Region-Native Visual Tokenization

Mengyu Wang[1,5(†)], Yuyao Huang[1,5(†,‡)], Henghui Ding[4], Xinlong Wang[2], Tiejun Huang[2], Yao Zhao[1,5,6], Yunchao Wei[1,5,6(✉)], and Shuicheng Yan[3]

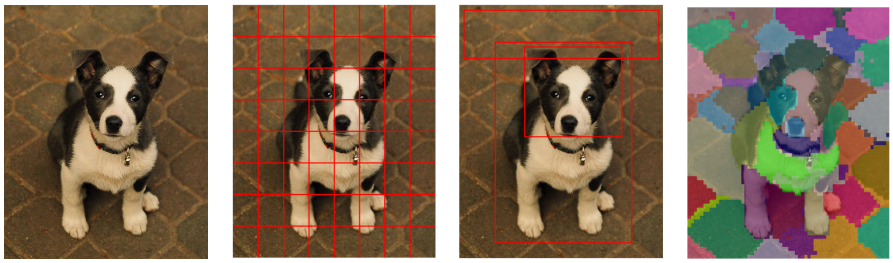
[1] Institute of Information Science, Beijing Jiaotong University, Beijing, China
[2] Beijing Academy of Artificial Intelligence
[3] Skywork AI
[4] Institute of Big Data, Fudan University, Shanghai, China
[5] Visual Intelligence + X International Joint Laboratory of the Ministry of Education
[6] PengCheng Laboratory, Shenzhen, China

Abstract. We explore an innovative region-based visual token representation and present the **RE**gion-native **A**utoenco**DER** (`Reader`). In contrast to the majority of previous methods, which represent each image as a grid-shaped tokens map, `Reader` perceives each image into sequential region-based tokens, with each token corresponding to an object or one part of an object in the image. Specifically, `Reader` comprises both an encoder and a decoder. The encoder can partition each image into an adaptive number of arbitrary-shaped regions and encode each region into a token. Subsequently, the decoder utilizes this adaptive-length token sequence to reconstruct the original image. Experimental results demonstrate that such region-based token representation possesses two main notable characteristics. Firstly, it achieves highly efficient image encoding. `Reader` can adaptively use more regions to represent complex areas and fewer regions in simpler ones, thus avoiding information redundancy. Consequently, it achieves superior reconstruction fidelity compared to previous methods, despite using significantly fewer tokens for each image. Secondly, the region-based manner enables manipulation on a local region without causing global changes. As a result, `Reader` inherently supports diverse image editing operations, including erasing, adding, replacing, and modifying shapes on the objects, and achieves excellent performance in the image editing benchmark of smile transferring. Code is provided at https://github.com/MengyuWang826/Reader.git.

1 Introduction

Tokenization, originating from Natural Language Processing (NLP), involves breaking a textual data stream into several elements called tokens. It plays an important role for converting raw data into a form that is predictable and analyzable by deep learning systems [13, 42]. With the gradual maturity of tokenization technique in the NLP field, it has also been adapted to the Computer

† Internship at BAAI.
‡ Internship at Skywork AI.
✉ Corresponding author.



(a) Image **(b)** Grid-based **(c)** Object-based **(d)** Region-based

Fig. 1: Illustration of different types of visual tokenization methods. Grid-based methods [16,56] represent images as grid-shaped tokens map. Object-based methods [1, 59] use detection models to represent objects as tokens. While our region-based method represents arbitrarily shaped regions as tokens, which offers more flexibility and retains the naturally existed boundaries presented in images.

Vision (CV) domain, known as visual tokenization, and has become the foundation for many tasks such as image/video generation [16, 27, 56] and multimodal generation/perception [11, 14, 18, 38].

Existing visual tokenization methods can be categorized into two main types, which we refer to as grid-based and object-based. **Grid-based** methods typically employ a CNN [23] or ViT [15] backbone to encode the original image into a downsampled grid-shaped feature map, followed by quantization of the feature map to get the image tokens. However, in such methods, each token corresponds to a patch in the original image (see Fig. 1b), which does not adhere to the following characteristics of the image prior. Firstly, different areas in images have varying information densities. While in these methods, patches are treated equally, leading to information loss in densely informative patches and redundancy in sparsely informative ones. Secondly, areas with different semantics within an image have natural boundaries. While employing patches as the units of image tokens disregards the natural boundaries, causing local information within the original image to be distributed and coupled across multiple tokens, limiting the extraction and utilization of local features. **Object-based** methods [1, 35, 51, 53, 59], as an alternative and more natural approach, typically appear in vision-language pretraining. These methods employ detection or segmentation models (see Fig. 1c) to encode the objects within images into tokens, which are then interacted with or aligned with language information. However, these methods only focus on extracting high-level semantic information, neglecting low-level visual details in the image. Consequently, they cannot be used for tasks that require image outputs, such as image synthesis and editing.

These challenges suggest that, an ideal visual token representation should take into account the crucial visual-prior characteristics, including the uneven information density and the inherent semantic boundaries, and further achieve local-information decoupling and global-information compression. However, previous practices either ignore the visual-prior characteristics or, due to their coarse granularity, result in significant information loss. To address these problems, we



Fig. 2: Examples of the proposed `Reader` performing various editing operations. `Reader` showcases its strengths in flexibly editing multiple attributes of existing real images under several interactive operations (see Sec. 3.4). In each pair, the left displays the original image, the right displays the edited result and the text describes the corresponding editing objective.

propose the **RE**gion-native **A**utoenco**DER** (`Reader`), which takes arbitrarily shaped and sized regions (see Fig. 1d) as the fundamental units for visual tokens, providing more flexibility and aligning better with the visual-prior characteristics. The `Reader` framework conceptualizes each image as a composite of several regions, with each region corresponding to either a small-sized object or a portion of a regular or large-sized object.

Specifically, the `Reader` architecture comprises an encoder and a decoder, as illustrated in Fig. 3. The encoder first partitions an image into an adaptive-number of non-overlapping regions, where each region is represented as a binary mask and an embedding. Following this, the encoder employs a novel region embedder to integrate the geometric information from the mask with the visual features from the embedding, and produce the desired token for each region. Subsequently, a novel cross-attention-based decoder is employed to reconstruct the original image from these tokens.

We further conduct both qualitative and quantitative experiments and demonstrate the following two main notable characteristics of `Reader`:

Efficiency: `Reader` achieves highly efficient encoding, yielding reconstruction fidelity superior to that of previous methods, despite employing a significantly smaller number of tokens. This is intuitive since the region-based manner enables `Reader` to naturally represent complex areas with more regions, while representing simpler areas with fewer regions. This effectively avoids information redundancy, thus possessing inherent efficiency.

Disentanglement: Representing non-overlapping regions as tokens guarantees their disentanglement from one another, and further empower `Reader` with the ability to perform image editing. Taking object erasure as an example, the disentanglement permits an object in the image to be represented solely by the tokens corresponding to its interior. Through removing these tokens and then reconstructing the image, object erasing can be naturally achieved. Consequently, without any additional modules or training data, `Reader` facilitates diverse object-level editing operations (see Fig. 2), such as erasing, adding, re-

placing, and modifying shapes on objects, and demonstrates excellent performance in the image editing benchmark of smile transferring [31].

We believe that these experiments effectively demonstrate the inherent advantages of region-based tokenization method. We hope this will bring some novel insights to the community about visual tokenization and unlock its value in various downstream tasks in the future, such as interaction with LLMs and multimodal perception, generation, and beyond.

2 Related Work

Grid-based Tokenization The majority of grid-based methods follows the feature extraction, quantization, and reconstruction paradigm, initially introduced by VQVAE [39]. These methods represent image tokens as a grid-shaped code map. Expanding upon this framework, VQGAN [16] incorporates perceptual loss [58] and GAN loss [28] to augment the original L1 reconstruction loss, with the aim of improving both reconstruction fidelity and codebook learning. RQ-VAE [32] introduces residual quantization to further enhance reconstruction fidelity, while leading to the cost of significantly higher number of tokens. ViT-VQGAN [56] replaces the previously employed CNN-based architecture with a ViT-based one. Sharing a similar motivation to ours, DQ-VAE [27] proposes dynamically selecting patch granularity between two or three pre-defined strides to achieve dynamic-length coding. However, this approach remains constrained by the grid-based paradigm. In contrast, our `Reader` takes arbitrarily shaped regions as image token units and offers greater flexibility.

Object-based Tokenization Methods of this kind typically find applications in vision-language pretraining and multimodal perception. For instance, [1] uses a Faster R-CNN [44] for proposals prediction and encodes the proposals into tokens to subsequently perform image captioning or visual question answering. Similarly, [59] also employs a Faster R-CNN to extract object tokens and maps these tokens with a Graph Neural Network [57]. GroupViT [53] proposes a learnable clustering architecture based on ViT [15] to extract object information, and aligns it with text embedding akin to CLIP [41]. However, these methods focus on extracting semantic information at object-level, leading to excessively coarse-grained and overlooking on detailed visual information. In contrast, our `Reader` employs an autoencoder architecture to extract dynamic-grained local regions as the unit of image tokens, which better preserves detailed information and showcases superior performance in image reconstruction and editing.

Scene Decomposition The scene decomposition task [4,37,55], also referred to as object-centric representation, has the most similar form to our `Reader`. This task aims to decompose a scene into multiple objects and learn the representation of each object. However, these works focus on application in some synthetic datasets containing only basic 3D shapes [20,29,43]. In contrast, our `Reader` can be applied to a wide range of natural images.

Image Editing Most early image editing methods are GAN-based [21]. Some of these approaches [22, 34, 40, 60] rely on the controllability of the GAN's latent space, employing supervised or unsupervised approaches to achieve desired transformations of GAN's latent variable. Other works [8, 31, 52] utilize GANs' conditional generation ability and synthesize the edited images based on the input conditions. The remarkable success of diffusion models [25, 48] in image synthesis has also led to the emergence of numerous diffusion-based editing methods [3,10,19,47,54]. These approaches largely build upon the robust semantic understanding and rendering capabilities of large pre-trained diffusion models [45].

The semantic-map based methods [19, 31, 34] can produce the most similar effects to our `Reader`. However, compared to these works, our `Reader` benefits from its region-based representation, enabling a wider variety of editing operations without requiring additional input, with details elaborated in Sec. 3.4.

3 Methodology

We propose `Reader` (region-native autoencoder) in this work, which performs region-based visual tokenization and enables diverse controllable image editing operations. In the following, we will first provide detailed descriptions about the model architecture of the `Reader`'s encoder and decoder in Sec. 3.1 and Sec. 3.2, respectively. Sec. 3.3 outlines the objective function utilized, while Sec. 3.4 delves into the detailed implementation for image editing. The model architecture is illustrated in Fig. 3.

3.1 Encoder Architecture

The encoder of `Reader` needs the capability to first segment an image into numerous regions and then encode each region into a token. Thus, we propose a encoder with three core components: an image feature extractor, a region decoder and a region embedder.

Feature Extractor To extract information from the original image for subsequent image segmentation, tokenization, and reconstruction steps, we employ a standard ViT [15] along with a simple FPN [33] as our feature extractor. Given the importance of fine-grained details for high-fidelity reconstruction, we configure the patch size of the ViT to be 8×8, a smaller size compared to conventional settings, yet effective in preserving local structure. The strides of the simple FPN are set to $(1, \frac{1}{2})$, thus the output features are at resolutions of $\frac{1}{8}$ and $\frac{1}{4}$ of the original image.

Region Decoder We employ a region decoder to learn the region representation and segment image into regions. Specifically, we employ a standard Transformer decoder [50], which is similar to previous Transformer-based segmentation works [5–7, 36], augmented with an additional bidirectional Transformer block [30].

In the Transformer decoder, N learnable embeddings are used as input queries to extract the region information, termed region queries. Within each

layer of the Transformer decoder, one of the two-sized image features alternately serves as the key and value. Through cross-attention, the region queries can extract region-level information from the image features. Subsequently, the bidirectional Transformer block takes the updated region queries and the larger-sized image feature as input to perform both region-to-image and image-to-region attention and update them. This process yields the region embeddings $\mathcal{E}_r \in \mathbb{R}^{N\times c}$ and a region-aware image feature $\mathcal{E}_{img} \in \mathbb{R}^{c\times h\times w}$. Here, (h, w) represents the resolution of the image feature, which is $\frac{1}{4}$ of the original image size, and c denotes the channel dimension.

We then compute the similarity matrix between the region embeddings and the image feature via a dot product, formulated as:

$$S = \mathcal{E}_r \odot \mathcal{E}_{img} \in \mathcal{R}^{N\times h\times w}. \tag{1}$$

We use this similarity matrix to get the desired regions. To prevent information redundancy caused by overlapping, we apply an `argmax` operation along the first dimension of the similarity matrix. This process means matching each pixel with one embedding that has the highest similarity to the pixel's feature, which can be formulated as:

$$S_{\text{hard}} = \text{one-hot}(\text{argmax}(S)) \in \{0, 1\}^{N\times h\times w}. \tag{2}$$

Consequently, the S_{hard} in Eq. (2) can also be seen as N binary masks, where the foreground of each mask indicates which pixels belong to that specific region. However, not each one among the N masks contains foreground. In fact, in some simple images, numerous pixels may be matched with a single region embedding, while other region embeddings remain unmatched with any pixel (illustrated as "used" and "unused" in Fig. 3). Therefore, while N is a fixed hyper-parameter, the actual number of regions corresponding to each image is dynamic, determined by the complexity of the image.

Region Embedder Through the region decoder, the original image is represented as several regions. Each region comprises a region embedding $\mathcal{E}_r[i]$ representing the visual feature and a binary mask $S_{\text{hard}}[i]$ representing the geometric information (*e.g.*, shape, position and scale). The region embedder is further employed to merge the above two components and produce the desired tokens.

Following GroupViT [53], we first aggregate the feature of each pixel into the corresponding region embedding. To make the entire process differentiable, enabling the gradients of image reconstruction to guide the granularity of regions, we employ the straight-through trick [39] here. This process is formulated as:

$$S = \text{softmax}(S, \text{dim} = 0), \tag{3}$$

$$\hat{S} = S_{\text{hard}} + S - \text{sg} \cdot (S), \tag{4}$$

$$\hat{\mathcal{E}}_r[i] = \mathcal{E}_r[i] + \sum_{j,k} \hat{S}[i, j, k] \cdot \mathcal{E}_{img}[:, j, k], \tag{5}$$

in which the $\text{sg} \cdot ()$ refers to stop gradient. While the pixel features aggregation process shown in Eq. (5) is actually a pooling process, leading to over-smoothing

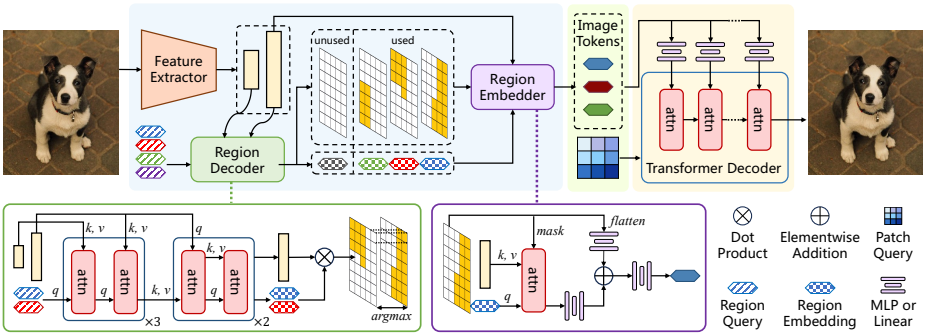


Fig. 3: An overview of the proposed `Reader`. It comprises an encoder and a decoder. The encoder can segment each image into an adaptive number of regions, which is illustrated as the "used" part, and embeds each region into a token with the Region Embedder. The decoder then reconstructs the original image from these tokens with the learnable patch queries.

on local details. We additionally employ a masked cross-attention layer [6] to enhance the visual feature extraction. This layer uses the region embeddings $\hat{\mathcal{E}}_r$ as queries to perform attention to the image feature $\mathcal{E}_{img}$ only in the foreground of each region in order to extract more local details.

Subsequently, we incorporate a module to explicitly encode the geometric information in the masks and integrate it into the region embeddings. This approach serves two primary purposes: Firstly, the inclusion of geometric information enhances the delineation of boundaries between regions, thereby facilitating image reconstruction. Secondly, such explicit encoding also enables more editing operations in subsequent steps. Specifically, we flatten each mask and utilize a linear layer to project it into a latent space. Additionally, the region embeddings are also projected with an MLP and are then fused with the former projection through element-wise addition. Following a normalization and another MLP, the region embedder produces the desired region-based image tokens.

3.2 Decoder Architecture

The encoder of `Reader` represents each image as an adaptive-length sequence. Consequently, the decoder needs the capacity to decode this adaptive-length sequence into a uniformly-sized image. This poses a challenge of mapping variable-length input to a fixed-length output. To address this, the decoder architecture of `Reader` is based on the Transformer decoder, leveraging the cross-attention mechanism to resolve the aforementioned problem.

Specifically, we introduce M learnable embeddings as the input queries for the Transformer decoder. Each embedding is responsible for reconstructing a local patch in the output image and is referred to as a patch query. The queries are also incorporated with the positional embeddings of the center coordinate of the patches. To prevent information loss and enable the Transformer decoder to focus on different information at each layer, we employ a separate MLP for each

layer to project the image tokens. These projections then serve as the key and value for the corresponding layer.

Through the Transformer decoder, the patch queries interact with the image tokens through cross-attention, gathering information related to the corresponding patches and updating their values. After the Transformer decoder, we integrate an linear layer to finally map the output patch queries to the pixel space, which is a reverse process of the patch embedder of ViT.

3.3 Objective Function

The objective functions related to the image reconstruction are kept consistent with VQGAN [16], which is a combination of the L1 loss, perceptual loss [58] and a patch-based GAN loss [28], denoted as $\mathcal{L}_{img}$.

However, in our observation, relying solely on the image reconstruction loss is insufficient for training the encoder to produce semantically meaningful regions. In practice, the regions easily collapse into uniformly distributed patches. Therefore, additional supervision on the output regions becomes necessary. We employ either the pseudo-annotations produced by SAM [30] or manual annotations to supervise the output regions in different experiments, with detailed information provided in Sec. 4.

In order to train the similarity matrix in Eq. (3) to resemble a one-hot form at each pixel, we employ a cross-entropy loss here. Unlike the tasks, where cross-entropy typically used with defined category IDs, there is no fixed correspondence between the predicted regions and the masks of annotations here. Following [5,7], we employ a Hungarian-algorithm based bipartite matching to determine the matching relationship, with more details provided in supplementary material. The overall objective function is formulated as:

$$\mathcal{L} = \mathcal{L}_{img} + \mathcal{L}_{ce}. \tag{6}$$

3.4 Image Editing Implementation

The proposed `Reader` inherently facilitates diverse image editing operations. Without any additional modules, this is achieved by simply manipulating the image tokens outputted by the encoder and then inputting these manipulated tokens into the decoder to get the edited image.

Specifically, before the region embedder of `Reader`, each region is represented by two components: the geometric information (represented as a binary mask) and the visual information (represented as the region embedding). This enables independent manipulations on both two parts and achieves different editing effects. Therefore, we define the following three-level image editing operations. A schematic diagram is shown in Fig. 4 to provide an intuitive understanding.

Geometry Level These operations refer to only manipulating the masks through affine transformations or interactive brushing by users, while maintaining the embeddings unaltered. Geometry-level operations allows to perform moving, scaling, and shape modification on objects. As an example, the left branch in Fig. 4 alters the shape of the lips without changing their color.

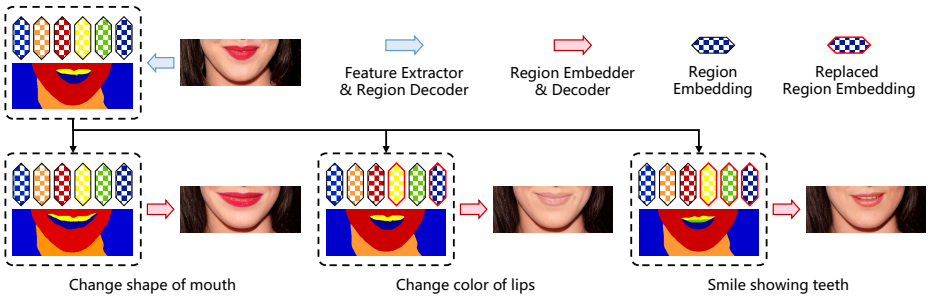


Fig. 4: Illustration of editing operations. Different operations on the embeddings and masks leads to different editing effects. The left, middle and right branches correspond to the three-levels operations described in Sec. 3.4, respectively.

Feature Level Feature-level operations maintain the mask unchanged while replacing the corresponding embedding with one from another object or another image. This facilitates local feature transferring without impacting geometric characteristics. An example is illustrated in the middle branch of Fig. 4, where the color of lips is changed by replacing the corresponding two embeddings.

Token Level The erasure, addition and replacement of objects are categorized as this level. Erasure can be achieved by removing tokens corresponding to a specific object. Conversely, addition can be accomplished by appending tokens from another image to the tokens of the current image. Object replacement can then be achieved by combining both operations. Furthermore, these operations can also be combined with the geometry-level manipulations to achieve addition and replacement on objects while controlling their shapes, scale and position, *etc*. An example is illustrated in the right branch of Fig. 4, where the mouth and lips are replaced by those from another image.

4 Experiments

4.1 Experiment on ImageNet

Herein we train `Reader` on ImageNet training set and report qualitative results in image editing and quantitative comparisons on reconstruction fidelity.

Implementation As outlined in Sec. 3.3, additional supervision is essential to prevent the collapse of producing regions. Consequently, in this experiment, we employ SAM [30] to produce pseudo-annotations on the ImageNet training set. Specifically, we keep the original settings of the *segment anything* mode of SAM, which effectively segments all prominent objects or local parts in images. Since there is substantial overlapping in the original results of SAM, while the *cross-entropy* loss needs non-overlapping annotations, we implement the following two post-processing steps. Firstly, removing all larger masks that encompass other ones, retaining only smaller, more localized masks. Secondly, regions not encompassed by the pseudo annotations do not contribute to the objective function.

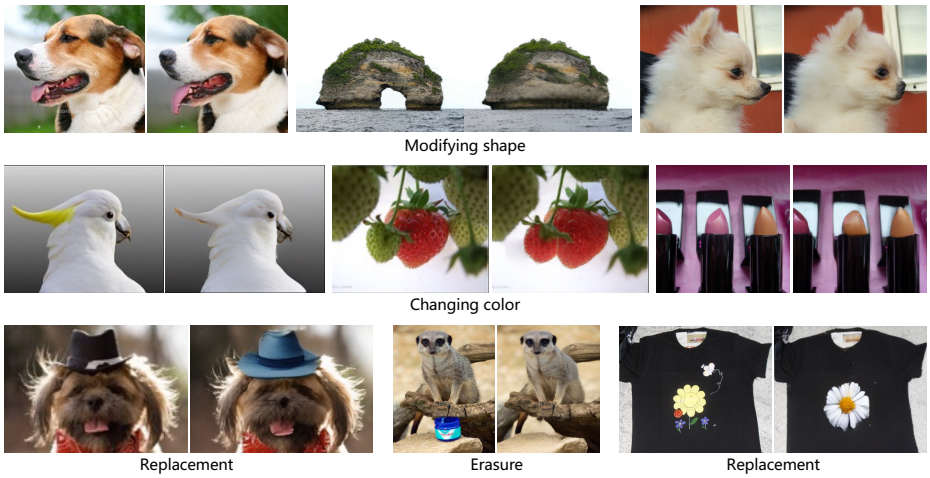


Fig. 5: Qualitative results in ImageNet validation set and Internet images. The editing objective is labeled with text under image.

The input and output sizes of images are set to (256×256) in this experiment. Both the ViT in the encoder and the Transformer in the decoder follow the configuration of ViT-tiny, which utilizes 192 channels and 3 heads in the multi-head attention. The dimension of image tokens is set to 256 to maintain consistency with the previous methods. The number N of learnable region queries in the encoder is set to 256, which is the minimal tokens number of each image in previous works and serves as the maximum regions number of our model. The number M of learnable patch queries in the decoder is set to 1024, with each query responsible for reconstructing a 8×8 patch in the output image. For more implementation details, please refer to our supplementary material.

Results We provide several image editing examples in Fig. 5 corresponding to all the three-level editing operations described in Sec. 3.4. For example, "sticking out the tongue" needs only manipulation on the tongue's mask, "replacing the color" is achieved by replacing the region embedding of the strawberry and "replacing with another hat" needs deleting the tokens of the original hat and replacing them with new ones from another image.

For quantitative comparison of the reconstruction fidelity, we employ both the distribution-level metric rFID [24] and the image-patches-level metrics PSNR, SSIM [26], LPIPS [58] to evaluate the reconstructed images of ImageNet validation set. We employ the VQGAN [16], RQ-VAE [32] and DQ-VAE [32] as baselines. The former two employ fixed tokens number of each image, while the latter uses dynamic tokens number through picking one granularity for each patch from three pre-defined strides $(\frac{1}{8}, \frac{1}{16}, \frac{1}{32})$. The tokens number of DQ-VAE is computed by the ratio of different strides provided in their paper. Tokens number of our `Reader` corresponds to the **average** number observed across the entire ImageNet validation set.

Table 1: Quantitative results for the reconstruction fidelity on the ImageNet validation set. "Tokens" refers to the number of image tokens utilized for each image. The dimension of image tokens is 256 for all methods. The "†" denotes data excerpted from the original paper. Top 1 and top 2 results are highlighted in red and blue, respectively.

Method	Tokens	PSNR ↑	SSIM ↑	LPIPS ↓	rFID ↓	Params(M)	FLOPs(G)
VQGAN(1024)	16×16	19.36	0.48	0.31	8.05	72.14	196.33
VQGAN(16384)	16×16	19.83	0.49	0.28	4.92	76.07	197.33
RQ-VAE(16384)†	$8 \times 8 \times 2$				10.77		
RQ-VAE(16384)	$8 \times 8 \times 4$	20.03	0.50	0.28	5.24	104.40	193.64
DQ-VAE(16384)†	256				4.08 - 7.32		
Reader(ours)	79.09	24.75	0.72	0.20	2.43	26.60	40.24

Results shown in Tab. 1 indicate that, in comparison to the baselines, our `Reader` attains superior reconstruction fidelity despite having the smallest tokens number, model parameters and computational complexity. While compared to DQ-VAE, which also employs dynamic length encoding, our method also demonstrates significant superiority. This finding is intuitive that, compared to both the fixed- and dynamic- sized patches, arbitrarily shaped regions can better handle the uneven information density in images and avoid information redundancy, thus achieving higher reconstruction fidelity with fewer tokens.

4.2 Experiment on Human Face

To further evaluate the editing capabilities of our `Reader`, we transfer it to images of human faces, which are commonly employed in previous image editing works, allowing to provide quantitative comparisons.

Implementation We fine-tune the `Reader` in this experiment, which is pre-trained as outlined in Sec. 4.1. We utilize the CelebAMask-HQ dataset [31], a face parsing dataset with manual annotations. The input and output sizes of images for this experiment are configured as (512×512) to align with the previous methods [31, 34]. In order to preserving the knowledge already learned by the pre-trained model, we keep the majority of `Reader` architecture unchanged and make the following modifications.

(1) Adding a downsampling-and-convolution layer with a residual connection in the region embedder and initializing the new convolution's output to zero. This is employed to make the double-sized masks meet the input length of previous mask projection layer. (2) Increasing the number of learnable patch queries from 1024 to 4096 in order to get double-sized reconstructed images. While loading the pre-trained weights, we perform bicubic interpolation on the weights of the positional embeddings and the patch queries to match the new size.

Results In Fig. 6, we present the qualitative results corresponding to the three-level editing operations. In the top and middle rows, we performed overlay editing, gradually changing one part of the geometric and visual properties at a



Fig. 6: Qualitative results in huaman faces. The editing objective is labeled with text under corresponding image.

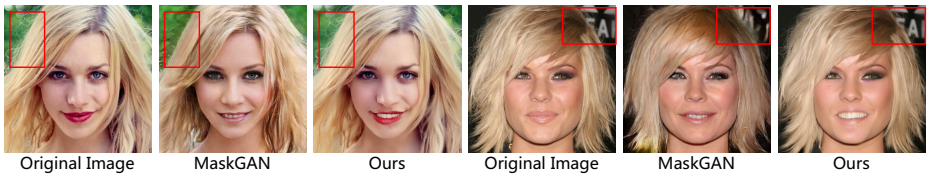


Fig. 7: Qualitative comparisons in the smile transferring benchmark. Our `Reader` can selectively manipulate only the target attributes without affecting other features or regions. The red boxes highlight the local details that should remain unchanged after editing. Please kindly zoom in for a better view.

time. In the bottom row, we simultaneously manipulated both the masks and embeddings, facilitating significant change in expression.

To quantitatively measure `Reader`'s image editing capabilities, we utilize the smile transferring benchmark introduced by MaskGAN [31], which requires converting the faces with neutral expressions into smiling faces. We compare our `Reader` with the MaskGAN [31], LocalEditing [9], InterFaceGAN [46] and EditGAN [34], which also provide results in this benchmark. We employ the following metrics to keep consistent with [34].

(1) Semantic Correctness: Training a binary classifier to recognize the smile attribute and evaluating whether the edited faces exhibit a smiling expression. (2) Distribution-level Image Quality: Utilizing FID [24] and KID [2] to assess the quality of edited images. (3) Identity Preservation: Leveraging a pretrained ArcFace feature extraction network [12] to evaluate whether the face's identity is maintained post-editing.

To automate smile transferring without manual adjustments, we employ a similar approach to MaskGAN. Specifically, we select 400 pairs of neutral-

Table 2: Quantitative results in the smile transferring benchmark. Top 1 and top 2 results are highlighted in red and blue, respectively. Our `Reader` significantly outperforms previous methods in the FID and ID Score metrics by a large margin.

Metric	Attribute Acc. (%)↑	FID↓	KID↓	ID Score↑
MaskGAN [31]	77.3	46.84	0.020	0.4611
LocalEditing [9]	26.0	41.26	0.012	0.5823
LocalEditing - Encoding4Editing [49]	41.8	48.28	0.016	0.7295
InterFaceGAN [46]	83.5	39.42	0.010	0.7295
EditGAN [34]	91.5	41.74	0.013	0.7047
EditGAN^{+30} [34]	85.8	40.83	0.012	0.7452
Reader (ours)	84.2	37.91	0.011	0.8470

expression and smiling-expression faces with the closest head position as the source and target images. Subsequently, we replace both the tokens corresponding to mouth and lips of the source image with those from the target image.

Results shown in Tab. 2 demonstrate that our `Reader` achieves excellent performance across all metrics. Specifically, our `Reader` significantly outperforms previous methods in the FID and ID Score metrics by a large margin. This indicates that the edited images produced by `Reader` exhibit higher quality while better preserving the ID features of the original images. This observation effectively highlights the inherent advantage of our method. By representing the image through a series of decoupled tokens, `Reader` can selectively manipulate only the target attributes without affecting other regions. To provide a more intuitive perception, we illustrate some examples of the original and edited image pairs in Fig. 7. It can be observed that our method effectively prevents changes in the areas which are not the editing target, while achieving editing goals. In contrast, the GAN-based baseline [31] is prone to inducing global changes.

4.3 Ablation Study

In Tab. 3a, we conducted ablation studies to evaluate the effectiveness of some optional hyper-parameters and modules in the model architecture, where different configurations represent: **Patch Size**: The patch size of the ViT in `Reader`'s encoder. **Bidirectional**: Whether to use bidirectional transformer block in the region decoder. **Masked-Attn**: Whether to employ the masked cross-attention layer in the region embedder. Results indicate that a ViT with a smaller patch size paired with the additional modules produces the best image fidelity. This demonstrates that our default settings can better preserve the fine-grained details in the image, thereby enhancing the quality of the output image.

In Tab. 3b, we conducted ablation experiment to determine the granularity of pseudo-labels utilized in training. In this experiment, during preprocessing, we only removed the overlapping parts from the pseudo-labels and retained masks of all different sizes to train our `Reader`. Subsequently, we calculated the average

Table 3: Ablation studies on both model architecture, hyper parameter and data preprocessing. Our default setting is marked with gray.

Patch Size	Bidirectional	Masked-Attn	rFID ↓
	✗	✗	98.39
8×8	✓	✗	37.50
	✓	✓	11.42
16×16	✓	✓	61.64

(a) Module configurations.

Area Proportion	L1 ↓
$< 1/32$	1.44
$1/32 \sim 1/16$	1.56
$1/16 \sim 1/8$	2.53
$> 1/8$	3.46

(b) Annotation configurations.

L1 distance of different-sized regions between the original images and the reconstructed images. The "Area Proportion" in Tab. 3b represents the proportion of the region's area relative to the entire image. The results indicate that when the region's area proportion is less than $\frac{1}{16}$, the L1 distance stabilizes with changes in area. Therefore, in the formal experiments, we only retained pseudo-labels with area proportion smaller than $\frac{1}{16}$ for training.

5 Conclusion and Discussion

In conclusion, this paper explores an innovative region-based visual token representation and presents `Reader`. `Reader` employs arbitrarily shaped and sized regions as the units of image tokens, which aligns better with the visual prior: the non-uniformity of information density and the naturally existing semantic boundaries. Through experiments, we demonstrate the following inherent advantages of this region-based tokenization method. **Efficiency**: This method has a high compression ratio and can achieve superior reconstruction fidelity while employing a significantly smaller number of tokens for each image. **Disentanglement**: The region-based tokens guarantees their disentanglement from one another, and further empower `Reader` with the ability to perform image editing.

Limitation and Feature Work As this paper focuses on exploring the feasibility and effectiveness of region-based tokenization, we adopted a minimal configuration in our experiments, manifested as employing the tiny-version transformer and training model on relatively small-sized images (256×256). Although `Reader` can produce already satisfactory results under such configurations, the parameter size and data resolution still limit its broader application. For example, some high-resolution images may exhibit a degree of blurring after editing.

In the future work, we will pursue scaling up both model parameters and the volume of data. Additionally, we aim to explore its application in various scenarios, such as interaction with LLMs and multimodal perception, generation.

Broader Impact `Reader` possesses the capability to serve as an image editor, yielding numerous interesting outcomes, but also opens up the possibility of real photo manipulation for nefarious purposes. However, with the recent advancements in watermarking techniques and manipulated images identifying techniques [17], this risk is avoidable.

Acknowledgements

This research was funded by the Fundamental Research Funds for the Central Universities (2024XKRC082) and the National NSF of China (No.U23A20314).

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR. pp. 6077–6086 (2018)
2. Binkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying MMD gans. In: ICLR. OpenReview.net (2018)
3. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR. pp. 18392–18402. IEEE (2023)
4. Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M.M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. CoRR **abs/1901.11390** (2019)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) ECCV. Lecture Notes in Computer Science, vol. 12346, pp. 213–229. Springer (2020)
6. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR. pp. 1290–1299 (2022)
7. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) NeurIPS. pp. 17864–17875 (2021)
8. Choi, Y., Choi, M., Kim, M., Ha, J., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR. pp. 8789–8797. Computer Vision Foundation / IEEE Computer Society (2018)
9. Collins, E., Bala, R., Price, B., Süsstrunk, S.: Editing in style: Uncovering the local semantics of gans. In: CVPR. pp. 5770–5779. Computer Vision Foundation / IEEE (2020)
10. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. In: ICLR. OpenReview.net (2023)
11. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: Vqgan-clip: Open domain image generation and editing with natural language guidance. In: ECCV. pp. 88–105. Springer (2022)
12. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR. pp. 4690–4699. Computer Vision Foundation / IEEE (2019)
13. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) NAACL-HLT. pp. 4171–4186. Association for Computational Linguistics (2019)
14. Ding, H., Liu, C., He, S., Jiang, X., Loy, C.C.: MeViS: A large-scale benchmark for video segmentation with motion expressions. In: ICCV (2023)

15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR. OpenReview.net (2021)
16. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR. pp. 12873–12883. Computer Vision Foundation / IEEE (2021)
17. Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T.: The stable signature: Rooting watermarks in latent diffusion models. CoRR **abs/2303.15435** (2023)
18. Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., Taigman, Y.: Make-a-scene: Scene-based text-to-image generation with human priors. In: ECCV. pp. 89–106. Springer (2022)
19. Goel, V., Peruzzo, E., Jiang, Y., Xu, D., Sebe, N., Darrell, T., Wang, Z., Shi, H.: Pair-diffusion: Object-level image editing with structure-and-appearance paired diffusion models. CoRR **abs/2303.17546** (2023)
20. Gondal, M.W., Wuthrich, M., Miladinovic, D., Locatello, F., Breidt, M., Volchkov, V., Akpo, J., Bachem, O., Schölkopf, B., Bauer, S.: On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) NeurIPS. pp. 15714–15725 (2019)
21. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) NeurIPS (2014)
22. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable GAN controls. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) NeurIPS (2020)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778. IEEE Computer Society (2016)
24. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) NeurIPS. pp. 6626–6637 (2017)
25. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) NeurIPS (2020)
26. Horé, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: ICPR. pp. 2366–2369. IEEE Computer Society (2010)
27. Huang, M., Mao, Z., Chen, Z., Zhang, Y.: Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization. In: CVPR. pp. 22596–22605 (2023)
28. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 5967–5976. IEEE Computer Society (2017)
29. Kim, H., Mnih, A.: Disentangling by factorising. In: Dy, J.G., Krause, A. (eds.) ICML. Proceedings of Machine Learning Research, vol. 80, pp. 2654–2663. PMLR (2018)
30. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., Girshick, R.B.: Segment anything. CoRR **abs/2304.02643** (2023)
31. Lee, C., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: CVPR. pp. 5548–5557. Computer Vision Foundation / IEEE (2020)

32. Lee, D., Kim, C., Kim, S., Cho, M., Han, W.: Autoregressive image generation using residual quantization. In: CVPR. pp. 11513–11522. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.01123, https://doi.org/10.1109/CVPR52688.2022.01123
33. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: ECCV. pp. 280–296. Springer (2022)
34. Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S.: Editgan: High-precision semantic image editing. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) NeurIPS. pp. 16331–16345 (2021)
35. Liu, C., Ding, H., Jiang, X.: GRES: Generalized referring expression segmentation. In: CVPR (2023)
36. Liu, C., Jiang, X., Ding, H.: Primitivenet: decomposing the global constraints for referring segmentation. Visual Intelligence **2**(1), 16 (2024)
37. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) NeurIPS (2020)
38. Ning, J., Li, C., Zhang, Z., Wang, C., Geng, Z., Dai, Q., He, K., Hu, H.: All in tokens: Unifying output space of visual tasks via soft token. In: ICCV. pp. 19900–19910 (2023)
39. van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) NeurIPS. pp. 6306–6315 (2017)
40. Pan, X., Tewari, A., Leimkühler, T., Liu, L., Meka, A., Theobalt, C.: Drag your GAN: interactive point-based manipulation on the generative image manifold. In: Brunvand, E., Sheffer, A., Wimmer, M. (eds.) SIGGRAPH. pp. 78:1–78:11. ACM (2023)
41. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763. PMLR (2021)
42. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
43. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) NeurIPS. pp. 1252–1260 (2015)
44. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. NeurIPS **28** (2015)
45. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10674–10685. IEEE (2022)
46. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE TPAMI **44**(4), 2004–2018 (2022)
47. Shuai, X., Ding, H., Ma, X., Tu, R., Jiang, Y.G., Tao, D.: A survey of multimodal-guided image editing with text-to-image diffusion models. arXiv preprint arXiv:2406.14555 (2024)
48. Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Bach, F.R., Blei, D.M. (eds.) ICML. JMLR Workshop and Conference Proceedings, vol. 37, pp. 2256–2265. JMLR.org (2015)
49. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM TOG **40**(4), 133:1–133:14 (2021)

50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) NeurIPS. pp. 5998–6008 (2017)
51. Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K., Vajda, P.: Visual transformers: Token-based image representation and processing for computer vision. arXiv preprint arXiv:2006.03677 (2020)
52. Wu, R., Zhang, G., Lu, S., Chen, T.: Cascade EF-GAN: progressive facial expression editing with local focuses. In: CVPR. pp. 5020–5029. Computer Vision Foundation / IEEE (2020)
53. Xu, J., Mello, S.D., Liu, S., Byeon, W., Breuel, T.M., Kautz, J., Wang, X.: Groupvit: Semantic segmentation emerges from text supervision. In: CVPR. pp. 18113–18123. IEEE (2022)
54. Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., Wen, F.: Paint by example: Exemplar-based image editing with diffusion models. In: CVPR. pp. 18381–18391. IEEE (2023). https://doi.org/10.1109/CVPR52729.2023.01763, https://doi.org/10.1109/CVPR52729.2023.01763
55. Yang, T., Wang, Y., Lu, Y., Zheng, N.: Visual concepts tokenization. In: NeurIPS (2022)
56. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved VQGAN. In: ICLR. OpenReview.net (2022)
57. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: CVPR. pp. 5831–5840 (2018)
58. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595. Computer Vision Foundation / IEEE Computer Society (2018)
59. Zhong, Y., Wang, L., Chen, J., Yu, D., Li, Y.: Comprehensive image captioning via scene graph decomposition. In: ECCV. pp. 211–229. Springer (2020)
60. Zhu, J., Yang, C., Shen, Y., Shi, Z., Zhao, D., Chen, Q.: Linkgan: Linking GAN latents to pixels for controllable image synthesis. CoRR **abs/2301.04604** (2023)