

## RESEARCH ARTICLE

# Can Image Compression Rely on CLIP?

**TOM BACHARD**<sup>1</sup> AND **THOMAS MAUGEY**<sup>1</sup>, (Member, IEEE)

Inria, IRISA, University of Rennes, Campus de Beaulieu, 35042 Rennes, France

Corresponding author: Tom Bachard (tom.bachard@irisa.fr)

This work was supported by the French National Research Agency through MAssively DAta REpurposing (MADARE) under Project ANR-21-CE48-0002 and Contrats Doctoraux en Intelligence Artificielle under Grant ANR-20-THIA-0018.

**ABSTRACT** Coding algorithms are usually designed to faithfully reconstruct images, which limits the expected gains in compression. A new approach based on generative models allows for new compression algorithms that can reach drastically lower compression rates. Instead of pixel fidelity, these algorithms aim at faithfully generating images that have the same high-level interpretation as their inputs. In that context, the challenge becomes to set a good representation for the semantics of an image. While text or segmentation maps have been investigated and have shown their limitations, in this paper, we ask the following question: do powerful foundation models such as CLIP provide a semantic description suited for compression? By suited for compression, we mean that this description is robust to traditional compression tools and, in particular, quantization. We show that CLIP fulfills semantic robustness properties. This makes it an interesting support for generative compression. To make that intuition concrete, we propose a proof-of-concept for a generative codec based on CLIP. Results demonstrate that our CLIP-based coder beats state-of-the-art compression pipelines at extremely low bitrates (0.0012 BPP), both in terms of image quality (65.3 for MUSIQ) and semantic preservation (0.86 for the Clip score).

**INDEX TERMS** Compression algorithms, deep learning, image coding, image processing, image reconstruction, image representation, semantic.

## I. INTRODUCTION

Since decades, strong research efforts have been spent to improve the rate-distortion performance in image compression. On average, gains of 50% are reached every decade [1], [2], [3], [4]. Even though these improvements are impressive, they are not sufficient to cope with the tremendous amount of data produced every day [5].

More recently, a new type of approach has arisen: the *semantic*, or *generative*, compression methods. Their principle is to abandon the pixel fidelity criterion, classically measured with *MSE* (Mean Squared Error), *PSNR* (Peak Signal-to-Noise Ratio) or *SSIM* (Structural Similarity Index Measure) [6]. The motivation behind this is that the important information carried by an image does not reside at the pixel level but instead at a higher level. Moreover, in some applications, having an image that is pixel-wise close to the input is not necessary. Instead, it is sufficient to have a decoded image whose high-level content is preserved. This

is, for example, the case in coding for machines [7] or for cold data [8]. Basically, in such generative compression approaches, an encoder describes the image semantics in a compact form, and a decoder uses a generative method (*e.g.*, Generative Adversarial Network [9] or Diffusion Models [10]) to synthesize an image expressing the coded semantic. One of the research questions is thus: *how to describe the semantics of an image?*

By semantic, we have to understand all that deals with high-level information about an image, *e.g.*, objects, positioning, general atmosphere, and feelings. These features are human-dependent and remain very difficult to capture. However, some attempts have been made in the literature. A first category of methods models the semantics of an image with a segmentation map, *i.e.*, an image whose pixel values indicate the class label. A seminal work [11] proposed to describe an image as a semantic map. This map is used by the decoder to guide a GAN-based decoder. Similarly, [12], [13], [14], [15] estimate the segmentation map at the encoder and reconstruct an image at the decoder thanks to a diffusion model, such that the content of the reconstructed image is

The associate editor coordinating the review of this manuscript and approving it for publication was Chang Choi<sup>1</sup>.

faithful to the segmentation map. Clearly, representing the image semantics with labels can rapidly become limited since the semantics must belong to a predefined list of classes.

Other, more expressive, high-level descriptions have been explored, such as the textual description. In [16] and [17], the input image is mapped to a text that constitutes the compressed semantic information. On the decoder side, a diffusion model is used to generate an image corresponding to the caption. The difficulty resides in generating the text corresponding to an image, which is not always straightforward. To overcome this challenge, [18] proposed to complete the textual compressed vector with a compressed sketch of the input. This addition helps guide the generative model to reconstruct images structurally closer to the inputs.

Recently, foundation models have been explored to represent information in an embedding space. This can further be used for several applications, such as [19], which unifies image generation and image compression, or [20], which uses large language models to extract the semantic description of images compactly. One of the most commonly used foundation models is *CLIP* (Contrastive Language-Image Pretraining) [21]. In a nutshell, CLIP is trained to align, in the same embedding space, the vectors representing the image content and its corresponding textual caption. As a consequence, one part of the CLIP model can take an image as an input and map it to a vector in its latent space. From the way CLIP is trained, we can expect this vector to represent the image semantics in some way or another.

In this paper, we ask the following question: *Is CLIP suitable for image compression?* More precisely, we wonder to what extent CLIP represents the semantics of an image and if CLIP's latent vectors are robust to transformations applied through traditional compression tools (and in particular quantization). To tackle these questions, we define two properties that CLIP must satisfy. The first property deals with how faithful the description of the semantics of the image is to the CLIP representation. For the second property, we investigate how compact the CLIP representation is so that it can help reach low bit rates. In the same spirit as [22], which explores the latent space of diffusion models, and [23] that explores the limits of CLIP for image compression, we first propose an experimental methodology used to demonstrate that CLIP possesses these properties. Finally, we propose a proof-of-concept CLIP-based generative coder, highlighting the huge potential for image representation to rely on CLIP.

In this work, the main contributions are the following:

- We derive two properties that a semantic representation must fulfill when it is used in the context of image compression;
- We experimentally prove that CLIP satisfies the two aforementioned properties on multiple datasets;
- We propose a proof-of-concept CLIP-based compression scheme, and we show that it outperforms classical codecs both in terms of quality and semantics conservation at extremely low bitrates.

## II. PROBLEM FORMULATION

### A. MODELING IMAGE SEMANTICS WITH CLIP

An image is usually represented as a vector  $\mathbf{x} \in \mathbb{D}^N$ , where  $N$  is the dimension of the image. Each vector element,  $x[n]$ , describes a pixel color, represented in a color domain  $\mathbb{D}$ . Typically,  $\mathbb{D} = \llbracket 0, 255 \rrbracket^3$  for RGB format. While each pixel value gives point-wise color information, the concatenation of these pixels can form more general concepts such as contours, textures, shapes, etc. Going further, the concatenation of these concepts can lead to a high-level interpretation of the scene described by the image (*e.g.*, objects, actions, atmosphere, feelings). These elements are typically referred to as the general concept of *semantic*. In the following, we denote by  $\text{sem}(\mathbf{x})$  the semantics of an image  $\mathbf{x}$ .

Modeling the sem function has been an intensive research topic for a long time (image representation [24], image embedding [21]). Recently, foundation models, and more specifically CLIP [21], have been recognized as powerful tools to model image semantics [25]. Concretely, the CLIP method casts an image onto a reduced space  $\mathcal{L} \subset \mathbb{R}^M$  where  $M$  is the dimension of the CLIP space. In the following,  $\mathcal{L}$  is called the CLIP latent space:

$$\begin{aligned} f: \mathbb{D}^N &\longrightarrow \mathcal{L} \\ \mathbf{x} &\longmapsto \mathbf{z} \end{aligned} \quad (1)$$

where  $M < N$ . We indeed look for this inequality for two main reasons: first, as we cast this work in a compression paradigm, it is interesting to gradually reduce the dimension of the data in the pipeline. Second, as we suppose that the latent space of CLIP is more semantic than the pixel domain, we suppose that the dimensions needed to encapsulate the high-level description of the image are lower than the dimensions of the pixel space. The function  $f$  has been trained such that two images  $\mathbf{x}_1$  and  $\mathbf{x}_2$  with close semantic have aligned CLIP vectors  $\mathbf{z}_1$  and  $\mathbf{z}_2$ . The following property is thus, by construction, verified:

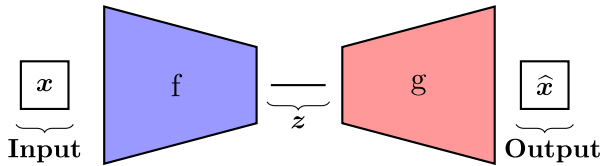
*Property( $\mathcal{P}_0$ ):* For two images  $\mathbf{x}_1$  and  $\mathbf{x}_2$  and their respective CLIP representation  $\mathbf{z}_1 = f(\mathbf{x}_1)$  and  $\mathbf{z}_2 = f(\mathbf{x}_2)$ ,

$$\text{sem}(\mathbf{x}_1) \approx \text{sem}(\mathbf{x}_2) \Leftrightarrow \frac{\mathbf{z}_1^\top \mathbf{z}_2}{\|\mathbf{z}_1\|_2 \|\mathbf{z}_2\|_2} = \cos(\mathbf{z}_1, \mathbf{z}_2) \approx 1 \quad (2)$$

This property  $\mathcal{P}_0$  has been very useful for many tasks (such as classification [21], [24]) as the semantics of two images can easily be compared by computing the cosine between their respective CLIP representations.

### B. IS CLIP SUITABLE FOR GENERATIVE COMPRESSION?

In this paper, we would like to study whether CLIP's latent space respects additional properties that could be useful for other image processing tasks, such as our task of interest in this work: compression. Recently, some algorithms have been developed to explore the problem of image compression at *extremely low bitrates* [26]. In such conditions, when coding an image  $\mathbf{x}$ , trying to be faithful to the original image's pixels



**FIGURE 1.** Encoding-decoding pipeline with CLIP, as  $f$  and unCLIP, as  $g$ . The input image is noted  $x$ , the latent vectors  $z$  and the output image  $\hat{x}$ .

is no longer efficient [27]. Instead, it is preferable to describe the image by its semantics  $\text{sem}(x)$ , especially since the arrival of powerful image generation techniques that enable reconstructing images from a semantic description [15], [26]. These so-called *generative compression* algorithms can, for example, rely on CLIP. In those cases, an image  $x$  is encoded by  $f$ , and the compact semantics vector  $z = f(x)$  is used to guide a generative model, denoted by  $g$  in the following. These algorithms allow for reaching extremely low bitrates and to decode an image  $\hat{x}$  such that  $f(x) \approx f(\hat{x})$ .

To verify that a CLIP-based compression approach is meaningful, we must verify that having  $f(x) \approx f(\hat{x})$  implies that  $\text{sem}(x) \approx \text{sem}(\hat{x})$ . In other words, we have to evaluate how exhaustively the function  $f$  captures the semantics of an image. We therefore consider the following property:

*Property ( $\mathcal{P}_1$ ): For an image  $x$ ,*

$$\text{sem}(x) \approx \text{sem}(g \circ f(x)) \quad (3)$$

The property  $\mathcal{P}_1$  is investigated in Sec. IV.

In a CLIP-based generative compression architecture, the CLIP vector  $z$  of an image  $x$  constitutes the main element of the code-word.<sup>1</sup> The size of the compressed image is thus strongly linked to the number of bits necessary to describe the vector  $z$ . This number can be reduced by performing a quantization (denoted by  $q$ ), as classically done in conventional compression schemes. The quantization  $q$  consists in reducing the size of the alphabet with which the elements of  $z$  are expressed. This can be done only if it does not affect the semantics of the decoded image  $\hat{x}$ . We then explore the following property in Sec. VI:

*Property ( $\mathcal{P}_2$ ): For an image  $x$ ,*

$$\text{sem}(g \circ f(x)) \approx \text{sem}(g \circ q \circ f(x)) \quad (4)$$

### III. METHODOLOGY

In this section, we define the set-up in which we study the properties  $\mathcal{P}_1$  and  $\mathcal{P}_2$ . First, we introduce the pipeline architecture: the models and datasets used. In a second time, we present and discuss the different metrics used to evaluate the images: the quality metrics and how we plan to evaluate the preservation of the semantics between the inputs and the outputs.

<sup>1</sup>The CLIP vector might be completed by some light additional information to bring more consistency between  $x$  and  $\hat{x}$ .

### A. PROPOSED FRAMEWORK

Figure 1 presents the studied codec (encoder-generator) for this work. Input images  $x$  are encoded with  $f$ , the image encoder, into a latent vector  $z$  via  $f(x) = z \in \mathcal{L}$ , where  $\mathcal{L}$  is the latent space. Finally,  $g$ , the image generator, reconstructs outputs images  $\hat{x}$  from the latent vectors  $\hat{x} = g(z)$ .

#### 1) MODELS

For the image encoder  $f$ , we use CLIP [21], as it is a popular foundation model for image embedding. Specifically, we use the Vital/14 version of the model. In this version, images are encoded in a 768-dimensional (thus  $\mathcal{L} \subset \mathbb{R}^{768}$ ) vector coded on 16-bits vectors. For the image generator  $g$ , we use the stable unCLIP [28] model, a CLIP fine-tuned latent diffusion model based on the Stable Diffusion model [29]. The used weights can be found here.<sup>2</sup>

We specify that CLIP and Stable unCLIP are *not* fine-tuned nor retrained for any of the experiments presented in this work.

#### 2) DATASETS

In this work, we benchmark our explorations on multiple datasets to prove the aforementioned properties. The first dataset used for benchmarking is Kodak [30]. This is a classical dataset used for evaluating and comparing compression pipelines. We also evaluate the pipeline on images from two other datasets: Landscape [31] and CelebA [32]. The former has been selected as it is expected to behave nicely in the context of semantic generative compression – as landscapes in general were used to train CLIP and also have an easily extractable high-level interpretation. The latter, on the other hand, was used as it was not expected to easily comply with semantic compression. Indeed, faces were explicitly removed from CLIP training set to avoid generating known people into displeasing images. Also note that the semantic high-level description of faces is far more complicated to grasp [33].

### B. EVALUATION OF THE GENERATED IMAGES

By nature, classical MSE-based metrics are not efficient for evaluating a generative coding pipeline. Instead, we have to assess to what extent the semantic is conserved during compression. We also have to ensure the quality of the generated images.

#### 1) SEMANTIC CONSERVATION METRICS

To evaluate the semantic fidelity, we first propose to compute a segmentation map of both images, and then compare them. Concretely, the segmentation maps are computed with [34], which is a Deeplab implementation with a ResNet101 backbone [35]. The segmentation maps can be made of more than a hundred of classes, based on the classes of the MS-COCO dataset [36]. We represent the segmentation map as a vector  $s$ , in which each component  $s[i]$  corresponds to one class and depicts the proportion of the image belonging to this class. We denote by  $s^b$  its binary version, where only the

<sup>2</sup>[https://huggingface.co/docs/diffusers/api/pipelines/stable\\_unclip](https://huggingface.co/docs/diffusers/api/pipelines/stable_unclip)



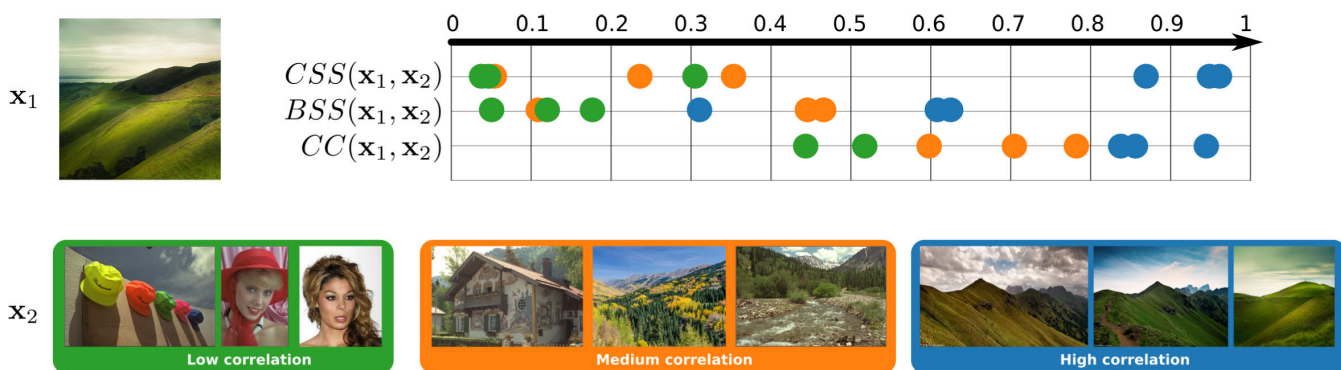


FIGURE 2. Some examples of semantic conservation scores obtained at different levels of correlation.



FIGURE 3. Examples of generated images with  $g \circ f$ . (Row-wise, top to bottom) Inputs respectively taken from Landscape, CelebA and Kodak. For each sub-figure, the input is the top left image, and the generated images are the three others.

presence or absence of a class is described. Let us consider two segmentation maps represented in their vector forms:  $s_1$  and  $s_2$ , taken from two images,  $x_1$  and  $x_2$ . To compare these

segmentation maps, we define two scores:

$$CSS(x_1, x_2) = \frac{s_1^T s_2}{\|s_1\|_2 \|s_2\|_2} \quad (5)$$

$$\text{BSS}(x_1, x_2) = \frac{\|s_1^b \wedge s_2^b\|_1}{\|s_1^b \vee s_2^b\|_1} \quad (6)$$

The CSS shows to what extent the content of both images is semantically the same – regardless of where the different classes are present in the images. However, this score will not be able to detect whether the generated images produce weird, small artifacts that were not initially present in the inputs. On the other hand, the BSS tells us to what extent both images share the same semantic class, regardless of their importance in the image. Both of these metrics range from 0 to 1, the latter being the better.

Another way to evaluate semantic similarity between two images is to compare their CLIP latent representation  $z_1$  and  $z_2$  [37]:

$$\text{CC}(x_1, x_2) = \frac{z_1^\top z_2}{\|z_1\|_2 \|z_2\|_2}, \quad (7)$$

ranging from 0 to 1, the latter being the better.

Finally, to ensure that the generated images follow the same semantic distribution as the inputs, we compute the Fréchet Inception Distance (FID) between the input images and the generated images. This metric gives the distance between two groups of images, considered as probability distributions in the latent space of a certain classification model. In this work, we use the Torch Metric implementation [38], based on the third version of the Inception model [39]. This metric is a distance, so the closer to 0, the better.

To give some intuition about the behavior of these semantic metrics, we compare the expected high-level semantic correlation to the one given by the metrics. Figure 2 gives the typical value scores obtained for different levels of semantic correlation. These metrics are relevant to the proposed study, as the obtained scores correlate with the human-given semantic correlation.

## 2) NO-REFERENCE IMAGE QUALITY METRICS

Generated images have to follow natural images distribution. To ensure this, we propose to evaluate the realism of the outputs, regardless of the original input; we use Image Quality Assessment (IQA) metrics.

To evaluate realism, we use two no-reference IQA metrics: MUSIQ [40] and DBCNN [41]. MUSIQ is a multiscale image quality transformer processing images with varying resolutions and ratios. DBCNN is a deep bilinear model for blind image quality assessments specialized in synthetic and authentic distortions, one for each network. The higher, the better for both of these IQA metrics.

## IV. PRESERVATION OF IMAGE SEMANTICS WITH CLIP (PROPERTY $\mathcal{P}_1$ )

In this section, we evaluate how the property  $\mathcal{P}_1$  is verified experimentally. Specifically, we would like to measure how much the CLIP function  $f$  captures the semantics of images. For that purpose, a high number of images are processed with the pipeline  $g \circ f$ , depicted in Figure 1. First, we ensure

**TABLE 1. No-reference quality metrics applied to kodak, landscape, and CelebA. (Columns 1 and 3) original images. (Columns 2 and 4) images generated via  $g \circ f$ .**

	DBCNN (i)	DBCNN (g)	MUSIQ (i)	MUSIQ (g)
Kodak	0.69( $\pm 0.04$ )	0.43( $\pm 0.1$ )	74.6( $\pm 2.3$ )	58.6( $\pm 9$ )
Landscape	0.61( $\pm 0.1$ )	0.48( $\pm 0.1$ )	68.3( $\pm 6.8$ )	63.9( $\pm 7.2$ )
CelebA	0.58( $\pm 0.13$ )	0.32( $\pm 0.1$ )	51.8( $\pm 10$ )	49( $\pm 10.3$ )

that the quality of the rendered image is perceptually good and that the  $\hat{x}$  are semantically close to  $x$  visually. Then, we propose a quantitative assessment method for semantic coherence evaluation and show that the proposed pipeline indeed preserves the inputs' semantic.

### A. QUALITATIVE ASSESSMENT EVALUATION

To assess the quality of the generated images, we process a set of 100 images from the three datasets with the pipeline  $g \circ f$  depicted in Figure 1. For the sake of robustness, we generate 3 images  $\hat{x} = (g \circ f)(x)$  per input. First, we would like to ensure that the generated images are good-looking (for the moment, without any consideration of the input). Hence, we measure the quality of each  $\hat{x}$  with the no-reference metrics introduced in Section III-B. Each metric is also benchmarked on original images, without any modifications, to estimate the scores on natural images.

Table 1 presents the no-reference scores obtained for each tested dataset. We first observe that the metrics are coherent from one to another, which means that they are reliable. Second, we observe that the input images' scores ( $i$  columns) are slightly better than the ones from the generated images ( $g$  columns). While this decrease in quality can be considered a problem for generating images, we observe from Figure 3 that the quality of the output is sufficient for the generative compression needs. For each example, we verify that the generated images  $\hat{x}$  are visually coherent with their respective input image. This tends to prove that the image semantics are well captured by the function  $f$ . To go further in the demonstration, we propose a quantitative assessment in the following.

### B. QUANTITATIVE ASSESSMENT EVALUATION

Now that we are armed with simple yet effective semantic metrics, we can evaluate to what extent the generation pipeline preserves the semantics of the input image. To do so, we select 100 images, and we generate 3 variations from each latent vector. We evaluate, for each dataset, the semantic score of the outputs regarding the inputs: each input image is compared to each of the 3 generated variations.

Table 2 shows the semantic score of the generated images regarding their inputs. We first observe that, similarly to the previous semantic evaluation experiments, the four metrics are correlated, here in the high values. Indeed, we observe a high semantic correlation (from 0.79 to 0.89 CC, 1.43 FID for Landscape and CelebA and 0.93 CSS for CelebA) between the generated images and their counterpart



**TABLE 2. Semantics metrics applied to images generated from Kodak, Landscape and CelebA via  $g \circ f$ .**

	CC	FID	BSS	CSS
Kodak	0.86 ( $\pm 0.05$ )	5.83	0.43 ( $\pm 0.1$ )	0.76 ( $\pm 0.2$ )
Landscape	0.89 ( $\pm 0.06$ )	1.43	0.49 ( $\pm 0.1$ )	0.75 ( $\pm 0.2$ )
CelebA	0.79 ( $\pm 0.06$ )	1.45	0.44 ( $\pm 0.2$ )	0.93 ( $\pm 0.1$ )

inputs. In terms of “high-level” correlation, we are in the “highly correlated images” range for the previous subsection experiment. Some low values can be observed, such as 5.83 FID for Kodak. We suppose that the FID is not a good metric for evaluating semantics for highly heterogeneous datasets, as their statistical features may not be highly correlated and more images may be required for more precise results.

These values can be compared to those of Figure 2. We observe, as expected, that the generated images are around the highly correlated values for each metric. This demonstrates that the property  $\mathcal{P}_1$  is, in general, verified, and that we can rely on the CLIP function  $f$  to model quite exhaustively the sem function.

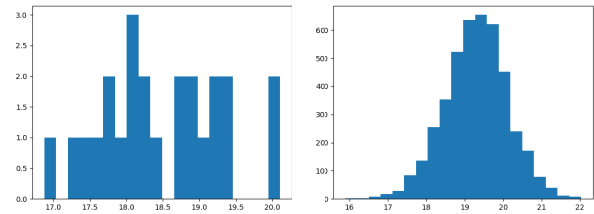
## V. SHAPE OF THE CLIP LATENT SPACE $\mathcal{L}$

Before tackling property  $\mathcal{P}_2$ , we must acknowledge that in property  $\mathcal{P}_2$ , the latent vectors (resulting from the mapping of an image with  $f$ ) are modified due to quantization, before being processed by the generator  $g$ . There is no guarantee that performing such operations in the latent space is compatible with the way the generator  $g$  was trained (and we recall that we want to avoid retraining or even fine-tuning  $f$  or  $g$ ). In this section, we first investigate the shape of  $\mathcal{L}$  and we show that it is included in a thin spherical shell of dimension  $M$ , and thus can be approximated by a sphere  $M$  dimensional  $\hat{\mathcal{L}}$ . In a second time, we define an operator  $\pi$  that projects vectors of  $\mathbb{R}^M$  onto  $\hat{\mathcal{L}}$ . The goal is to use  $\pi$  to move the modified CLIP latent vectors to a space that is safe for the generator  $g$ .

### A. $\mathcal{L}$ AS A SPHERICAL SHELL

At first glance, the CLIP latent space  $\mathcal{L}$  is a subset of  $\mathbb{R}^M$  ( $M = 768$ ) with no *a priori* organization. To find the general shape of  $\mathcal{L}$ , it is important to note that the function  $f$  is trained such that the mapping of an image  $f(x)$  is aligned with the embedding of its textual description. Moreover, the training strategy leads to the property  $\mathcal{P}_0$ , stating that two images are semantically correlated if the cosine similarity between their CLIP description is close to 1. For all these reasons, one can expect that the CLIP latent vectors are characterized by their orientation in  $\mathbb{R}^M$ . It is thus reasonable to assume that  $\mathcal{L}$  has a seemingly spherical shape.

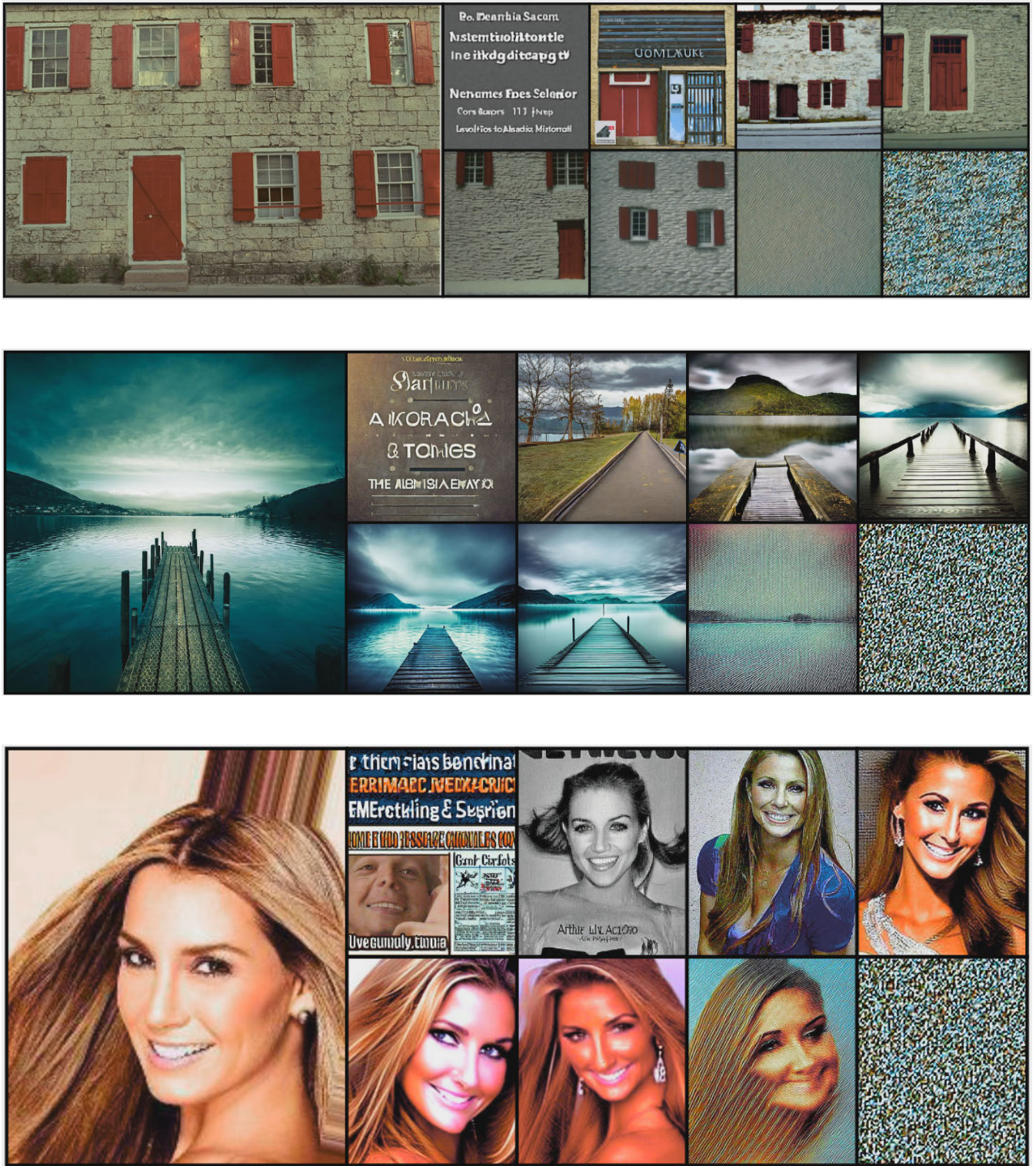
This hypothesis is verified when looking at the distribution of the norms of the encoded images of Kodak and Landscape, pictured in Figure 4. We observe that the norms are concentrated on a given value ( $\approx 19$  or  $20$ ), which can be interpreted as the radius  $r_{\text{qual}}$  of a sphere. We then hypothesize that  $\mathcal{L}$  is included in a thin spherical shell. Said differently, we hypothesize that most of the CLIP latent vectors of natural

**FIGURE 4. Norm distribution of the latent vectors for input images from Kodak (left) and from Landscape (right).**

images lie in an  $M$ -dimensional spherical shell. By doing so, we need to show that the radius contains no information regarding the semantic.

To verify this statement, we map images  $x$  to their latent vectors  $z$ , and in a second time, rescale each latent vector with a factor  $\lambda \in [0.1, 2.5]$ :  $z_\lambda = \lambda z$ . We then generate images  $\hat{x}_\lambda$  from the rescaled latent vectors  $\hat{x}_\lambda = g(\lambda f(x))$ . Finally, we evaluate the quality of the generated images, and more importantly, we evaluate their semantic coherence with the input image. A visual toy example is presented in Figure 5, and quantitative results are presented in Figure 6. From Figure 5 we observe that indeed the semantic coherence with the input images seems to be maintained for  $\lambda$  values not too far from 1, proving the spherical shell form of  $\mathcal{L}$ . More precisely, we observe that for low values of  $\lambda$ , *i.e.*,  $\lambda < 0.75$ , the generated image is either gibberish or of a lesser quality in terms of structural coherence. Furthermore, the semantics seem to change as well, becoming broader and more general. For high values of  $\lambda$ , *i.e.*  $\lambda > 1.5$ , we observe that the generated images are more and more noisy as  $\lambda$  increases, showing that the model has not been trained to generate images from latent vectors whose norms are too big. Finally, as expected by our intuition, when the values of  $\lambda$  are reasonably close to 1, the generated images show no differences from control experiments in Fig. 3. We conclude that the generator  $g$  does not work for either high rescaling values or low rescaling values, demonstrating once again that the meaningful latent vectors should be placed not too far from the sphere of radius  $r_{\text{qual}}$ . This trend is confirmed when looking at the quantitative results in Figure 6. More precisely, we observe two interesting phenomena. First, we see that the quality of the generated images does not depend on the scale factor, except for tiny and considerably large latent vectors, as the DBCNN and MUSIQ graphs show. This simply demonstrates that the generator  $g$  has been trained to maximize the quality of the outputs, regardless from where in  $\mathcal{L}$  the latent vector  $z$  has been drawn. The second phenomenon we observe in this figure concerns the semantic metrics. Indeed, for CC, BSS, CSS, and FID to a lesser extent, the semantic coherence between the inputs and the outputs is maximized (minimized for the FID) on a plateau around  $\lambda = 1$ . When the scale factor is too far from this plateau, *i.e.*,  $\lambda < 0.5$  or  $\lambda > 1.5$ , the semantic coherence quickly drops in quality.



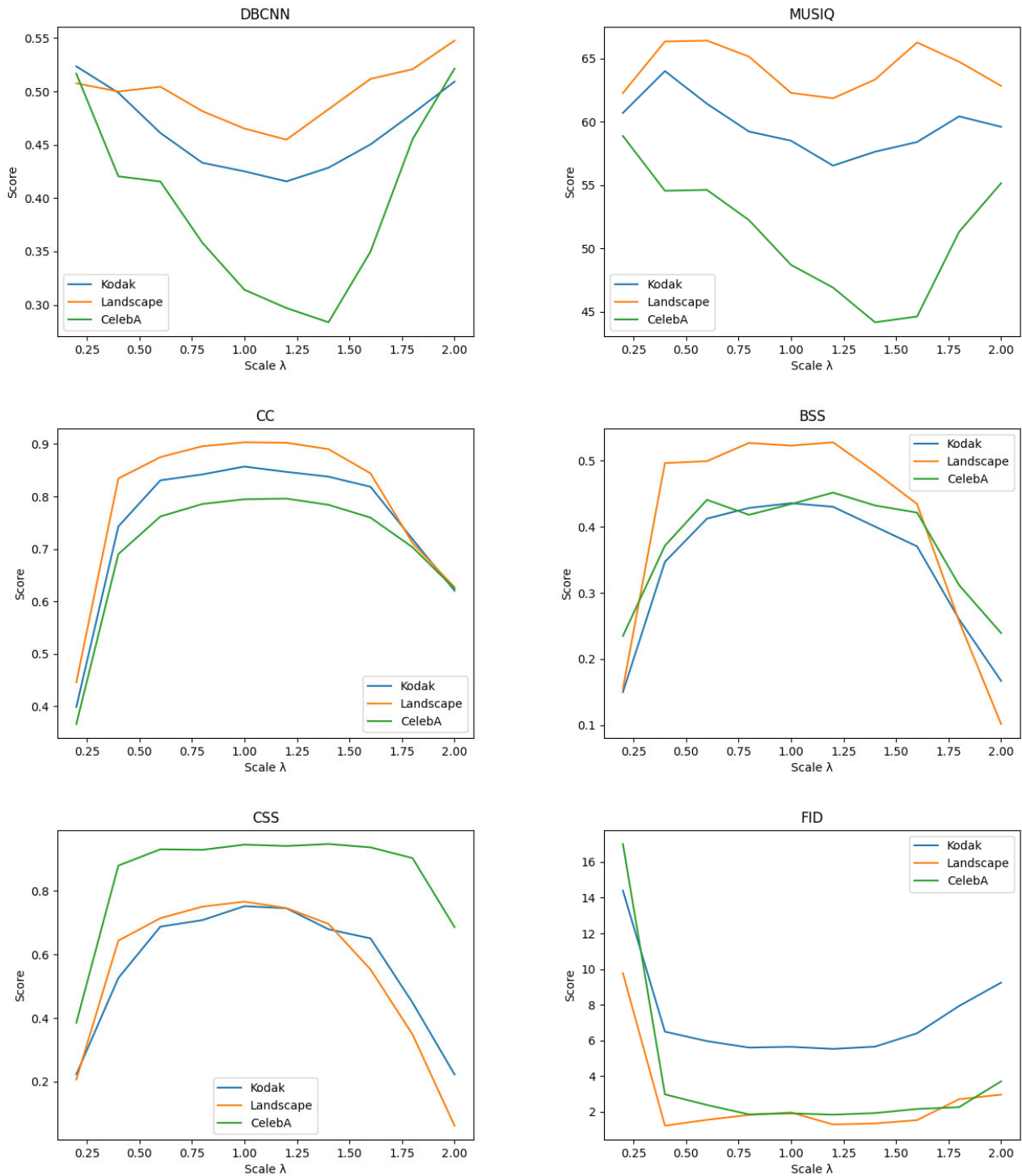


**FIGURE 5.** Visual examples of generating from scaled latent vector. (Top to Bottom) Inputs respectively taken from Kodak, Landscape and CelebA. For each sub-figure: (Left) Input image. (Right, left to right, top to bottom) Generated images from the scaled latent:  $\lambda \in [0.1, 0.25, 0.5, 0.75, 1.25, 1.5, 2, 4]$ .

This experiment confirms that  $\mathcal{L}$  is included in a thin spherical shell. To guarantee good visual quality and semantics coherence, we propose, in the next section, an operator  $\pi$  that maps  $\mathbb{R}^M$  vectors onto  $\hat{\mathcal{L}}$ , an  $M$ -dimensional sphere included in  $\mathcal{L}$ .

**B. PROJECTION ONTO  $\hat{\mathcal{L}}$**

Applying quantization operations on latent vectors in  $\mathcal{L}$  may displace the resulting latent vectors in a region of  $\mathbb{R}^M$  that is outside the interesting subspace  $\mathcal{L}$ , where coherent generation is not guaranteed. In this subsection, we motivate



**FIGURE 6.** Quality and semantic coherence scores regarding the scale factor  $\lambda$ , applied to Landscape, Kodak and CelebA. (Top to bottom, left to right) DBCNN, MUSIQ, CC, BSS, CSS, and FID.

and introduce a projection operator  $\pi : \mathbb{R}^M \rightarrow \hat{\mathcal{L}} \subset \mathcal{L}$  that preserves the semantics after projection while ensuring good generative properties.

In Section V-A, we showed that, around a certain norm, the latent vectors share the same semantic. This is especially true when the displacement is radial. Figure 6 shows that



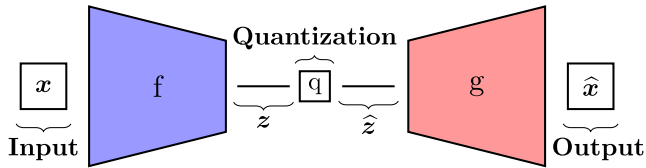


FIGURE 7. Proposed quantization pipeline.

the cosine similarity between different latent vectors is conserved. This result motivates the definition of the  $\pi$  operator as a radial rescaling operator. Because the semantic along a line passing through the origin is the same, we only have to maximize the quality of the output image. This rescaling value is given by the most representative norm of the encoded images from our datasets. According to Figure 4, this value should be set to around  $r_{\text{qual}} \simeq 19.5$ . By doing so, we approximate the latent space  $\mathcal{L}$  by a spherical sphere  $\widehat{\mathcal{L}} \subset \mathcal{L}$ .

We thus define the following projection operator  $\pi$ , that is applied to all the latent vectors before generation, when they result from any displacement in the latent space:

$$\begin{aligned} \pi : \mathbb{R}^M &\longrightarrow \widehat{\mathcal{L}} \subset \mathcal{L} \\ z &\mapsto \frac{r_{\text{qual}}}{\|z\|} z \end{aligned} \quad (8)$$

## VI. QUANTIZATION IN THE LATENT SPACE (PROPERTY $\mathcal{P}_2$ )

In this section, we discuss the property  $\mathcal{P}_2$ . We are looking at the effects of quantization in the latent space on generated images. First, we discuss the quantization pipeline and how we model the quantization process, first as a uniform bit reduction per dimension and then as an additive Gaussian noise. We then evaluate both the quality and the semantic preservation of the generated images for both quantization models.

### A. METHODOLOGY

To test how much the CLIP latent dimension can be reduced (property  $\mathcal{P}_2$ ), we introduce a new operator  $q$  performing quantization in the latent space. This operator is represented in Figure 7. The quantized vectors are noted  $\hat{z}$  and the generated images  $\hat{x}$ . Note that because we use an operator that may cast the latent vectors outside  $\mathcal{L}$  (see Section V-B), we also have to compose with the  $\pi$  operator after the quantization step. Indeed, this ensures that the latent vectors end up in  $\widehat{\mathcal{L}} \subset \mathcal{L}$ , a suitable space for image generation.

In this work, we explore quantization in its simplest form: uniform quantization alongside each dimension of the latent vector. This is motivated by the fact that classical encoders, such as JPEG, also use this form of quantization [42]. For a given quantization step  $q$  (usually  $q = 2^{-b}$  where  $b$  is the number of bits allowed per dimension), the quantization is performed as follows:

$$\hat{z} = q(z) = \lfloor \frac{\bar{z}}{q} \rfloor q + \frac{q}{2} \quad (9)$$

where  $\forall i, \bar{z}[i] = \max(\min(z[i], 1), -1)$

The lower and upper bounds for quantization are set to  $\pm 1$  as prior experiments showed no degradations in the reconstructed images with wider ranges. As the models used in this pipeline use 16-bits vectors as inputs, the quantization experiments only consider 16 bits through 1 bit per dimension quantization.

### B. EFFECT OF UNIFORM QUANTIZATION

To observe the effects of uniform quantization on generated images, we evaluate both the quality and the semantic conservation of generated images where the latent vectors have been compressed with different levels of quantization. To quantify these effects, we select 20 images from different datasets, encode their latent vectors with  $f$ , quantify them ( $b \in [1, 2, 4, 8, 16]$ ) and generate 3 variations for each quantified latent vector. We then compare  $x$  and  $\hat{x}$  with the metrics introduced in Sec III-B. In particular, these metrics measure the quality of the rendering and the semantic similarity.

Visual examples of this experiment are presented in Figure 8. We observe that, regardless of the quantization level, the generated images seem both semantically close to their respective inputs and qualitative. No degradation can be observed, even when the  $\hat{z}[i]$  are represented with only 1 bit. This tendency is confirmed by Figure 9 where we do not observe any significant score decreasing for any metric, except a slight decrease for the CC score when going from 2 bits to 1.

These experiments strongly suggest that the CLIP latent vectors  $z$  can be represented in a coarse, quantized form without affecting the content of the generated image. These results are interesting for two reasons. First, it indicates that we can drastically reduce the number of bits necessary to describe CLIP latent vectors when they are used as a compressed representation of images. Second, it shows that the shape of CLIP’s latent space seems pretty smooth, in the sense that the neighborhood of a CLIP latent leads to pretty consistent generation results in terms of semantics. We further investigate these observations in the next section.

### C. GAUSSIAN NOISE

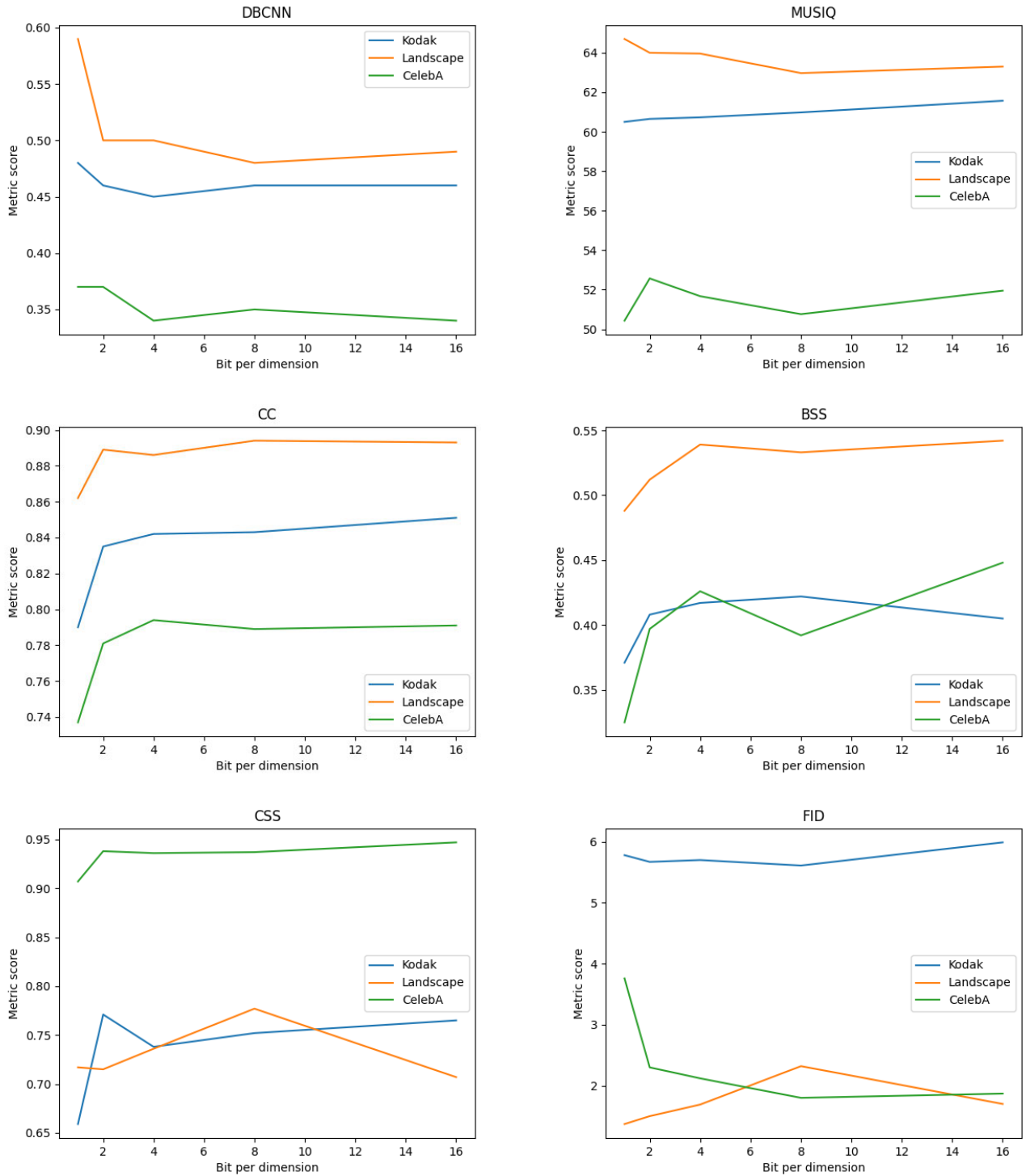
As shown in the results of the previous experiments, even harsh compression does not seem to impact the quality of the generated images or the semantic relations with their inputs. Only slight semantic degradations seem to appear when going from 2 bits per dimension to 1. In this section, we would like to investigate how far from each other two CLIP latent vectors can be and still lead to the same generated images in terms of semantics. For that purpose, we consider an experiment where we map an image to a latent vector  $z$ , to which we add a Gaussian noise with a fixed controllable variance  $\Sigma = \sigma I$ . This operation classically mimics quantization, with the quantization step being controlled by the variance of the noise:

$$\hat{z} = z + \eta, \text{ where } \eta \sim \mathcal{N}(0, \Sigma) \quad (10)$$



**FIGURE 8.** Generated images from quantized latent vectors. (Left to Right) Inputs respectively taken from Kodak, Landscape, and CelebA. For each sub-figure: (Top) Input image. (Middle to Bottom) Variation with different levels of quantization. (Left to Right, Top to Bottom) Quantization: 1 bit, 2 bits, 4 bits, and 8 bits.





**FIGURE 9.** Quality and semantic coherence scores based on the quantization level, applied to Landscape, Kodak and CelebA. (Top to bottom, left to right) DBCNN, MUSIQ, CC, BSS, CSS, FID.

For each noisy latent vector  $\hat{z}$ , where  $\sigma$  ranges from 0.1 to 2.5 with a 0.1 step, we generate 3 variations of each input images. The results from different datasets are presented in 10, for the quantitative results, and in Figure 11, visual examples.

From the different metric scores, presented in Figure 10, at low variance noise, we observe that the generated images from the noisy latent vectors are semantically close to their original inputs while still being of good quality. Furthermore, the more the noise variance increases, the farther semantically

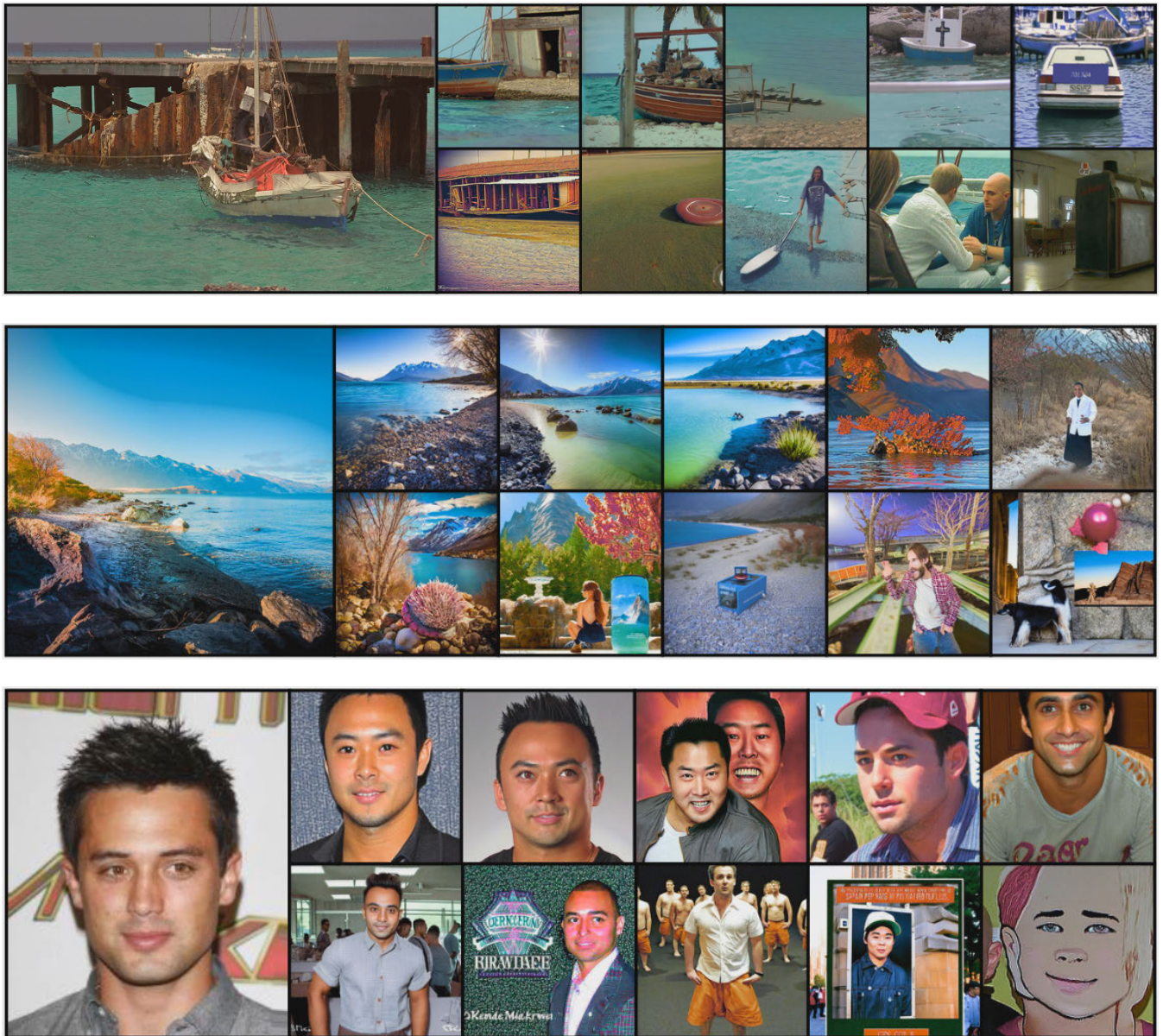


**FIGURE 10.** Quality and semantic coherence scores, regarding the proportion of noise added regarding the norm of the input latent vector, applied to Landscape, Kodak, and CelebA. Scattered crosses represent the quantized images generated in Section VI-B. (Top to bottom, left to right) DBCNN, MUSIQ, CC, BSS, CSS, FID.

the generated images are from their respective inputs. However, we observe that, even at high variance noise, the generated images are still qualitative in terms of natural images (regardless of the original latent vector or image). This

further strengthens the interpretation that unCLIP is trained to generate natural images from any latent vector, regardless if it can be obtained from a natural image or not. These observations are confirmed by the visual examples presented





**FIGURE 11.** Visual evolution of the generation of the addition of increasing Gaussian noise on the latent vector. (Top to bottom) Inputs respectively taken from Kodak, Landscape, and CelebA. For each sub-figure: (Left) Input image. (Right, top to bottom, left to right) Variation with different levels of noise. From left to right, top to bottom:  $\sigma \in [0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2]$ .

in Figure 11. We indeed observe that for a small noise, *i.e.*  $\sigma < 0.6$ , the generated images are semantically close to their respective inputs and most high-level details are kept (some exceptions arise, such as the second face in the CelebA example). For a moderate noise addition, *i.e.*  $0.6 \leq \sigma \leq 1.4$ , only the structural semantics is conserved, while medium-to-high-level details are modified or suppressed. For example, we observe the apparition of a priest or shellfish in the Landscape example or a complete change of topic in the Kodak example. Finally, when the added noise is too big, *i.e.*  $\sigma > 1.4$ , the generated images have low-to-no-semantic resemblance with their current inputs. This can especially be observed in the Landscape and Kodak examples. Note

that, for some of these examples, some generated images with large noise may seem closer than some generated with lower noise. While this is statistically wrong, see Fig. 10, we chose not to cherry-pick the results to also highlight the fact that some directions may alter the semantics in a worse way than others. This work is, however, suitable for future study.

Furthermore, we added the scores of the metrics we obtained with the previous quantization experiments on Figure 10. We observe that the quantization steps, even the harsher ones at 1 bit or 2 bits per dimension, lay in the no-to-small noise variance area. This explains the good rendering results we obtain with the previous quantization experiment.

**TABLE 3. Quality and semantics quantification evaluations between VVC and the proposed method at very low bitrates. Scores calculated on Landscape.**

	BPP	DBCNN	MUSIQ	CC	BSS	CSS	FID
VVC [43]	0.0045	0.19 ( $\pm 0.02$ )	20.33 ( $\pm 5.23$ )	0.64 ( $\pm 0.06$ )	0.091 ( $\pm 0.05$ )	0.116 ( $\pm 0.15$ )	3.5
[18] (without sketches)	0.003 ( $\pm 0.001$ )	0.7 ( $\pm 0.04$ )	73.3 ( $\pm 2.5$ )	0.84 ( $\pm 0.04$ )	0.44 ( $\pm 0.11$ )	0.68 ( $\pm 0.13$ )	1.48
[18] (with sketches)	0.026 ( $\pm 0.005$ )	0.68 ( $\pm 0.06$ )	71.1 ( $\pm 3.0$ )	0.87 ( $\pm 0.04$ )	0.52 ( $\pm 0.2$ )	0.65 ( $\pm 0.28$ )	1.69
Ours	0.0012	0.58 ( $\pm 0.12$ )	65.3 ( $\pm 4.7$ )	0.86 ( $\pm 0.09$ )	0.48 ( $\pm 0.07$ )	0.72 ( $\pm 0.14$ )	1.53

We can conclude that, while already very compact in its 16-bits precision form, the CLIP latent  $z$  of an image can be largely compressed until reaching the impressive size of 1 bit per component (*i.e.*, a compression ratio of 16). This demonstrates that CLIP manages to describe the image semantics well and can be a good representation for a generative compression method. This is what we develop in the next section.

## VII. CLIP-BASED SEMANTICS GENERATIVE CODING SCHEME

This section introduces a simple compression scheme based on the discussions of  $\mathcal{P}_1$  and  $\mathcal{P}_2$ . We compare the proposed compression algorithm with the intracodec of VVC [43] at low bitrates, as well as Text+Sketch [18], an extremely low bitrate generative compression pipeline. To put the future work into perspective, we conclude with the limitations of the proposed coding method.

### A. CODING SCHEME AND EXAMPLES

We introduce a proof-of-concept coding algorithm based on the semantic properties  $\mathcal{P}_1$  and  $\mathcal{P}_2$  demonstrated on the CLIP-unCLIP codec. The proposed coding scheme is the same as the one proposed in Figure 7:

- We encode the images via the encoder  $e = q \circ f$ ;
- To prepare the quantization, we clamp the latent vectors to  $[-1, 1]$ ;
- We quantize the latent vectors to 1-bit per dimension;
- On the user side, they decode the quantized latent vector via the decoder  $d = g \circ \pi$ .

Note that we use of the projector  $\pi$  to ensure that the generated images are in  $\hat{\mathcal{L}} \subset \mathcal{L}$ . For this experiment, we fix the generated images size to  $768 \times 768$ , thus the bit per pixel (BPP) for this pipeline is fixed to  $\frac{768}{768^2} \simeq 0.0012$  BPP.

### B. STATE-OF-THE-ART COMPARISON

To compare our framework among the extremely low bitrate compression pipelines, we encoded the same images with the VVC intra coder [43], the current best standard image compression scheme. The compression is done at the highest possible Quantization Parameter (QP) to reach the same BPP magnitude. At QP 63, we reach an average BPP of 0.0045. This BPP value is still 4 times higher than the one proposed by our coding scheme. We also compare our pipeline to the Text+Sketch model [18]. This pipeline relies on a generative compression algorithm using a textual description of the images, with or without a sketch of the image, as side information for the generator. While the sketch

helps to reconstruct closer images to the input, it also adds a supplementary cost in terms of compression. To make a fair comparison, we compare our model to both of the modes.

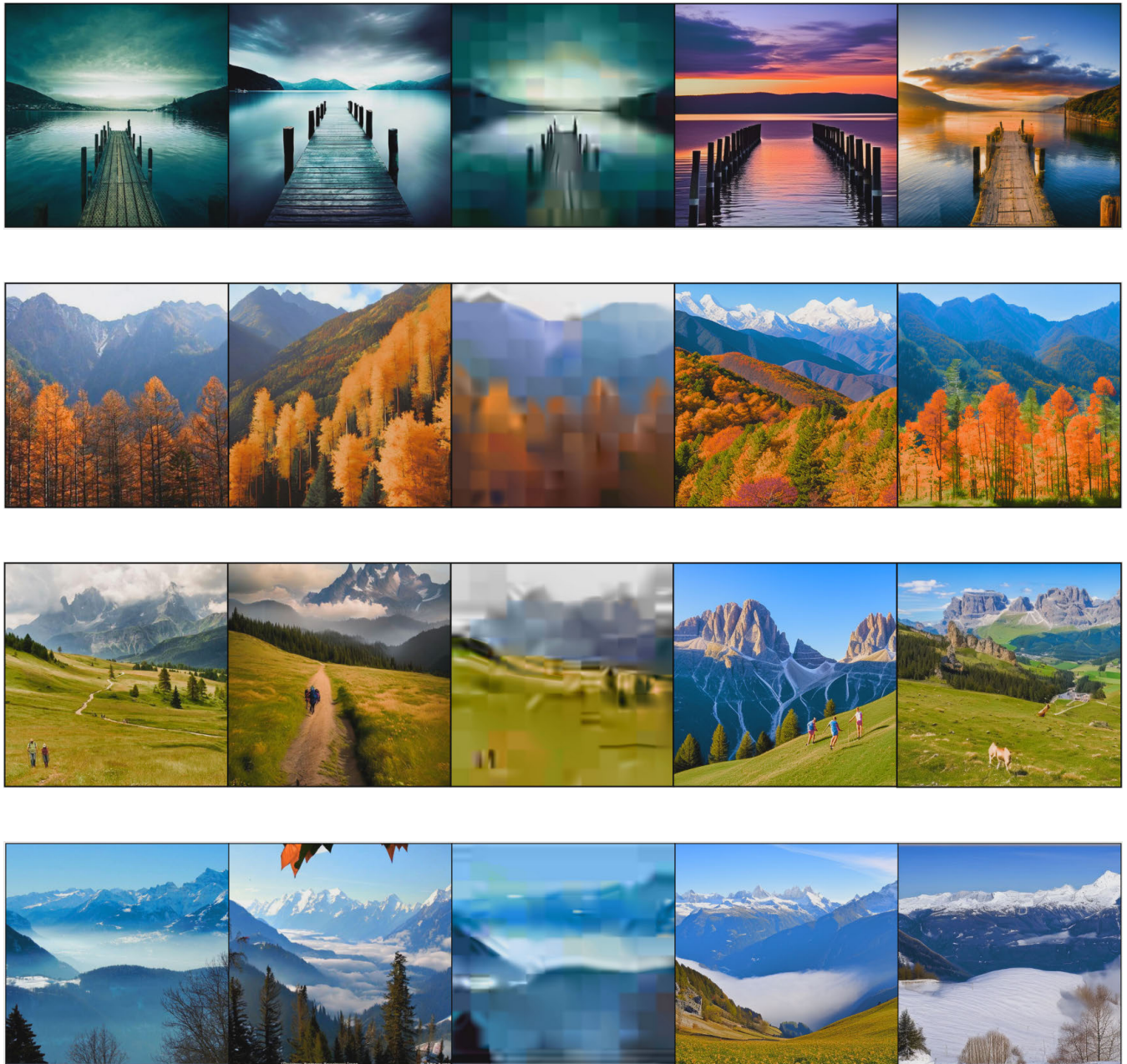
The comparisons of 20 compressed and decompressed images from Landscape (as it is the best dataset to work on, according to previous experiments) are presented in Table 3. We show that for each of the metrics used in this work (quality of the outputs and conservation of the inputs' semantics), our method performs better at extremely low bitrate than VVC, the classical coder. Indeed, if we observe the outputs of both methods, see Figure 12, we see that VVC produces poor-quality images from which one can barely recognize the inputs. We emphasize that VVC is one of the best current image encoders used for video compression. Yet, it has not been designed to be efficient at such low bitrates, hence the poor visual results. Note that this is also the case for all the other classical compression algorithms, as such extremely low compression rates are not considered in general. On the other hand, our codec, while encoding at a BPP 4 times lower, generates qualitative images that are semantically close to the different inputs. However, this codec still has room for improvements to be used as a complete compression framework, as discussed in the following.

Comparing to a generative compression framework, the differences between our pipeline and [18] are less obvious in terms of metrics. For the IQA metrics [18]'s image quality outperforms our model, and even outperforms the quality of natural images, see Tab. 1. Regarding the semantic metrics, our method competes with the other generative models. One can observe that one of the advantages of our method is that the generated images are visually closer to their respective inputs in terms of style (as one can observe in the first example) and in the conservation of some semantic details (as one can observe in the third example with the disappearance of the walking path or even the individuals). All in all, in the proposed framework, both the quality of the generated images and the conservation of the semantics are competitive with other generative compression pipelines while providing a BPP 3 to 20 times lower.

### C. LIMITATIONS

The results presented in the last experiments are promising: a CLIP (and unCLIP)-based compression framework is suited for extremely low bitrate compression. Both the quality of the generated images and the conservation of the semantics make the pipeline suitable for semantic image compression. However, we observe that the generated images are sometimes far from the inputs in terms of structural





**FIGURE 12.** Visual comparison of the different compression pipelines at extremely low bitrates. (Left to right for each row) Input image. Proposed method (0.0012BPP). VVC (0.0045 BPP). [18] without sketches (0.003 BPP). [18] with sketches (0.026 BPP).

organization (colors, place of the objects, themes, *etc.*). Thus, a possible enhancement of the proposed framework could be to add a bit of side information containing the structure of the input images in the compressed vectors. This information can be a color map of the inputs, as proposed in [15], or a sketch of the inputs, as proposed in [18]. While this side information may add to sticking closer to the inputs, it would come at the cost of a few extra bits for the compressed vectors. This would highlight a rate-distortion trade-off between the conservation of structural information between the inputs and the outputs,

and the size of the compressed latent vectors. A typical rate-distortion trade-off, as proposed by [27].

## VIII. CONCLUSION AND FUTURE WORK

### A. CONCLUSION

In this work, we demonstrated that CLIP can be used as a semantic image encoder for generative compression. Indeed, after showing that the relevant part of its latent space is a spherical shell, we proved two important properties for generative compression. First, the CLIP latent representations

are faithful to the descriptions of their respective inputs. Second, CLIP is resistant to harsh uniform quantization. These two properties allowed us to suggest a proof-of-concept generative compression pipeline for extremely low bitrate compression that even beats VVC with a BPP 4 times higher, both in terms of image quality and semantic preservation.

## B. FUTURE WORK

As the visual examples showed through all this work, the generated images are of good quality and semantically close to their respective inputs. However, the structural information (color, style, position, theme, *etc.*) are sometimes a bit off regarding the inputs. So, a possible way to continue this work is to find a way to encapsulate this non-semantic information to help the guidance during generation. For example, using a color map or a sketch as side information, as proposed in the state-of-the-art. However, this side information would come with a cost to the compression rate, and one would not be able to easily achieve the compression rate submitted in this work.

Another interesting study could also be to generate these results on other semantic encoders and generators outside the CLIP-unCLIP codec. Indeed, other foundation models can possibly fulfill better semantic properties for semantic generative compression. Moreover, one can look for a way to generalize or automate the proposed semantic properties in other foundation models.

## REFERENCES

- [1] B. Girod, E. Steinbach, and N. Färber, "Performance of the H.263 video compression standard," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 17, pp. 101–111, Nov. 1997.
- [2] I. E. Richardson, *The H.264 Advanced Video Compression Standard*. Hoboken, NJ, USA: Wiley, 2011.
- [3] J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1885–1898, Dec. 2012.
- [4] 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.
- [5] *Cisco Annual Internet Report (2018–2023) White Paper*, Cisco, San Jose, CA, USA, 2020, pp. 1–35.
- [6] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.
- [7] A. Harell, A. De Andrade, and I. V. Bajić, "Rate-distortion in image coding for machines," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2022, pp. 199–203.
- [8] J. J. Levandoski, P.-Å. Larson, and R. Stoica, "Identifying hot and cold data in main-memory databases," in *Proc. IEEE 29th Int. Conf. Data Eng. (ICDE)*, Apr. 2013, pp. 26–37.
- [9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [10] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Comput. Surv.*, vol. 56, no. 4, pp. 1–39, Apr. 2024.
- [11] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Extreme learned image compression with GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 2587–2590.
- [12] F. Pezone, O. Musa, G. Caire, and S. Barbarossa, "Semantic-preserving image coding based on conditional diffusion models," 2023, *arXiv:2310.15737*.
- [13] S. Barbarossa, D. Communiello, E. Grassucci, F. Pezone, S. Sardellitti, and P. Di Lorenzo, "Semantic communications based on adaptive generative models and information bottleneck," *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 36–41, Nov. 2023.
- [14] E. Grassucci, J. Park, S. Barbarossa, S.-L. Kim, J. Choi, and D. Communiello, "Generative AI meets semantic communication: Evolution and revolution of communication tasks," 2024, *arXiv:2401.06803*.
- [15] T. Bordin and T. Maugey, "Semantic based generative compression of images for extremely low bitrates," in *Proc. IEEE 25th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2023, pp. 1–6.
- [16] H. Nam, J. Park, J. Choi, M. Bennis, and S.-L. Kim, "Language-oriented communication with semantic coding and knowledge distillation for text-to-image generation," 2023, *arXiv:2309.11127*.
- [17] M. Careil, M. J. Muckley, J. Verbeek, and S. Lathuilière, "Towards image compression with perfect realism at ultra-low bitrates," in *Proc. 12th Int. Conf. Learn. Represent.*, 2023, pp. 1–21.
- [18] E. Lei, Y. B. Uslu, H. Hassani, and S. S. Bidokhti, "Text + sketch: Image compression at ultra low rates," 2023, *arXiv:2307.01944*.
- [19] N. Xue, Q. Mao, Z. Wang, Y. Zhang, and S. Ma, "Unifying generation and compression: Ultra-low bitrate image coding via multi-stage transformer," Tech. Rep., 2024.
- [20] C. Li, G. Lu, D. Feng, H. Wu, Z. Zhang, X. Liu, G. Zhai, W. Lin, and W. Zhang, "MISC: Ultra-low bitrate image semantic compression driven by large multimodal model," Tech. Rep., 2024.
- [21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *CoRR*, vol. abs/2103.00020, pp. 1–48, Jul. 2021.
- [22] M. Kwon, J. Jeong, and Y. Uh, "Diffusion models already have a semantic latent space," 2022, *arXiv:2210.10960*.
- [23] J. Dotzel, B. Kotb, J. Dotzel, M. Abdelfattah, and Z. Zhang, "Exploring the limits of semantic image compression at micro-bits per pixel," in *Proc. ICLR*, Vienna, Austria, 2024.
- [24] H. Xue, Y. Sun, B. Liu, J. Fu, R. Song, H. Li, and J. Luo, "CLIP-ViP: Adapting pre-trained image-text model to video-language representation alignment," 2022, *arXiv:2209.06430*.
- [25] D. Huang, F. Gao, X. Tao, Q. Du, and J. Lu, "Towards semantic communications: Deep learning-based image semantic coding," in *Proc. GLOBECOM*, Madrid, Spain, 2022.
- [26] E. Hoogeboom, E. Agustsson, F. Mentzer, L. Versari, G. Toderici, and L. Theis, "High-fidelity image compression with score-based generative models," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2023.
- [27] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," *CoRR*, vol. abs/1711.06077, pp. 1–18, Jun. 2017.
- [28] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," Tech. Rep., 2022.
- [29] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10684–10695.
- [30] Kodakt. (1999). *Kodak Lossless True Color Image Suite*. [Online]. Available: <https://r0k.us/graphics/kodak/>
- [31] M. Afifi, M. A. Brubaker, and M. S. Brown, "HistoGAN: Controlling colors of GAN-generated and real images via color histograms," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7937–7946.
- [32] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.
- [33] Y. Dalva, H. Pehlivan, C. Moran, Ö. I. Hatipoglu, and A. Dündar, "Face attribute editing with disentangled latent vectors," in *Proc. ECCV*, Tel Aviv, Israel, 2023.
- [34] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.



- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [36] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, Zurich, Switzerland, 2015.
- [37] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "CLIP-Score: A reference-free evaluation metric for image captioning," 2021, *arXiv:2104.08718*.
- [38] 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, 2017, pp. 1–12.
- [39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.0056, pp. 2818–2826, Jun. 2015.
- [40] J. Ke, Q. Wang, Y. Wang, P. Milanfar, and F. Yang, "MUSIQ: Multi-scale image quality transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5128–5137.
- [41] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 36–47, Jan. 2020.
- [42] ITU. (1993). *ISO/IEC 10918-1: 1993(e) CCIT Recommendation T.81*. [Online]. Available: <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>
- [43] A. Wiecekowsky, J. Brandenburg, T. Hinz, C. Bartnik, V. George, G. Hege, C. Helmrich, A. Henkel, C. Lehmann, C. Stoffers, I. Zupancic, B. Bross, and D. Marpe, "VVenC: An open and optimized VVC encoder implementation," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2021, pp. 1–2.



**TOM BACHARD** received the M.Sc. degree in theoretical computer science from École Normale Supérieure de Rennes, Université Rennes 1, Rennes, France, in 2021. He is currently pursuing the Ph.D. degree with Inria, Rennes, under the supervision of Thomas Maugey. His research interests include signal processing, image compression, and deep learning.



**THOMAS MAUGEY** (Member, IEEE) received the M.Sc. degree in fundamental and applied mathematics from Supélec, Université Paul Verlaine, Metz, France, in 2007, and the Ph.D. degree in image and signal processing from TELECOM ParisTech, Paris, France, in 2010. From October 2010 to October 2014, he was a Postdoctoral Researcher with the Signal Processing Laboratory (LTS4), Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. From November 2014 to October 2023, he was a Research Scientist with Inria. Since October 2023, he has been the Research Director at Inria. His research interests include image and video processing/compression and graph-based signal processing. He serves as an Associate Editor for *EURASIP Journal on Advances in Signal Processing* and *IEEE SIGNAL PROCESSING LETTERS*.

• • •