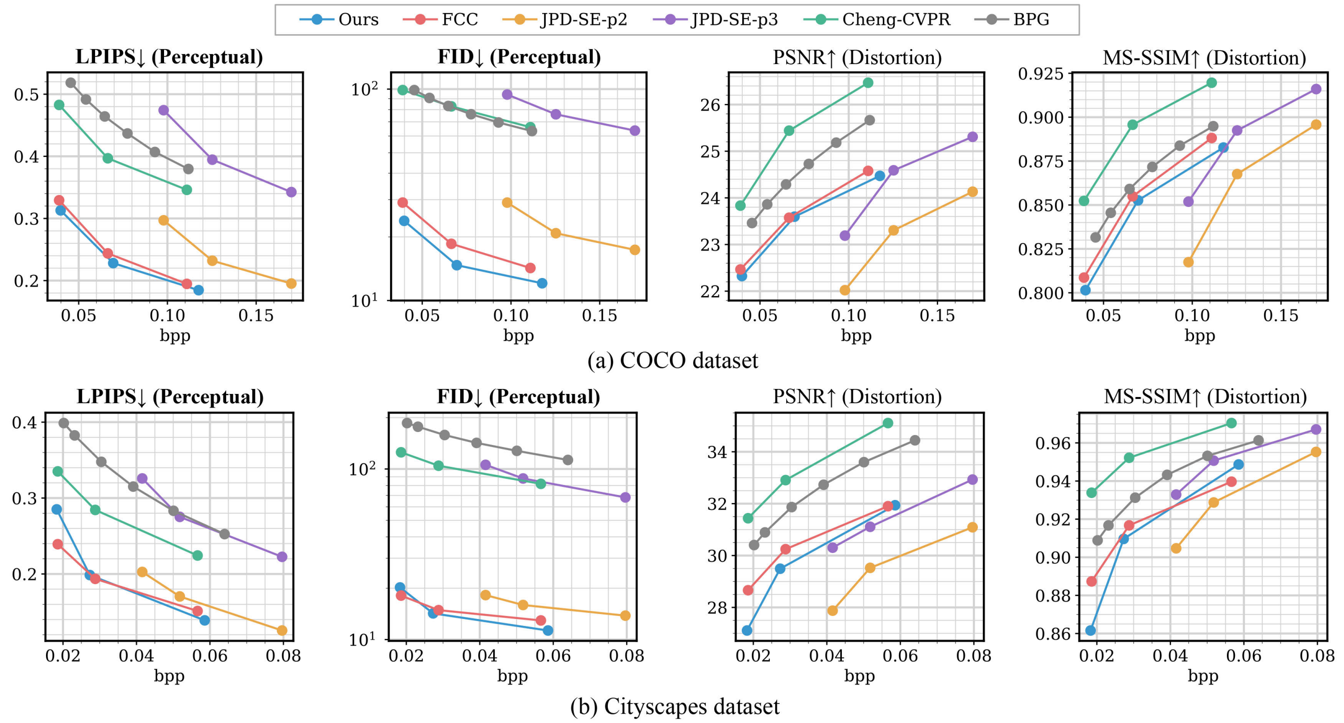
4) EVALUATION METRICS

We evaluated compression methods using Learned Perceptual Image Patch Similarity (**LPIPS**) [65], Fréchet Inception Distance (**FID**) [66], Peak Signal-to-Noise Ratio (**PSNR**), Multi-Scale Structural Similarity Index Measure (**MS-SSIM**), and mean Intersection over Union (**mIoU**). PSNR and MS-SSIM are distortion metrics that have been shown to be inconsistent with human perceptual quality [67], [68]. Consequently, our primary focus lies on the perceptual metrics, LPIPS and FID, which better align with human visual perception. Additionally, we employ mIoU to assess the consistency between the output images and their semantic classes. Specifically, we perform semantic segmentation on the reconstructed images from each compression model using a pre-trained DeepLab v3 [59] and calculate the mIoU score between the predicted label maps and the ground truth label maps. A higher mIoU score indicates that the output images contain proper textures that enable the pre-trained segmentation model to accurately identify the semantic classes. On the COCO dataset, we calculated mIoU for both selected nine and all 183 classes, whereas we calculated mIoU across all 34 classes on Cityscapes. Importantly, the DeepLab V3 model was trained using only original images; reconstructions were not included in the training data.

B. QUALITATIVE RESULTS

1) COCO

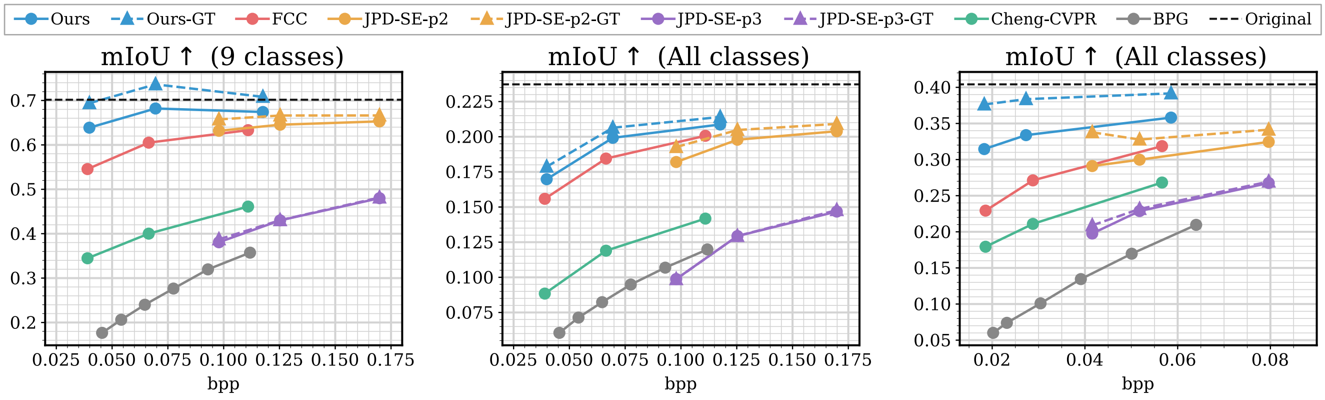
Fig. 5 shows the original image, semantic label map, along with the reconstructions and corresponding bitrates for *Ours*



(a) COCO dataset

(b) Cityscapes dataset

FIGURE 8. Quantitative results on (a) COCO and (b) Cityscapes dataset. Our study primarily focuses on perceptual metrics, LPIPS and FID. PSNR and MS-SSIM serve as distortion metrics, which are less indicative of human perception.



(a) COCO dataset

(b) Cityscapes dataset

FIGURE 9. Segmentation results on reconstructions of each model. "-GT" represents that the ground truth label maps were used instead of predicted label maps in the decoding process. "Original" represents mIoU scores of segmentation on the original (real) images.

and the baseline methods. We also report the bitrates of label maps for semantically guided methods, *i.e.*, *Ours*, *JPD-SE-p2*, and *JPD-SE-p3*.

As shown in Fig. 5, the reconstructions of BPG, *Cheng-CVPR*, and *JPD-SE-p3* appear blurred. BPG also suffers from noticeable block noise. These results demonstrate that the non-GAN-based approaches fail to reconstruct detailed textures. The reconstructions of *FCC* and *JPD-SE-p2* are not blurry; however, their textures often appear unnatural due to artifacts. For example, the *FCC*'s reconstructions in the second row of the first and second samples (the surfer and cat images) have artifacts. Despite *JPD-SE-p2*'s use of semantic information, it struggles with accurately reconstructing complex textures, such as the splashing water and cat fur in the first and second samples, respectively. Furthermore, the high bitrates of the label maps in *JPD-SE-p2* result in higher total bitrates than those of *Ours*. By contrast, the reconstructions produced by our method have appropriate textures corresponding to their semantic classes without blurriness and noise. Although the original

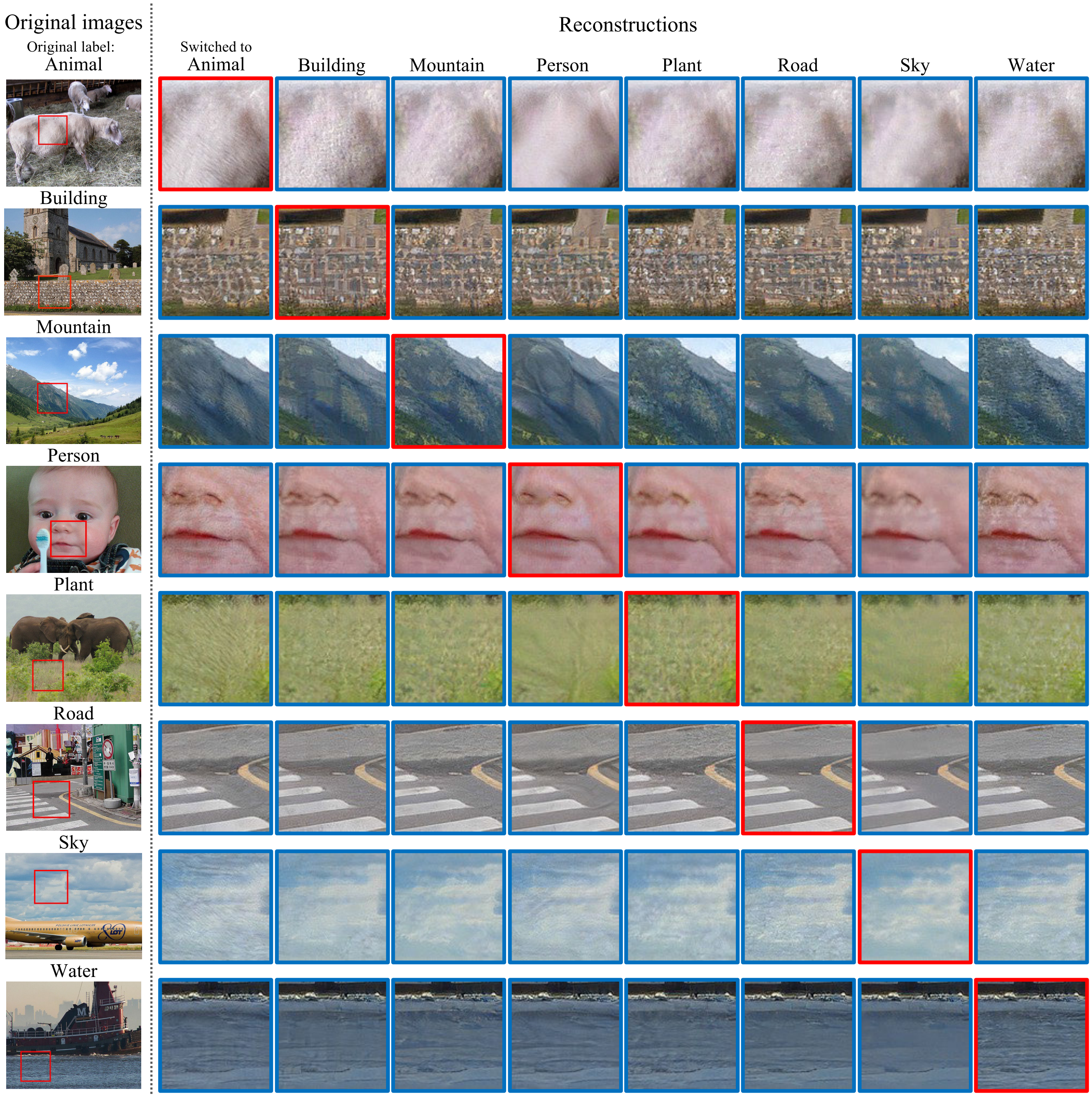


FIGURE 10. **Reconstructions of the proposed method using different label maps for decoding. The red boxes indicate that the original class is used in decoding, while the blue boxes are reconstructed with different classes.**

images are not reconstructed perfectly in this extremely low bitrate setting, the proposed approach can synthesize realistic textures lost through compression, leading to more visually pleasing reconstructions than baselines.

2) CITYSCAPES

Fig. 6 shows the results of the Cityscapes dataset. As with the results of the COCO dataset, *BPG*, *Cheng-CVPR*, and *JPD-SE-p3* suffer from blur. The reconstructions of *FCC* are not blurry but do contain noise and artifacts. For example, the tree in the first image and the road in the second image contain noticeable artifacts. Meanwhile, despite using full-resolution label maps, *JPD-SE-p2* still fails to accurately maintain the distinct boundaries, such as between the sign and the tree in the first image. On the contrary, our method reconstructs natural textures and content for both samples.

These results on two datasets demonstrate that the proposed method effectively utilized semantic information in both domain-agnostic (COCO) and domain-specific (Cityscapes) settings, leading to high-quality reconstructions even at low bitrates.

C. USER STUDY

To compare the subjective quality of our method and existing methods, we conducted a user study involving 20 participants. We showed users the original image and two compressed images, a reconstruction of the proposed method and one of the baseline methods. Then, the users were asked to choose which reconstruction was preferable. We used 20 pairs of randomly selected reconstructions from each of the COCO and Cityscapes datasets.

Fig. 7 reports the results, where the numbers in the bars represent the user preference rates of our method and their 95% confidence interval. As shown in the figure, our method outperformed other methods. In particular, our method was preferred to *BPG*, *Cheng-CVPR*, and *JPD-SE-p3* due to their blurred reconstructions. Moreover, our method outperformed *FCC* on both datasets. It validates the effectiveness of explicit semantic guidance in enhancing perceptual quality by generating textures that align more closely with their semantic contexts. Finally, compared to *JPD-SE-p2*, our method was preferred even though *JPD-SE-p2* has larger data sizes (0.069 bpp vs. 0.098 bpp on COCO and 0.027 vs. 0.042 bpp on Cityscapes). By saving data size on label maps with our strategies, the model can allocate more bitrate to latent information, enhancing the detail and fidelity of the reconstructed images. Consequently, our model achieves superior reconstruction quality compared to *JPD-SE-p2*, resulting in a high preference rate.

D. QUANTITATIVE RESULTS

Fig. 8 (a) and (b) show the quantitative results on the COCO and Cityscapes datasets, respectively. For the semantically guided methods, *Ours*, *JPD-SE-p2*, and *-p3*, the average bitrate includes that of label maps.

RATE-DISTORTION-PERCEPTION PERFORMANCE

Fig. 8 (a) shows the quantitative results in different bitrates on the COCO dataset. *Ours* outperforms other methods on the perceptual metrics LPIPS and FID, which indicates that our method reconstructed more realistic images than baselines. Though *Cheng-CVPR* achieved the highest PSNR and MS-SSIM, these metrics are known to be inconsistent with human perceptual quality [67], [68]. Meanwhile, the performances of *JPD-SE-p2, -p3* are limited due to the large data size required by using full-resolution label maps. This indicates the effectiveness of our label map compression strategy, which significantly reduced the data size of the label maps, leading to higher compression performance.

Fig. 8 (b) illustrates the results on Cityscapes dataset. Except for the lowest bitrate, our method outperformed other methods in terms of FID and was competitive in LPIPS and MS-SSIM against *FCC*. Since FID measures the realism of images rather than reconstruction accuracy, these results indicate that the proposed method synthesized realistic textures. Similar to COCO, the full-resolution label maps led to the limited performance of *JPD-SE-p2, -p3*.

TABLE 2. **Results on lossless label map compression with different configurations on COCO dataset. Bold in the "Configuration" columns indicates that our strategy is used.**

	Configurations			Lossless bitrate
	Size	#classes	Coding method	(bpp↓)
	Full	183	PNG	2.16×10^{-1}
	1/16	183	PNG	9.43×10^{-3}
	Full	**9**	PNG	1.44×10^{-1}
	1/16	**9**	PNG	6.73×10^{-3}
	Full	183	**Autoregressive**	5.89×10^{-2}
	1/16	183	**Autoregressive**	3.05×10^{-3}
	Full	**9**	**Autoregressive**	2.60×10^{-2}
Ours	**1/16**	**9**	**Autoregressive**	$\mathbf{1.30 \times 10^{-3}}$

TABLE 3. **Quantitative comparison between the proposed method and *Full label map model*, a baseline using full-resolution label map, on COCO dataset. The "input label map" column indicates whether predicted or ground-truth label maps are used as input.**

	Input label map	9 classes mIoU ↑	All classes mIoU ↑
Full label map model	Predicted	0.680	0.194
	GT	0.740	0.204
Ours	Predicted	0.682	0.199
	GT	0.736	0.206

EVALUATION WITH PRE-TRAINED SEGMENTATION MODEL

Fig.9 (a) and (b) show the segmentation scores, mIoU, using the pre-trained semantic segmentation model on the reconstructions of each method on the COCO and Cityscapes datasets, respectively. Higher scores indicate that the reconstructed textures more accurately correspond to the actual semantic classes. The notation *"-GT"* in Fig. 9 (e.g., *Ours-GT*) signifies the use of ground truth label maps instead of predicted ones for decoding. This allows us to directly assess the correspondence between the input label map and the output texture, independent of prediction accuracy of the input label map.

As shown in Fig. 9, itOurs achieved the highest mIoU on both datasets, validating that the proposed method successfully added textures corresponding to the label maps. Moreover, even though our method used the selected nine classes segmentation maps, it achieved higher mIoU on COCO (all classes) than *JPD-SE-p2*. This suggests that our method synthesized appropriate textures across a broad range of classes, including those categorized as *others*. Furthermore, using ground-truth label maps as input improved mIoU in both datasets. Since the mIoU score is calculated between the predicted label maps of the reconstructions and ground-truth ones, the improvement indicates that the proposed method accurately reflects the input semantic information in reconstructions.

TABLE 4. Results of ablation study on COCO dataset. Among the baseline models and *Ours*, the best results are highlighted in bold.

	bpp ↓	PSNR ↑	LPIPS ↓	FID ↓	mIoU ↑ 9 class	 All classes
w/o GAN	0.0695	**25.46**	0.391	78.92	0.446	0.127
w/o VGG	0.0695	23.86	0.274	29.24	0.605	0.180
Blank label map	0.0707	23.29	0.245	18.67	0.569	0.168
Ours	0.0695	23.60	**0.228**	**14.70**	**0.682**	**0.199**

TABLE 5. Comparison of the number of parameters and average runtime between our method and existing methods. The runtimes for the encoding, decoding, and total processes on GPU and CPU for 256 × 256 patches on COCO dataset are displayed. *In JPD-SE [11], the runtime for encoding and decoding the label map is excluded because the implementation of its label map compression is unavailable.

	Parameter Count	Runtime on GPU [ms]			Runtime on CPU [ms]		
		Enc.	Dec.	Total	Enc.	Dec.	Total
FCC [8]	21.0M	207.4	247.5	455.0	290.9	354.4	645.3
JPD-SE [11]*	204.1M	232.3	271.6	503.9	373.4	972.7	1346.2
Ours	39.3M	243.4	283.1	526.5	352.3	508.8	861.1

E. ANALYZING THE EFFECTS OF SEMANTIC INFORMATION

To assess the impact of semantic information on texture synthesis in output images, we conducted controlled experiments on the COCO dataset by manipulating the input semantic label map. Specifically, we used label maps where all pixels were assigned to a single class and systematically varied this class to generate different reconstructions. We changed the semantic class and obtained eight distinct reconstructions (*i.e.*, all classes except the "others" class) for each sample.

The results are shown in Fig. 10. For example, while the actual label of the image in the first row is *animal*, the reconstructions with other class labels, such as *building*, *mountain*, and *person*, are also displayed. In Fig. 10, we observed clear distinctions in texture characteristics corresponding to each input class. For example, in the top row, given the input *animal* class, a fine hair-like texture is generated. Conversely, when using the input class *sky* instead of *animal*, the output becomes blurry like a cloud. Moreover, in the eighth row, while wave-like patterns emerge with the label *water*, this pattern does not appear when other labels are used as the input. These results demonstrate that our model learned the characteristics of each semantic class and synthesized appropriate textures according to the input label map. Additionally, these results show why our method achieved high mIoU scores in Sec. IV-D, which measures the alignment of the textures and actual semantic classes.

F. EVALUATION ON LABEL MAP COMPRESSION

To evaluate the effect of each of our label map compression strategies, we performed lossless compression on label maps with different numbers of classes (our reduced set of nine classes and all the 183 classes), varying resolutions of the label maps ($\frac{1}{16}$ down-scaled or full-resolution), and different coding methods (our autoregressive compressor or PNG), obtaining $2^3 = 8$ results. Table 2 compares the average bitrates on the COCO dataset across all settings. These results illustrate that all of our strategies, downscaling the label map, reducing the number of classes, and using the autoregressive compressor, contribute to reducing the data size of the label maps. Notably, by applying all three of our strategies (as shown in the "Ours" column in the table), the average bitrate is reduced to 0.0013 bpp, which is just 0.6% of the average bitrate when none of our strategies are used. Furthermore, given that the lowest average bitrate of our method on the COCO dataset is 0.040 bpp, as shown in Fig. 8(a), the label map occupies only 3.25% of the total data size on average. This demonstrates that our label map compression strategy effectively reduces the data size overhead introduced by semantic information.

G. EFFECTS OF DOWNSCALED LABEL MAP

To investigate the influence of downscaled label maps on compression performance, we compared our model with a baseline model that is trained with full-resolution label maps. This baseline, named *"Full label map model"*, has nearly the same architecture as ours, but it does not include an up-sampling block for the label map shown in Fig. 4. The *Full label map model* is trained so that the data size without the label map is approximately the same as ours. Thus, the *Full label map model* has a higher total bitrate than our approach because the data sizes of the full-size label maps are larger than those of downscaled ones. Table 3 shows the segmentation performance on the reconstructions produced by our method and the *Full label map model*, using DeepLab V3 as the segmentation model. "Predicted" and "GT" in the table represent that we used the predicted label maps and ground truth label maps as inputs, respectively. The results show that despite using a down-scaled label map, the proposed method's segmentation scores were nearly the same as those of the *Full label map model*. This might be attributed to two primary factors. First, the latent code $\hat{\boldsymbol{y}}$ contains the information of the boundary between objects. Hence, even if the boundary information of the label map is lost through down-sampling, the decoder can reconstruct the image correctly. Second, our extra up-sampling blocks learn to expand the label map appropriately, compensating for detailed information of the down-scaled label maps.

H. ABLATION STUDY

To evaluate the effectiveness of our image compression method, we conducted ablation studies. We compared our proposed model with the following three baselines:

- **w/o GAN**: in this configuration, the compression model is trained without the second stage of the training.
- **w/o VGG loss**: in this configuration, we set the weight of the VGG perceptual loss λ_{vgg} to 0 in Eq. (14).
- **Blank label map**: in this configuration, we feed the blank semantic label map (i.e., all pixels are labeled as "others") into the SPADE layers during both training

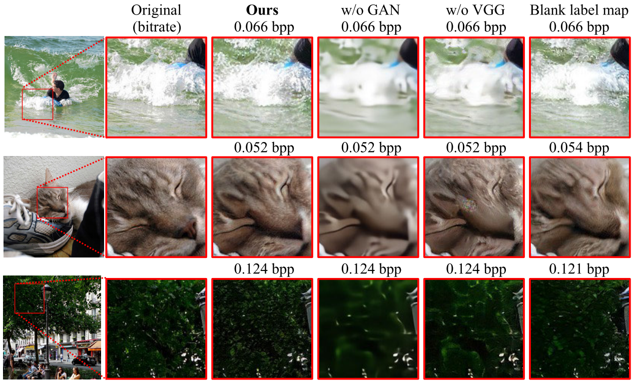


FIGURE 11. **Qualitative comparison between our model and baseline models in the ablation study.**

and inference. In other words, while the model has the same architecture as ours, it does not receive semantic information. This setup aims to confirm that our method's performance gains are not merely due to the extra parameters introduced by the SPADE layer.

The quantitative and qualitative results of the above configurations are summarized in Table 4 and Fig. 11, respectively. Table 4 shows that our method achieves the best performance except on PSNR, which does not correlate to the perceptual quality. While the *w/o GAN* achieves high PSNR, it lags behind other configurations in perceptual metrics like FID and LPIPS, highlighting its compromised perceptual quality. Indeed, Fig. 11 shows that the reconstructions of *w/o GAN* appear blurred. *w/o VGG* suffers from artifacts as shown in the second row in Fig. 11, resulting in inferior LPIPS and FID in Table 4. These results confirm the necessity of the VGG loss in our model. Compared to the *Blank label map*, *Ours* achieved better FID and LPIPS as well as higher mIoU. It indicates that the semantic information helped the model to reconstruct not only semantically accurate textures (measured by mIoU) but also high-quality images (measured by FID and LPIPS). In Fig. 11, the reconstructions of *Blank label map* are not blurred; however, those of *Ours* have more detailed textures reflecting the semantic category.

I. COMPUTATIONAL COMPLEXITY ANALYSIS

We evaluated the computational complexity of our method compared to existing methods. Table 5 presents the number of parameters and runtimes on both CPU and GPU. For runtime evaluation, we randomly selected 100 images from the COCO validation dataset and applied center-crop to obtain 256 × 256 pixel patches. We then calculated the average runtime. The experiments were conducted on a machine equipped with an NVIDIA GeForce RTX 3080 GPU with CUDA version 11.6, an AMD Ryzen 7 3700X 8-Core Processor, and running Ubuntu 18.04.6 LTS. Due to the unavailability of the label map compression strategy implementation in JPD-SE [11], we could not measure its actual runtime. Consequently, the table excludes the runtime of the label map compression process for JPD-SE. The runtimes for our method include both the encoding and decoding of label maps.

Regarding the number of parameters, Table 5 shows that our model has more parameters than FCC [8] due to the additional SPADE layers [54]. However, as demonstrated in Sec. IV-H, the performance improvement does not merely derive from the increased parameters. This was confirmed by comparing our method with the "Blank label map" baseline. Meanwhile, JPD-SE [11] has about five times as many parameters as our method due to its large decoder, indicating that our method incorporates semantic information with fewer parameters.

In terms of GPU runtime, our model was the slowest among the three due to the autoregressive label map compressor. Nevertheless, the difference between our method and JPD-SE [11] is only about 20 ms. This is because using a down-scaled label map significantly reduced the number of processes from $H \times W$ to $H/16 \times W/16$, resulting in a 99.6% cost reduction. Moreover, on the CPU, JPD-SE [11] was significantly slower than our method by a margin of 485 ms. This is because its Pix2PixHD-based [52] post-processing in the decoder demands heavy computation, resulting in slow decoding without GPU acceleration.

V. CONCLUSION

In this study, we have proposed an image compression method that explicitly uses semantic information through semantic label maps. By leveraging semantic information, the proposed method can reconstruct images with proper textures even at low bitrates. Additionally, we introduced three simple yet effective compression strategies. We verified that each strategy contributed to reducing the data size, leading to an average bitrate of the label maps of 0.001 bpp on COCO dataset. Despite the small data size, our experimental results demonstrate that semantic information effectively enhances reconstruction quality, improving overall compression performance. Moreover, compared to existing GAN-based image compression methods, including the state-of-the-art semantically guided method, our method achieves superior performance in both quantitative evaluations and user studies. Furthermore, we conducted experiments to analyze the effect of semantic information by switching the input semantic labels. The results show that our model adaptively synthesized proper textures corresponding to the input label map.

Our current limitation is the manual selection of nine classes, which, while effective, may not be optimal. Future work will explore algorithms for class selection to refine and enhance our compression method.

REFERENCES

[1] F. Bellard. (2014). *BPG Image Format*. [Online]. Available: https://bellard.org/bpg/

[2] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[3] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[4] J. Ball, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[5] D. Minnen et al., "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 10794–10803.

[6] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[8] S. Iwai, T. Miyazaki, Y. Sugaya, and S. Omachi, "Fidelity-controllable extreme image compression with generative adversarial networks," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 8235–8242.

[9] R. Wang, Z. Sun, and S.-I. Kamata, "Adaptive image compression using GAN based semantic-perceptual residual compensation," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 9030–9037.

[10] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 221–231.

[11] S. Duan, H. Chen, and J. Gu, "JPD-SE: High-level semantics for joint perception-distortion enhancement in image compression," *IEEE Trans. Image Process.*, vol. 31, pp. 4405–4416, 2022.

[12] J. Chang, Z. Zhao, L. Yang, C. Jia, J. Zhang, and S. Ma, "Thousand to one: Semantic prior modeling for conceptual coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[13] M. Akbari, J. Liang, and J. Han, "DSSLIC: Deep semantic segmentation-based layered image compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2042–2046.

[14] T. Lin, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[15] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[16] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[17] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5435–5443.

[18] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[19] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1683–1698, Jun. 2020.

[20] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4385–4393.

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

[22] Y. Zhu, Y. Yang, and T. Cohen, "Transformer-based transform coding," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=IDwN6xjHnK8

[23] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17471–17480.

[24] J. Liu et al., "Learned image compression with mixed transformer-CNN architectures," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14388–14397.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[26] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5708–5717.

[27] F. Mentzer, E. Agustsson, and M. Tschannen, "M2T: Masking transformers twice for faster decoding," 2023, *arXiv:2304.07313*.

[28] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Aug. 2017, pp. 2922–2930.

[29] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11913–11924.

[30] S. Iwai, T. Miyazaki, Y. Sugaya, and S. Omachi, "Self texture transfer networks for low bitrate image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1901–1905.

[31] E. Agustsson et al., "Multi-realism image compression with a conditional generator," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22324–22333.

[32] S. Iwai, T. Miyazaki, and S. Omachi, "Controlling rate, distortion, and realism: Towards a single comprehensive neural image compression model," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 2900–2909.

[33] W. Jiang, W. Wang, and Y. Chen, "Neural image compression using masked sparse visual representation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 4177–4185.

[34] M. J. Muckley, A. El-Nouby, K. Ullrich, H. Jegou, and J. Verbeek, "Improving statistical fidelity for neural image compression with implicit local likelihood models," in *Proc. 40th Int. Conf. Mach. Learn. (ICML)*, vol. 202, 2023, pp. 25426–25443.

[35] A. El-Nouby, M. J. Muckley, K. Ullrich, I. Laptev, J. Verbeek, and H. Jegou, "Image compression with product quantized masked image modeling," *Trans. Mach. Learn. Res.*, Dec. 2023. [Online]. Available: https://jmlr.org/tmlr/papers/

[36] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12868–12878.

[37] J. Ho et al., "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 6840–6851.

[38] E. Hoogeboom, E. Agustsson, F. Mentzer, L. Versari, G. Toderici, and L. Theis, "High-fidelity image compression with score-based generative models," 2023, *arXiv:2305.18231*.

[39] N. F. Ghouse, J. Petersen, A. Wiggers, T. Xu, and G. Sautière, "A residual diffusion model for high perceptual quality codec augmentation," 2023, *arXiv:2301.05489*.

[40] R. Yang and S. Mandt, "Lossy image compression with conditional diffusion models," 2022, *arXiv:2209.06950*.

[41] M. Careil, M. J. Muckley, J. Verbeek, and S. Lathuilière, "Towards image compression with perfect realism at ultra-low bitrates," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[42] T. Xu, Z. Zhu, D. He, Y. Li, L. Guo, Y. Wang, Z. Wang, H. Qin, Y. Wang, J. Liu, and Y.-Q. Zhang, "Idempotence and perceptual image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[43] N. Patwa, N. Ahuja, S. Somayazulu, O. Tickoo, S. Varadarajan, and S. Koolagudi, "Semantic-preserving image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1281–1285.

[44] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, and E. Rahtu, "Image coding for machines: An end-to-end learned approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 1590–1594.

[45] S. Sun, T. He, and Z. Chen, "Semantic structured image coding framework for multiple intelligent applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3631–3642, Sep. 2021.

[46] N. Yan, C. Gao, D. Liu, H. Li, L. Li, and F. Wu, "SSSIC: Semantics-to-signal scalable image coding with learned structural representations," *IEEE Trans. Image Process.*, vol. 30, pp. 8939–8954, 2021.

[47] Y.-H. Chen, Y.-C. Weng, C.-H. Kao, C. Chien, W.-C. Chiu, and W.-H. Peng, "TransTIC: Transferring transformer-based image compression from human perception to machine perception," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 23240–23250.

[48] M. Jia, "Visual prompt tuning," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2022, pp. 709–727.

[49] J. Chang, Z. Zhao, C. Jia, S. Wang, L. Yang, Q. Mao, J. Zhang, and S. Ma, "Conceptual compression via deep structure and texture synthesis," *IEEE Trans. Image Process.*, vol. 31, pp. 2809–2823, 2022.

[50] J. Chang et al., "Consistency-contrast learning for conceptual coding," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2681–2690.

[51] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[52] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[53] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[54] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2332–2341.

[55] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "SEAN: Image synthesis with semantic region-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5103–5112.

[56] E. Schönfeld, V. Sushko, D. Zhang, J. Gall, B. Schiele, and A. Khoreva, "You only need adversarial supervision for semantic image synthesis," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[57] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.

[58] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 606–615.

[59] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 833–851.

[60] A. Van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4797–4805.

[61] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4105–4113.

[62] J. Lee, D. Kim, Y. Kim, H. Kwon, J. Kim, and T. Lee, "A training method for image compression networks to improve perceptual quality of reconstructions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 585–589.

[63] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 694–711.

[64] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2813–2821.

[65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[66] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 1–12.

[67] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6228–6237.

[68] Y. Blau and T. Michaeli, "Rethinking lossy compression: The rate-distortion-perception tradeoff," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 675–685.

SHOMA IWAI (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from Tohoku University, Japan, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree in communication engineering with the IIC-Laboratory. His current research interests include computer vision and image compression.

TOMO MIYAZAKI (Member, IEEE) received the B.E. degree from Yamagata University, in 2006, and the Ph.D. degree Tohoku University, in 2011. He worked on a Geographic Information System, Hitachi Ltd., until 2013. From 2013 to 2023, he was a Postdoctoral Researcher and an Assistant Professor with Tohoku University. Since 2024, he has been an Associate Professor. His research interests include pattern recognition and image processing.

SHINICHIRO OMACHI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in information engineering from Tohoku University, Japan, in 1988, 1990, and 1993, respectively. He was an Assistant Professor with the Education Center for Information Processing, Tohoku University, from 1993 to 1996. Since 1996, he has been affiliated with the Graduate School of Engineering, Tohoku University, where he is currently a Professor. From 2000 to 2001, he was a Visiting Associate Professor with Brown University. His current research interests include pattern recognition, computer vision, image processing, image coding, and parallel processing. He is a member of the Institute of Electronics, Information and Communication Engineers, and the Information Processing Society of Japan. He received the IAPR/ICDAR Best Paper Award, in 2007; the Best Paper Method Award of the 33rd Annual Conference of the GfKl, in 2010; the ICFHR Best Paper Award, in 2010; and the IEICE Best Paper Award, in 2012. He served as the Vice Chair for the IEEE Sendai Section, from 2020 to 2021. He served as the Editor-in-Chief for *IEICE Transactions on Information and Systems*, from 2013 to 2015.