

Lossy Neural Compression for Geospatial Analytics: A Review

Carlos Gomes¹, Isabelle Wittmann¹, Damien Robert⁶, Johannes Jakubik¹, Tim Reichelt⁵, Stefano Maurogiovanni^{4,9}, Rikard Vinge², Jonas Hurst⁷, Erik Scheurer⁴, Rocco Sedona⁴, Thomas Brunschwiler¹, Stefan Kesselheim⁴, Matej Batič³, Philip Stier⁵, Jan Dirk Wegner⁶, Gabriele Cavallaro^{4,9}, Edzer Pebesma⁷, Michael Marszalek², Miguel A. Belenguer-Plomer⁸, Kennedy Adriko^{4,9}, Paolo Fraccaro¹, Romeo Kienzler¹, Rania Briq⁴, Sabrina Benassou⁴, Michele Lazzarini⁸, and Conrad M Albrecht²

¹IBM Research - Europe, Switzerland

²German Aerospace Center, Germany

³Sinergise Solutions, Slovenia

⁴Forschungszentrum Jülich, Germany

⁵University of Oxford, United Kingdom

⁶University of Zurich, Switzerland

⁷University of Münster, Germany

⁸European Union Satellite Centre, Spain

⁹University of Iceland, Iceland

Abstract—Over the past decades, there has been an explosion in the amount of available Earth Observation (EO) data. The unprecedented coverage of the Earth’s surface and atmosphere by satellite imagery has resulted in large volumes of data that must be transmitted to ground stations, stored in data centers, and distributed to end users. Modern Earth System Models (ESMs) face similar challenges, operating at high spatial and temporal resolutions, producing petabytes of data per simulated day.

Data compression has gained relevance over the past decade, with neural compression (NC) emerging from deep learning and information theory, making EO data and ESM outputs ideal candidates due to their abundance of unlabeled data.

In this review, we outline recent developments in NC applied to geospatial data. We introduce the fundamental concepts of NC including seminal works in its traditional applications to image and video compression domains with focus on lossy compression. We discuss the unique characteristics of EO and ESM data, contrasting them with “natural images”, and explain the additional challenges and opportunities they present. Additionally, we review current

applications of NC across various EO modalities and explore the limited efforts in ESM compression to date. The advent of self-supervised learning (SSL) and foundation models (FM) has advanced methods to efficiently distill representations from vast unlabeled data. We connect these developments to NC for EO, highlighting the similarities between the two fields and elaborate on the potential of transferring compressed feature representations for machine-to-machine communication. Based on insights drawn from this review, we devise future directions relevant to applications in EO and ESM.

Index Terms—Earth Observation, Earth System Models, Neural Compression, Geospatial Analytics

CONTENTS

I	Introduction	3			
I-A	Motivation & Approach	3			
I-B	Compression	5			
	I-B1 Neural vs. Traditional				
	Compression	5			
	I-B2 Neural Compression for				
	Imagery	6			
I-C	Outline	7			
II	(Lossy) Neural Compression	8			
II-A	Background	8			
	II-A1 Lossy Compression	8			
	II-A2 Transform Coding	10			
II-B	Compression Taxonomy	12			
II-C	Methods in Neural Compression	12			
	II-C1 Transforms	12			
	II-C2 Quantization Strategies	13			
	II-C3 Entropy Models	14			
	II-C4 Optimization Objectives	16			
III	Neural Compression for Remote Sensing	17			
III-A	Challenges in Compressing Re-				
	remote Sensing Data	17			
	III-A1 Data Characteristics	17			
	III-A2 Data Acquisition and				
	Application	18			
III-B	Classification of Compression				
	Methodologies	19			
	III-B1 Traditional Approaches	19			
	III-B2 Neural Approaches	19			
III-C	Summary	23			
IV	Neural Compression for Climate Data	24			
IV-A	Challenges	25			
	IV-A1 Data Characteristics	25			
	IV-A2 Data Acquisition and				
	Application	25			
IV-B	Classification of Compression				
	Methodologies	26			
	IV-B1 Traditional Approaches	26			
	IV-B2 Neural Approaches	26			
IV-C	Summary	27			
V	Neural Compression: Implementation & Application	27			
V-A	Neural Compression for Geospatial				
	Analytics Platforms	27			
V-B	Cost- and Energy-Efficiency & La-				
	tency	29			
V-C	Democratization for Applications	32			
	V-C1 Global Vegetation				
	Structure Analysis	32			
	V-C2 Ship Detection for				
	Maritime Awareness	33			
			V-C3 Climate and Air Pol-		
			lution Prediction	33	
			V-C4 Early Crop Stress and		
			Yield Prediction	34	
VI	Perspectives & Recommendations	34			
	Acknowledgment	36			

I. INTRODUCTION

A. Motivation & Approach

EARTH Observation (EO) is the process of capturing data about the Earth's surface and atmosphere, carried out through instruments on board of satellites, airborne vehicles, ships, or through ground stations. Owing to their constant activity and wide coverage, the bulk of these data is produced by satellites, with the Copernicus system alone delivering a reported 16 terabytes of data per day [1]. As large as this amount of data already is, it is only set to increase, with over 100 new EO satellites launched in 2021, over 150 in 2022, and almost 250 in 2023 [2]. Earth System Models (ESMs) simulate the evolution of components of the Earth system to predict future climate and air pollution. These systems also produce large volumes of data, and, driven by the need for higher resolutions to predict increasingly complex phenomena, these volumes are certain to increase with next-generation ESMs. In fact, data output and storage have already become a major bottleneck for high-resolution climate modeling.

Among others, this situation sparked projects to utilize AI methodologies to come at rescue, cf. ESA's MajorTOM dataset with embeddings [3], ESA's project CORSA¹, the EU Horizon project *Embed2Scale*², as well as the *Earth Index*³ solution developed by the start-up Earth Genome. In general, and beyond EO and ESM, a vibrant and innovative start-up community for NC emerged in recent years, with notable examples such as Deep Render⁴.

The importance of accessing EO and ESM data for analysis cannot be overstated. ESMs are vital for predicting the course of climate change and its potential impacts across the Earth. For the next generation of high-resolution climate models, it is currently no longer possible to store the full range of simulated parameters. Hence, only a small subset is stored for future analysis, making computationally expensive re-runs necessary.

¹<https://remotesensing.vito.be/about>

²<https://embed2scale.eu/vision-strategy>

³<https://www.earthgenome.org/earth-index>

⁴<https://deeprender.ai>

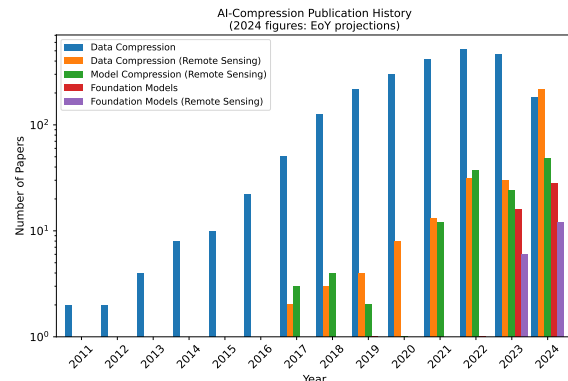


Fig. 1: Literature on neural compression (NC) summarized for the past 15 years. We plot separate bars for remote sensing (RS) and recent developments in foundation model (FM) methodology. Data Source: Queries to the *Web of Science* [4].

To best utilize EO data, it is critical to store it for long-term usage, enabling comparative studies, as well as distribute it to end users effectively. There are two main bottlenecks in doing this:

- 1) Bandwidth between satellite and base stations required for transferring the observations for storage and analysis on the ground is limited. This is a well-known problem in the community, referred to as the data down-link bottleneck [5]. While close-to-lossless transmission is desired for comprehensive EO data archives, dedicated (nano-)satellites with focus on specialized applications open the opportunity to implement lossy compression for near real-time geospatial analytics.
- 2) Storage of such a volume of data on a physical medium is expensive, and its transfer through a network (typically from data center to research institutions distributed globally) causes egress costs and delays research.

Traditional data compression methods, mostly based on the JPEG2000 standard [7], have long been used to reduce the required storage. However, the growth in data volume demands new approaches to more efficiently store only those aspects of the data that are required for their reconstruction or

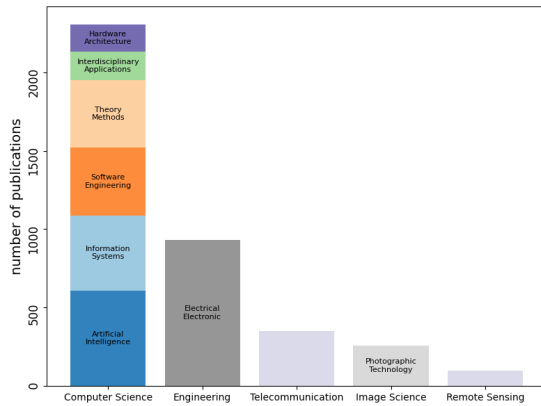


Fig. 2: Domain-specific shares in publications in NC methodologies from years 2000 through 2023 for the ten biggest (sub)categories as per *Web of Science* [6]. The stacked subcategories for *Computer Science* have been ordered from largest to smallest bottom-up.

usage for geospatial analytics.

Neural compression (NC) emerges as a natural candidate to improve compression algorithms. It has been shown in the literature to outperform traditional, hand-designed compression algorithms on curated datasets across several fields such as image compression [8], video compression [9], and audio compression [10]. NC uses deep neural networks to perform data compression. It seeks to identify and efficiently store the critical aspects of the data, discarding irrelevant or repetitive information, by learning directly from the data. Here, we associate the notion of *irrelevant information* with lossy compression. In contrast to traditional approaches, loss of information is not set by an external parameter in NC. Relevant information is inherently learned from the data distribution itself. NC algorithms are mostly based on autoencoders [11] and do not require labels to be trained. However, they do rely on large datasets which representatively sample the underlying distribution of the data. This creates an inviting environment for adapting and applying NC techniques to EO and ESM. *Neural Compression for Geospatial Analytics* embraces computational

algorithms employing artificial neural networks to reduce the storage required to digitize geospatial data while comprehending its (relevant) information content. The present review focuses on this approach, aiming to foster research on NC for EO and ESM data. Specifically, we focus on lossy NC, where higher rates of compression can be obtained as some loss of information is allowed to occur.

Fig. 1 reveals the historical growth in popularity of NC in research publications. The plot also demonstrates that applying NC to RS developed since 2017—with a lag of about 6 years to general NC. However, Fig. 2 illustrates that it is still a relatively unexplored topic in remote sensing (RS), making up only 3% of publications in NC methodologies.

Foundation Models (FMs) became an emergent and related topic to NC, recently (cf. Fig. 1): large pretrained neural networks that seek to learn embedding spaces can be leveraged for multiple downstream tasks in a domain. In contrast to NC, FMs have been quickly adopted in RS domains [12], and a community forms around FMs for ESM [13]. Despite not being directly optimized to compress data, FMs share similarities with NC, with both being trained on very large unlabeled datasets to extract fundamental features from data.

The emergence of FMs represents a fundamental paradigm shift in deep learning. This shift has primarily resulted from three factors: (1) the availability of vast amounts of *unlabeled (geospatial) data*, (2) the emergence of self-supervised learning (SSL)⁵ in RS [14], and (3) a significant increase of computational power enabling large models to be trained at scale [15]. The absence of human-annotated labels in such large-scale training processes results in *task-agnostic* deep learning models that are generally referred to as FMs. Within a couple years, a zoo of models for various satellite data modalities and weather models has been developed, [16–24].

Empirical evidence demonstrates several improved capabilities of FMs for RS [32] and atmospheric modeling compared to supervised deep

⁵a concept that allows deep learning models to learn from unlabeled data

Aspect	Neural Compression	Traditional Compression
Approach	neural networks & machine learning [25]	predefined algorithms
Adaptability	learn from data, flexible [26]	fixed algorithms tailored to data format
Performance	potentially higher compression efficiency [27]	consistent performance
Comput. Complexity	high [28]	low
Lossy vs. Lossless	handles both [29]	design specialized to either
Costs (today)	high (active research, model training) [30]	low (established field, algorithms optimized)
Use Cases (today)	image/video compression, specialized tasks [31]	general-purpose

TABLE I: Comparing key aspects of compression for traditional and neural approaches. For NC we picked a 2024 sample publication. A more comprehensive list of relevant papers for lossy NC provides Table II and Fig. 21

learning models. For example, recent work has shown a significant acceleration in solving tasks in RS based on pretrained large-scale SSL models (e.g., [18]). In addition, fine-tuning pretrained FMs can significantly reduce the required amount of task-specific, often human-annotated, data, thereby improving data efficiency compared to traditional supervised deep learning [16]. Furthermore, recent research demonstrated that FMs for RS benefit from their pre-training when generalizing to other, unseen geographical regions. For example, models have performed better on segmenting flood extents in regions that have not been part of the pre-training data compared to other supervised deep learning approaches [33].

Despite various benefits resulting from FMs for RS of land and atmosphere, several challenges remain—especially regarding their significant data consumption, creating significant data transmission and storage bottlenecks. While FMs can be seen as performing a certain compression of the raw data in the embedding space, those embeddings are still relatively large. This makes the NC of embeddings particularly interesting [34, 35], especially in an upcoming era of large growth in data generation. We discuss FMs in their combination with NC to generate compressed features in Section II and Section III.

B. Compression

Before diving into a more formal approach in Section II-A2, we share an intuitive picture of compression. When it comes to NC, it is worth noting that our review concerns elements of data

compression. There exists an entire field of *model compression* [36] that revolves around pruning the size of artificial neural networks. We will not touch this domain here.

1) *Neural vs. Traditional Compression:* Compression algorithms aim to encode a signal in as few symbols (and ultimately bits) as possible. These algorithms are core enablers of modern computing infrastructures, allowing different types of data to be stored and transmitted without prohibitively large costs.

Traditionally, compression algorithms (*codecs*) consist of a pipeline of components that have been hand-engineered by experts to compress signals of a specific type. We denote them as engineered *by hand* in the sense that they are not the direct result of data-driven algorithms, but rather human-crafted. These rely on signal processing and information theory, with each codec requiring a large number of human hours of work, often organized through consortia. Currently, virtually all codecs seeing widespread use belong to this category, such as MP3 [37], H.264 [38], HEVC [39] or JPEG [40], to name only a few.

Learning-based methods, including artificial neural networks, have been explored for compression since at least the 1980s [41]. However, with the recent emergence of deep learning, promising results [42, 43] led to a growth of research in the field of *neural data compression*. The main premise of NC is to replace traditionally hand-designed components of codecs with data-driven modules, usually neural networks, typically learned over a large representative dataset. Ultimately, not

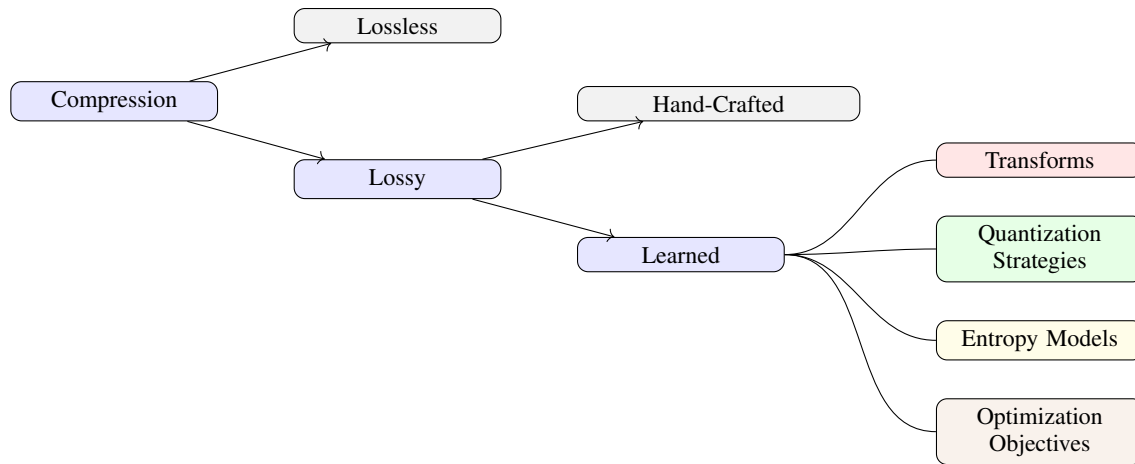


Fig. 3: Taxonomy of compression methods. This review focuses on the family of methods defined by following the blue nodes. We group all variations within this family into transforms, quantization strategies, entropy models, and optimization objectives.

just individual components are replaced, but rather the whole pipeline, leading to a codec that is learned fully end-to-end. Table I provides a high-level overview of pros and cons for both neural and traditional compression methods.

Two main benefits emerge from learned approaches. The first is the reduction in expensive expert hours required for elaborating algorithms, relying on data-driven processes to determine the transformations applied to the data. By modifying the loss function used to train the neural network, the codec can explicitly prioritize different aspects of the data, depending on its type and use case, as opposed to manually tuning the parameters of different components in a traditional codec.

The second is the potential for improved compression, in particular, due to the joint optimization of all learned components. Especially when the domain of data is known *a priori* (e.g. optical imagery from satellites) and fixed, a neural codec can specialize to that domain simply through the design of its training dataset, granting it an advantage compared to traditional codecs designed to offer stable performance across many domains.

Despite the complexity of modern hand-designed codecs, compression algorithms can essentially

achieve their goal in two main ways, both of which deep learning proposes to improve.

Encoding the signal using fewer symbols: An accurate model of the distribution of the data (in particular one that takes into account the context surrounding a given symbol) is a crucial building block to be able to cleverly encode data. Leveraging neural networks allows us to learn complex models of the underlying data distribution, leading to more effective encoding schemes.

Allowing for some loss of information: In lossy compression, some parts of a signal may be deemed as unimportant or too costly to encode and thus may be dropped, leading to a potentially large reduction in the length of the encoded signal. Understanding which parts of the signal to drop to minimize the impact on its reconstruction is critical in designing such algorithms. By leveraging deep neural networks to learn the structure of the data, better trade-offs between message length and reconstruction quality can be discovered.

2) *Neural Compression for Imagery:* As introduced above, we distinguish traditional and NC methods. While the former easily allows to design algorithms with perfect reconstruction (*Lossless*), the latter utilizes deep neural networks with millions

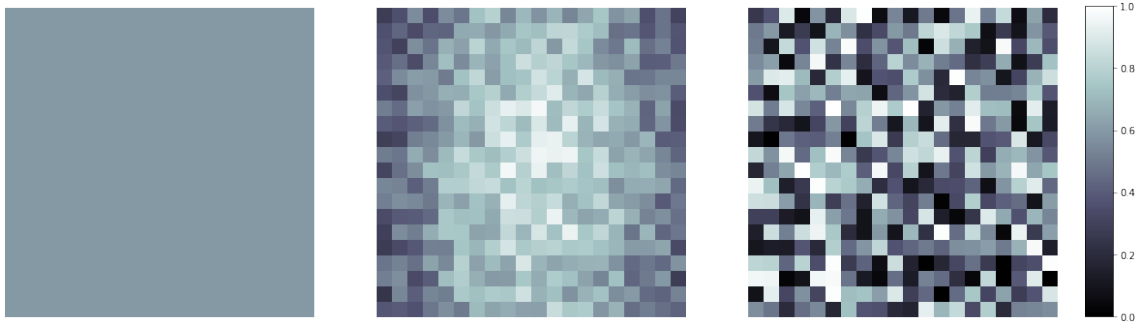


Fig. 4: Image compression from the perspective of pixel correlations, **Left to Right**: constant image, correlated noise, uncorrelated noise. A detailed description is provided in the main text.

to billions of learnt parameters that are, as of today, black box models hard to explain (*Learned*). Correspondingly, NC is primarily lossy; although there also exist initial approaches of designing lossless codecs with neural approaches [29]. Figure 3 summarizes our taxonomy categorizing NC (*Learned*) further as detailed in Section II-B.

Data compression exploits knowledge about the distribution $p(\mathbf{x})$ of data $\mathbf{x}$ when encoding a sequence of such data. Figure 4 illustrates the concept of compressibility of image data for an intuition on lossy compression. Let us consider the sequence of N data points being generated from a gray-scale image ($\mathbf{x}_i = \text{const} \in [0, 1]$) of dimensions $\sqrt{N} \times \sqrt{N}$ read out row-by-row (or column-by-column). In the extreme case of all pixels having the same constant value (Fig. 4, left), a neural network $f(\mathbf{x}) = z$ can learn to compress the input down to a single scalar z in feature space without loss. All image pixels are highly correlated by assuming exactly the same grayscale value. The entropy of the image is zero. On the other end of the spectrum, we may encounter random images (Fig. 4, right) where every single pixel is completely uncorrelated to all the other ones. Any compression to $N' < N$ symbols will imply information loss. Therefore, dimensional reduction is impossible.

Notice that the ability to losslessly compress is related to the smoothness (or sharpness) of an image. Certainly, if neighboring pixels are totally uncorrelated, on average, their random grayscale

value generates sharp contrasts. The introduction of correlation smoothens and causes the emergence of recognizable patterns, as exemplified by Fig. 4, center: A compressor may use the pattern that pixels towards the center of the image tend to become more bright compared to the boundary regions. In contrast to hand-crafted base functions—e.g., planar waves (Fourier transform) or wavelets (JPEG2000 encoder) [44]—NC bears the potential to automatically learn complex correlation patterns in images (and their associated base functions) from the data itself.

C. Outline

The goal of this review is to introduce the current status of *Neural Compression (NC) for Earth Observation (EO) and Earth System Modeling (ESM) data* with a focus on lossy image and video compression. We target an academic audience while staying as self-consistent as possible. More specifically,

- Section II will provide relevant background and an overview of state-of-the-art NC for lossy image and video compression,
- Section III continues by exploring NC in EO whereas Section IV is dedicated to the compression of the outputs from ESMs, with both sections laying out corresponding challenges in applying NC.

- Section V provides an introduction on how NC for geospatial analytics may integrate with corresponding data platforms for implementation.

Using concrete examples, we discuss how this integration democratizes geospatial applications in domains such as global vegetation monitoring, maritime awareness, climate modeling, and agriculture management. We close our review in Section VI, highlighting relevant future directions for the field of NC for geospatial analytics.

II. (LOSSY) NEURAL COMPRESSION

We begin with an introduction to lossy compression, presenting NC as an extension of transform coding. We then provide an overview of the NC literature, propose a taxonomy for navigating the field, and detail works most relevant to our geospatial focus. For a more theoretical and in-depth introduction to the field, we refer readers to Yang *et al.* [25].

A. Background

For the sake of simplicity, let us assume a finite set of data we wish to transmit, with each independent element indexed by $i = 1 \dots M$. In a sequence of data of length N , we count the frequency n_i of data point $\mathbf{x}_i$ such that $N = \sum_i n_i$ or equivalently $\sum_i p_i = 1$. The corresponding (Shannon) entropy, which quantifies the amount of information (or the uncertainty) associated with the data,

$$H(\mathbf{p} = (p_1, \dots, p_M)) = - \sum_i p_i \log p_i \quad (1)$$

is maximal for the uniform distribution when $p_i = 1/M = p$, $i = 1 \dots M$, i.e. all data points $\mathbf{x}_i$ are equally likely. A simple encoding scheme for M distinct data points can be represented as a *key-value* (lookup) table T where the *values* refer to the $\mathbf{x}_i$, and the *keys* represent a binary sequence of bits. Using a simple encoding scheme, we can number all M data points using $b = \log M (= -\log p)$ bits for each key, such that $2^b = M$. However, we can notice that in the extreme case where $p_I = 1$ for a fixed I and $p_{i \neq I} = 0$ the entropy H drops to zero. Given that we certainly know we are just

transmitting a single data point N times, we do not need to transmit any bit, i.e. we can use a different encoding scheme where we use 0 bits to encode data point I such that $b = 0 = \log 1 (= -\log p_I)$. If we more rigorously generalize this thought to include cases outside these two extremes, Shannon's source coding theorem [45] tells us that in the optimal encoding scheme, the length of the bit sequence indexing each data point $\mathbf{x}_i$ should be inversely related to its frequency n_i . Precisely,

$$b_i = \log \frac{1}{n_i} = -\log p_i \quad . \quad (2)$$

In this more generic scenario, we observe that the entropy H expresses the average bit length required to encode $\mathbf{x}_i$ based on its probability p_i :

$$H(\mathbf{p}) = \sum_i p_i b_i = \mathbb{E}[b] \quad . \quad (3)$$

Using the table T with keys whose lengths follow Shannon's source coding theorem (rounding up to the nearest integer) enables the encoding and decoding of the sequence of N data points without any loss—it is lossless compression.

NC replaces the lookup table T with a separate encoder artificial neural network f and a decoder artificial neural network g . The finite size M is enforced by a discretization step (*quantization*) in embedding/ feature space $\mathbf{z} = f(\mathbf{x})$. For the purpose of illustration $g \circ f$ may represent an autoencoder (AE), or its probabilistic sibling, a *Variational Autoencoder* (VAE) [46]. In case such a VAE is trained alongside with K learnable feature vectors $\{\mathbf{z}^{(k)}\}_{k=1 \dots K}$ in embedding space such that every encoding $\mathbf{z} = f(\mathbf{x})$ is *snapped* to one of these vectors (quantization step), the resulting neural compressor is termed *Vector Quantized-VAE* (VQ-VAE) [47].

1) *Lossy Compression*: Lossy compression techniques reduce data size by allowing some loss of information. This section outlines the typical steps involved in a compression pipeline, as illustrated in Fig. 5. In lossy compression, perfect signal reconstruction is not required. Instead, an approximate reconstruction is acceptable, enabling higher compression ratios. Let S be a random variable, known as the source, producing symbols

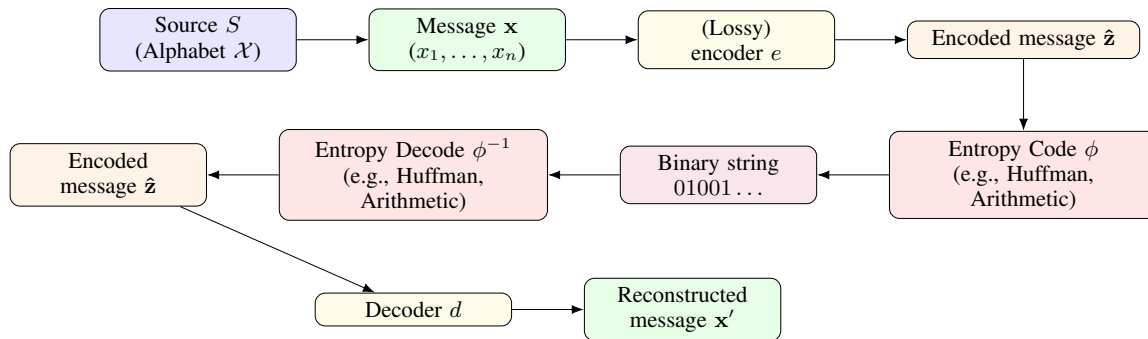


Fig. 5: Depiction of typical compression pipeline.

that form strings $\mathbf{x} := (x_1, x_2, \dots, x_n)$ from an alphabet $\mathcal{X}$. We refer to $\mathbf{x}$ as the signal or message to be compressed. To compress $\mathbf{x}$, we require an encoder e that maps it to a string of symbols $\hat{\mathbf{z}} := (z_1, z_2, \dots, z_m)$ from a different alphabet $\mathcal{Z}$. A decoder d then seeks to reconstruct $\mathbf{x}$ as $d(\hat{\mathbf{z}}) = \mathbf{x}'$. To efficiently transmit $\mathbf{z}$, the encoder and decoder agree on an encoding scheme ϕ , known as an *entropy code*, which losslessly encodes $\hat{\mathbf{z}}$ into a binary string. Examples of entropy coding schemes include *Huffman Coding* [48] or *Arithmetic Coding* [49]. Intuitively, ϕ assigns shorter binary codes to frequently occurring symbols, to reduce the overall length of the encoded message without losing information.

Together, the encoder e , decoder d , and entropy code ϕ constitute a *codec*. The objective of e is to minimize the length of the encoded string $\phi(\mathbf{z})$, known as the *code-length*, while minimizing the information loss in the reconstructed signal $\mathbf{x}'$. This introduces a trade-off between compression rate and the distortion of the reconstruction.

We can express this trade-off as a Lagrangian optimization problem.

$$\min \lambda D + R. \quad (4)$$

Here, R represents the rate, defined as the expected number of bits required to transmit a data point, and D denotes the distortion, the expected error between a data point and its reconstruction. By varying λ , different codecs achieve various trade-offs between

R and D , which can be visualized using a Rate-Distortion plot.

The rate term R is grounded in Shannon's source coding theorem [45], which provides a lower bound on the number of bits required to encode $\hat{\mathbf{z}}$ losslessly:

$$-\log_2 p(\hat{\mathbf{z}}), \quad (5)$$

Here, p is the probability mass function of the distribution of $\hat{\mathbf{z}}$. The entropy, defined as the expectation of this quantity:

$$\mathbb{E}[-\log_2 p(\hat{\mathbf{z}})], \quad (6)$$

measures the average minimum number of bits required per symbol for optimal encoding. It characterizes how difficult it is to compress samples drawn from p .

While the entropy value represents the theoretical lower bound on the rate, it is not the actual number of bits achieved by a specific algorithm (known as the *operational rate*). However, encoding schemes such as arithmetic coding [49] can achieve operational rates very close to this theoretical limit [50]. This makes entropy a good approximation which is useful in optimization frameworks to compute gradients for loss functions. It is important to note, that the accuracy of this approximation depends on the quality of the modeled distribution p .

The distortion term D relies on an underlying error function $\rho(\mathbf{x}, \mathbf{x}')$, which quantifies the difference between the input and reconstruction. Common choices for image and video compression are

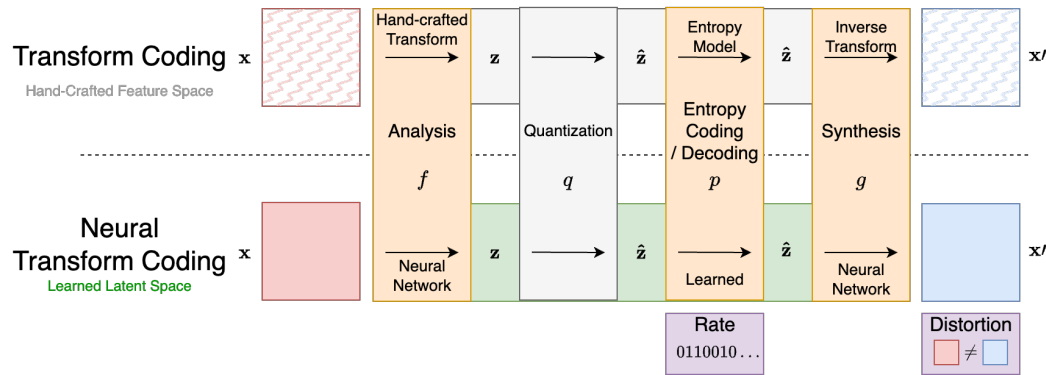


Fig. 6: Illustration of the transform coding framework and its adaptation to neural transform coding. Components within orange boxes are replaced with learned counterparts. Terms within purple boxes are used in the loss.

the Mean Squared Error (MSE) and the Structural Similarity Index Measure (SSIM) [51].

2) *Transform Coding*: Most NC methods can be viewed as learned, non-linear variations of the transform coding paradigm [52]. Transform coding, as illustrated in Figure 6, is fundamental to codecs like JPEG or HEVC. The encoder first applies a transform to the raw data, followed by quantization. This transform is designed to map the data to a space where it can be compressed more efficient.

Transform. In traditional compression, the transform f is typically a handcrafted, linear, and invertible mapping known as the *analysis transform*. Commonly used in image compression are the Fourier Transform [53] and the Discrete Wavelet Transform [54]. These transforms aim to decorrelate the input data to facilitate the quantization and entropy modeling steps that follow.

The input data x is often expressed as a vector whose coordinates are correlated. For example, adjacent pixels in natural images tend to have similar values. These correlations introduce redundancies in the input signal: knowing part of the signal allows one to predict other parts. Hence, discarding such correlations is desirable in a compression framework. To address this, a *transform* f is applied to map the data into new representation $z = f(x)$, where coordinates are less correlated and ideally independent. In NC, f is an artificial neural net-

work, trained to map x to an *embedding* z within a continuous latent space. Although ambiguous in the field of compression, this neural network f is often referred to as the *encoder network*. The ability to learn complex, non-linear transforms directly from dataset statistics puts NC approaches at a great advantage compared to traditional handcrafted methods. Figure 7 illustrates this advantage with a toy example.

Quantization. The output of the transform is embedded in a continuous latent space and must be quantized to allow compression with a finite number of bits. By quantization, we broadly mean any mapping from a continuous space to a discrete and countable set. Beyond this necessity, quantization introduces information loss into the compression process, and is thus also the desirable mechanism by which rate is traded for distortion. The chosen quantization method q is applied to z , resulting in $\hat{z} = q(z)$. A neural network-based transform f can learn to warp the embedding space to effectively manipulate which information is lost through quantization.

Entropy Coding. After quantization, the discrete representation $\hat{z}$ can be losslessly compressed using entropy coding. Assuming an encoding scheme that can approach the theoretical lower bound given by Shannon's source coding theorem, such as arithmetic coding [49], the efficiency of this com-

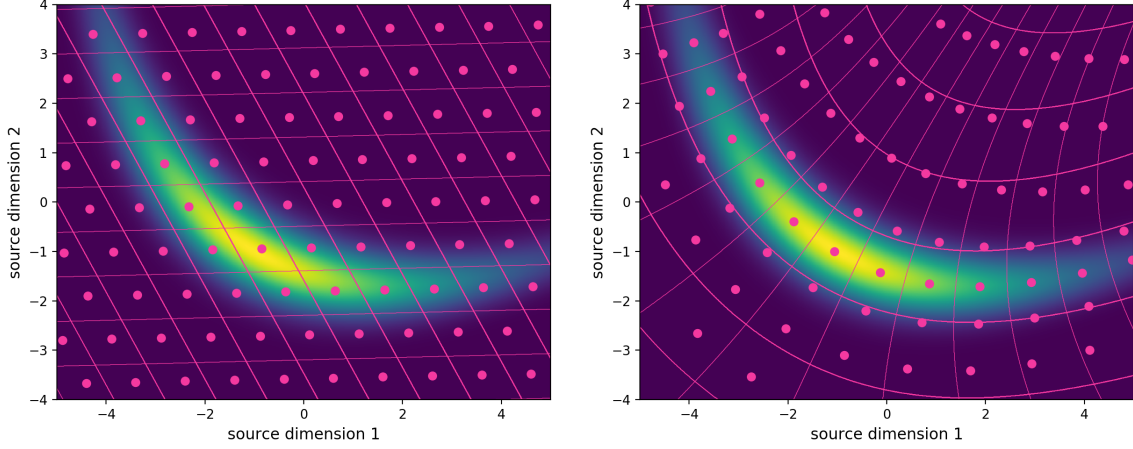


Fig. 7: Linear transform code (LTC, left), and nonlinear transform code (NTC, right) of a banana-shaped source distribution, both obtained by minimizing the rate–distortion Lagrangian. Lines represent quantization bin boundaries, while dots indicate code vectors. NTC adapts more closely to the source distribution, resulting in better compression performance compared to LTC. Figure and caption taken from [55].

pression is determined by the *entropy model* p' . Since the true data distribution p is unknown, it is modeled using p' . An accurate approximation p' is crucial for assigning shorter codes to more frequently occurring symbols, thereby minimizing the average length of the compressed representations. The closer p' matches the true distribution p , the nearer the final encoding length will be to the lower bound established by Shannon's theorem (Expression 5). In NC, the entropy model also takes on an additional role during training, by providing differentiable estimates of the bit cost for encoding a batch of data. This estimation is incorporated into the model's loss function, enabling end-to-end rate–distortion optimization. On the receiver side, the entropy decoding process reconstructs $\hat{\mathbf{z}}$ from the compressed binary string.

Inverse Transform. To reconstruct the data, the inverse transform g , often referred to as the *synthesis transform*, is applied to the quantized representation $\hat{\mathbf{z}}$. In NC, an analytical inverse of the encoder f is typically unavailable. Instead, a second neural network, the *decoder*, is trained to reconstruct the original data from $\hat{\mathbf{z}}$. This results in $\mathbf{x}' = g(\hat{\mathbf{z}})$, an approximation of the original input

$\mathbf{x}$, subject to some loss.

NC aims to optimize the Lagrangian from Expression 4 end-to-end using deep learning techniques. By defining f and g as neural networks, and flexibly modeling p , NC can learn non-linear transform functions and complex entropy models. This approach leads to superior rate–distortion performance compared to traditional compressors, as demonstrated for tasks such as image [8], video [9], audio [10], and 3D scene compression [56].

Two important aspects for optimizing the Lagrangian, which we postpone to Section II-C3, are the specific quantization methods used and how a continuous model p can be used to fit the resulting discrete distribution.

We can now express the complete loss function for a single input $\mathbf{x}$ as:

$$L(\mathbf{x}) = \underbrace{\lambda \cdot \rho \left(\mathbf{x}, \overbrace{g(q(f(\mathbf{x})))}^D \right)}_R - \underbrace{\log_2 p' \left(q(f(\mathbf{x})) \right)}_R.$$

For the distortion term D , only the encoder f and decoder g participate. The gradients of this loss will push the encoder to produce quantization-robust representations that the decoder can accurately reconstruct.

The rate term R , involving the entropy model p' and the encoder f , pushes the encoder to create compressible representations by minimizing the entropy of $\mathbf{z}$. The entropy model p' serves as an approximation of the true distribution p , aiming to assign high probability to $\hat{\mathbf{z}}$ to minimize the loss, under the constraint that it must be a valid probability density function. As p' only contributes to the loss through the entropy term, the gradients of its parameters with respect to the distortion term will be 0. Thus, using the same loss, f and g can be trained while jointly fitting p' to $\hat{\mathbf{z}}$.

B. Compression Taxonomy

To categorize compression approaches and research areas, we propose a framework for classifying compression techniques, as illustrated in Fig. 3. Within this framework, we distinguish lossy compression techniques, which are the focus of this work, from lossless approaches. Next, we further classify methods into explicitly engineered approaches and those that are learned from data. Most widely used compression algorithms today, such as JPEG [40], MP3 [37], and HEVC [39], fall into the explicitly engineered category. However, we focus on learned compression methods, which have consistently outperformed handcrafted approaches in rate-distortion metrics, demonstrating their potential for advancing the field.

Within this framework, we identify four primary axes of variation in learned compression methods:

- **Transforms:** The design and architecture of the encoder and decoder networks.
- **Quantization strategies:** How the continuous latent representations are discretized, and how quantization can be made compatible with end-to-end learning.
- **Entropy models:** The assumptions and implementation used to model the probability distribution of the latent representation.

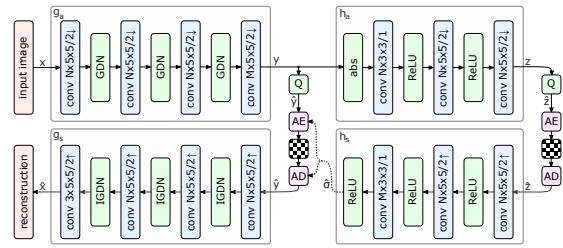


Fig. 8: Network architecture of the hyperprior model [57]. The left side shows an image autoencoder architecture, the right side corresponds to the autoencoder implementing the hyperprior. The factorized-prior model uses the identical architecture for the analysis and synthesis transforms g_a and g_s . Q represents quantization, and AE, AD represent arithmetic encoder and arithmetic decoder, respectively. Convolution parameters are denoted as: number of filters $\times$ kernel support height $\times$ kernel support width / down- or upsampling stride, where $\uparrow$ indicates upsampling and $\downarrow$ downsampling. N and M were chosen dependent on λ , with $N = 128$ and $M = 192$ for the 5 lower values, and $N = 192$ and $M = 320$ for the 3 higher values. Figure and caption taken from Ballé *et al.* [57].

- **Optimization objectives:** The optimization framework, particularly the choice of distortion measure used in the rate-distortion loss.

C. Methods in Neural Compression

The present section explores the functionalities, benefits, and limitations of different methodological approaches in the literature for each of the four axes of NC proposed in II-B. A summary can be found in Table II.

1) *Transforms:* In learned image and video compression, the synthesis and analysis transforms are typically implemented as two halves of a deep convolutional auto-encoder [58] as popularized by Ballé *et al.* [43] and Theis *et al.* [42].

The encoder gradually downsamples the spatial dimensions with a repeating pattern of convolutional layers and non-linear activations, while increasing the number of channels (embedding di-

mension). As shown in Fig. 8, the decoder mirrors this architecture to recover the original input.

Architectural innovations in deep learning and computer vision have introduced improvements to these transforms, such as the integration of attention mechanisms [59] and residual connections [60] into the network.

Building on the framework of neural transform coding, researchers have explored alternative architectures beyond convolutional networks. Indeed, unstructured data compression has leveraged fully connected feed-forward neural networks [55] and earlier works employed recurrent neural networks (RNNs) [61] as encoder and decoder architectures. More recent works have also explored the use of transformers [62, 63] and denoising diffusion models [64].

A distinct paradigm within NC that has also emerged is that of *Implicit Neural Representations* (INRs). Popularized in large part through their use in NeRF [65] for 3D scene representation, INRs have shown promise as an alternative way of representing and storing 3D geometry [66–68], audio [69], images [69], video [70], amongst others. INRs aim to represent any signal as an implicitly defined function. For instance, we may represent an image as a function $f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ mapping from pixel coordinates x and y to an RGB value. In practice, this function is learned by overfitting a neural network on a single input such that it can be recovered through inference on the network, essentially storing the input in the weights of the network. This representation allows for the leveraging of model compression literature to achieve general signal compression. It has successfully been employed for compressing 3D scenes [56], images [71, 72] and videos [73, 74], although their use together with an entropy penalty is still not ubiquitous, with many methods relying instead on more typical model compression techniques.

We note that INRs can also be interpreted as a pair of transforms. The encoder is replaced by the training process, mapping the input into the space of the neural network parameters, and the decoder is replaced by the forward pass of the

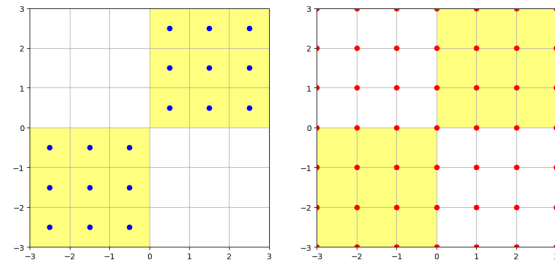


Fig. 9: Illustration of vector vs scalar quantization for a given data distribution, shown in yellow. **Left:** Vector quantization can efficiently cover the space by freely building a codebook of vectors, shown in blue. **Right:** Scalar quantization quantizes each dimension individually. This potentially leads to an inefficient coverage of the space, with many quantization points covering areas that are outside the distribution of the data. In this case, uniform quantization to the integers is shown.

network itself. The remaining 3 axes are then still fully applicable to INRs. INRs show competitive R-D performance as well as versatility in the types of signals they can encode. In particular, since they encode a single sample, they are not affected by the out-of-distribution problem that other NC methods may face and thus do not require a large dataset to be collected. Additionally, since decoding is simply a forward pass on the network, it typically can be fully parallelized, granting it great performance advantages in fields such as video [73, 74], image [75, 76], and NeRF [77] compression. However, the lengthy training process required for compressing each sample makes INRs impractical for deployment in many real-world applications.

2) *Quantization Strategies:* It can be shown that the optimal rate for a given distortion can be achieved through vector quantization [78]. In vector quantization, all dimensions of the space are jointly discretized, usually by mapping the given $\mathbf{z}$ to its nearest neighbor in a codebook. However, as the dimension of $\mathbf{z}$ grows, vector quantization becomes infeasible, with the curse of dimensionality requiring exponentially more entries in the codebook to optimally quantize the space, along with more data

and compute to optimize them.

As an alternative, the most popular form of quantization in the non-linear transform coding paradigm is scalar uniform quantization, as introduced by Ballé *et al.* [79] and Theis *et al.* [42]. Here, each dimension of the transformed data is quantized independently, typically by rounding each value to the nearest integer. This scheme can be seen as a constrained form of vector quantization where the grid is fixed and equal to the set of integers [25]. Despite its simplicity, scalar quantization is effective due to the flexibility of the non-linear transform, which can in essence warp this grid as desired. Figure 9 illustrates these two approaches. The main obstacle introduced by quantization is its non-differentiability, as backpropagating gradients is necessary for an end-to-end optimization of the network. Quantization has a gradient of 0 almost everywhere, preventing any components before it from receiving gradients. Two main techniques to address non-differentiability are:

Straight Through Estimator (STE): The STE [80] treats non-differentiable components as identity functions during backpropagation, fixing their gradient to 1 and thus allowing gradients to pass through unchanged. Theis *et al.* [42] applied this approach to the quantization function of NC models. For the forward pass, the quantization process remains unchanged. **Uniform Noise:** Ballé *et al.* [79] propose the replacement of quantization with additive uniform noise during training. In the case of quantization to the nearest integer, this noise has a range of $[-0.5, 0.5]$ and thus the same width as the quantization bins.

Empirically, the combination of both of these methods seems to be optimal for training [81], with STE used for calculating the distortion term and additive uniform noise used for the entropy term.

Beyond scalar quantization, other forms of quantization have been explored, despite being less popular. Early works employed binarization, reducing every element of $\mathbf{z}$ to two possible values [61, 82, 83]. Vector quantization (VQ) has also been successfully employed in neural transform coding [84, 85] with adaptations to mitigate its computational complexity problems. Promisingly, recent work in

generative modeling combines these approaches, leveraging a *Vector Quantized Generative Adversarial Network* (VQ-GAN) [86] for vector quantization and binarization to make this quantization more computationally feasible [87]. Despite the work focusing on video generation, it demonstrates competitive performance in video compression. VQ-GANs extend the *Vector Quantized Variational Autoencoder* (VQ-VAE) framework by employing an adversarial training strategy [88] to discriminate between real input images and the reconstructed outputs of the VQ-VAE decoder. Moreover, VQ-GANs enable the synthesis of high-resolution images (i.e., in the megapixel range) by modeling the learned (quantized) embeddings and their codebook through a transformer-based model. This interplay between GAN-enhanced autoencoder-based compression and transformer-based synthesis outperforms equivalent state-of-the-art approaches using plain autoencoders, thus opening the way for more context-rich compression strategies. Yang and Mandt [64] takes a different generative approach with diffusion models and outperforms a GAN benchmark on four metrics.

For INRs, quantization techniques often derive from general neural network compression, such as weight quantization or pruning [71, 73]. These methods can be applied after optimization or, often achieving better results, throughout training. Quantization-aware training [89] can be used to obtain INRs that are more robust to the error introduced by quantization, and finetuning after pruning can reduce its effect on distortion [73].

3) *Entropy Models:* The objective of the entropy model is to provide accurate approximations of $p(\hat{\mathbf{z}})$ for two purposes:

- a) Estimating the rate during training to be used in the loss.
- b) Entropy coding and decoding in operational use after the network has been trained.

Both of these uses impose a demand for reasonable computational efficiency. Additionally, differentiability is required to enable end-to-end training. A common approach to achieve this is to employ uniform quantization and define $p(\hat{\mathbf{z}})$ in terms of an underlying continuous density $p'(\hat{\mathbf{z}})$ [25]:

TABLE II: Collection of neural compression papers in this section, aligned by contributions along the axes described in Section II-B.

Axis	Approach	Papers
Transforms	CNN	[8, 42, 43, 57]
	RNN	[61]
	Transformer	[63, 90]
	Diffusion	[64]
	INR	[56, 71–77]
Quantization Strategies	Scalar Uniform Quantization	[42, 79]
	Binarization	[61, 82, 83]
	Vector Quantization	[84, 85]
	Weight Quantization	[56, 74]
Entropy Models	Fully Factorized	[57, 79]
	Hyperprior	[57, 91]
	Autoregressive and Transformer-based	[9, 56, 57, 74, 87, 92–95]
Optimization Objectives	Rate-Distortion-X	[96, 97]
	Downstream Embedding	[98, 99]
	Split Computing	[100, 101]

$$p(\hat{\mathbf{z}}) := \int_{[-0.5, 0.5]^n} p'(\hat{\mathbf{z}} + \mathbf{v}) d\mathbf{v}, \quad \forall \hat{\mathbf{z}} \in \mathbb{Z}^n.$$

The assumptions imposed on $p(\hat{\mathbf{z}})$ *a priori*, to make the integral tractable and ensure computational efficiency and differentiability, determine the possible architectures for the entropy model.

Fully factorized model. One of the stronger simplifying assumptions is that each element of $p(\hat{\mathbf{z}})$ is independent, allowing for a fully factorized model. Each marginal distribution can be modeled with varying degrees of complexity, from a simple parametric distribution such as a Gaussian or Laplacian to neural networks that estimate the cumulative distribution function (CDF) [57].

Hyperprior model. The assumption of full independence is likely too strong. An alternative approach to modeling dependencies between variables is to assume conditional independence given some other latent variable [102]. Ballé *et al.* [57] extend their fully-factorized model to a latent variable model by introducing a *hyperprior*:

$$\mathbf{z} \sim p(\mathbf{z}|\hat{\mathbf{h}}) \quad (7)$$

$$\mathbf{h} \sim p(\mathbf{h}) \quad (8)$$

Here, the hyperprior $p(\mathbf{h})$ is modeled as fully factorized, while $p(\mathbf{z}|\hat{\mathbf{h}})$ is conditioned on the quantized hyperprior. Typically, $p(\mathbf{z}|\hat{\mathbf{h}})$ is represented as a 0-mean Gaussian, with the standard deviation for each dimension of $\mathbf{z}$ derived from $\hat{\mathbf{h}}$. Although the hyperprior $\mathbf{h}$ introduces additional side information that must be transmitted, its size is negligible compared to $\hat{\mathbf{z}}$, and the added flexibility tends to significantly improve rate–distortion performance. Intuitively, the hyperprior allows the entropy model to adjust to the specific $\hat{\mathbf{z}}$ being transmitted.

Autoregressive and transformer-based models. More sophisticated entropy models capture complex dependencies between elements of $\hat{\mathbf{z}}$, often at the cost of added computational complexity. Autoregressive models [92] predict each element of $\hat{\mathbf{z}}$ based on the previously encoded elements, while transformer-based models leverage self-attention mechanisms to model complex relationships in the latent space of $\hat{\mathbf{z}}$, particularly in video compression. Mentzer *et al.* [9] greatly simplify video compression pipelines by relying on the modeling power of a transformer model, while Défossez *et al.* [93] use a small transformer to reason on quantized units for efficient audio compression. Recent advancements in large language models (LLMs) have inspired their exploration for lossless [94] as well

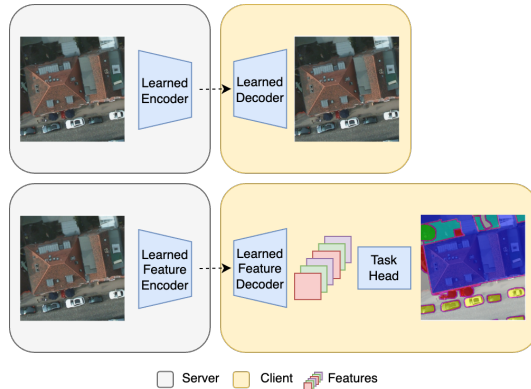


Fig. 10: Illustration of feature compression. **Top:** the usual scenario where a client wishes to reconstruct the input compressed by the server. **Bottom:** the server sends a compressed feature representation of the input, which the client can directly feed to a task-specific head (e.g. semantic segmentation in this case). Images from the Potsdam dataset [103].

as lossy [87] text, image, and video compression, demonstrating the potential of leveraging their predictive power and large context modeling for next-generation compression algorithms.

Implicit Neural Representations (INRs). In the case of INRs, the weights of the network themselves are treated as z . After quantization, the distribution of these weights is modeled by the entropy model. This method has shown competitive results in compressing 3D scene [56] and videos [74].

4) *Optimization Objectives:* Traditionally, NC methods aim to minimize distortion as perceived by humans, using loss functions like MSE and SSIM as proxies for human perception. Developing loss functions that more accurately reflect human perception remains an active area of research, both within NC and for generative visual models in general.

Recent studies have explored new trade-offs by reinterpreting the concept of distortion. Notably, the introduction of rate-distortion-perception [96] and rate-distortion-realism [97] frameworks allows for more nuanced optimization strategies.

The continuing growth and deployment of deep

learning algorithms in real-world applications introduces a new use case, which can be viewed as a further reinterpretation of distortion. This begs the question whether compression designed for human perception is the best choice when the end consumers of the data are algorithms (e.g. neural networks) instead of humans. From this perspective, recent works propose reframing distortion from an algorithmic point of view. As illustrated in Fig. 10, the goal is not necessarily to recover the original data such that it is minimally affected for a human observer, but rather to produce compressed feature representations that enable algorithms (e.g. classification, image segmentation, object detection) to perform well when using them as input [98]. Under this setting, the distortion metric is not a function of some reconstruction of the original data but rather based on the performance of such feature representations when fed to models for different downstream tasks.

A more general setting that makes the task somewhat more complex is that one may not know *a priori* the type of downstream tasks the embeddings will be used for, or labeled data for those tasks may not be available during pre-training. Instead, the process of learning general-purpose, compressible features must rely on proxy losses which may borrow from SSL [35] to identify which aspects of the data may safely be lost during compression without affecting downstream performance [99]. A similar idea has been applied to transfer data in the reverse direction, from an edge device collecting data to a powerful server where it can be analyzed, in a paradigm known as *Split Computing*. Matsubara *et al.* [100] combine *Knowledge Distillation* with NC to train a small encoder that can run on edge devices and produce compressible features that are fed to a larger network on a server with more compute resources. They demonstrate improved R-D performance compared to neural image compression methods that focus on reconstruction. Furtuanpey *et al.* [101] further develop the framework, conducting a thorough analysis of bottleneck placement within the network. They further introduce a saliency-guided loss and design blueprints for leveraging feature compression with different backbone archi-

textures, showing improved R-D performance.

Finally, while FMs for vision have been shown to generate embeddings that can generalize to several downstream tasks [32, 104, 105], the dimension of their output feature space may result in embeddings that are larger than the original data, making them impractical for storage or transmission. How to best generate such general-purpose compressed embeddings is an open question. However, a system capable of doing so could have a large-scale impact, democratizing both data and powerful models by enabling the widespread distribution of powerful, ready-to-use features. We deem this line of research to be particularly important as FMs become increasingly prevalent in EO [16, 32, 106, 107].

III. NEURAL COMPRESSION FOR REMOTE SENSING

Advances in RS technologies have led to an increasing number of EO satellite acquisitions with enhanced spatial resolution, broader spectral bands, and higher temporal frequency [2]. These data volumes present challenges for data transmission, storage, and processing [108, 109]. This section explores the application of NC to raster EO data from air- and space-borne instruments, drawing parallels to NC techniques used for natural images (Section II-C). We discuss the specific challenges of compressing RS data (III-A), review existing compression research (III-B), and explore future developments (III-C).

A. Challenges in Compressing Remote Sensing Data

RS data presents unique challenges and opportunities for compression, stemming from its distinctive data characteristics, acquisition constraints, and usage. These factors influence the design and applicability of existing image compression techniques [110].

1) Data Characteristics:

Spectral resolution. RS images often comprise multiple spectral bands beyond the visible range, such as near-infrared (NIR) and short-wave infrared (SWIR), enabling detailed surface and atmospheric

analysis [113]. While multispectral data typically captures a limited number of broad bands, hyperspectral data covers hundreds of narrow wavebands. This richer spectral dimension leads to larger data volumes and increased correlations between adjacent spectral bands [114, 115]. Compression techniques need to effectively exploit spectral correlations while avoiding excessive computational cost [116]. In contrast to optical sensor data, Synthetic Aperture Radar (SAR) captures the amplitude, phase, and polarization of radar signals reflected from the Earth's surface, making it suitable for applications like surface and moisture analysis [117].

Spatial resolution. The spatial resolution in RS varies depending on the physics of data acquisition. For example, the ESA Sentinel-2 satellite provides images with a spatial resolution of 10 to 60 meters per pixel [118]. Compared to other imaging domains, RS generally operates at lower spatial resolutions over large geographical areas. Consequently, individual pixels can contain highly relevant information for downstream tasks, and images exhibit complex textures with rich information [90, 119].

Temporal resolution. Most satellites capture data for specific geographic regions at regular intervals, generating time series that can support the monitoring of dynamic processes and environmental changes [120]. These successive images often exhibit temporal correlations, reflecting gradual landscape transformations and seasonal or weather-dependent variations. Unlike static image compression, where there are no temporal relationships, or video compression, where frames are closely spaced in time, RS imagery involves wider temporal gaps, often spanning several days between acquisitions [121].

Radiometric resolution. Radiometric resolution indicates a sensor's ability to measure the intensity of reflected radiation within a specified wavelength range. For instance, Sentinel-series satellites employ 12-bit resolution [118], providing higher precision than the 8-bit standard common in natural images. Greater precision enables more detailed measurements but increases the complexity for compression algorithms as it expands the input alphabet.

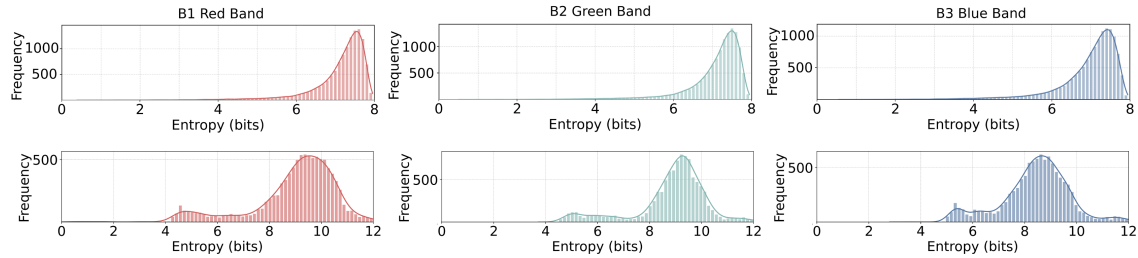


Fig. 11: Comparison of per-image entropy distributions for RGB bands across two datasets.

Top: ImageNet dataset – Random sample of 10,000 images [111].

Bottom: BigEarthNet dataset – Random sample of 10,000 Sentinel-2 images [112].

The varying resolutions in RS data make it difficult to define a 'typical' RS image. Designing compression algorithms that address the diverse spectral, spatial, temporal, and radiometric characteristics is inherently complex. As a result, compression methods in RS are often dataset-specific with limited generalizability across different types of RS data [122].

2) Data Acquisition and Application:

Data Downlink Bottleneck. A core practical challenge is the limited bandwidth to transmit satellite imagery to Earth [5, 123, 124]. To facilitate transmission, images are often compressed onboard using compression algorithms such as the Consultative Committee for Space Data Systems (CCSDS) standards [7].

Onboard Processing Limitations. In addition to bandwidth constraints, satellites have limited computational and storage resources, restricting the complexity of compression approaches that can be deployed onboard. Such constraints can limit the usability of NC techniques for onboard satellite applications.

Preservation of Critical Information. RS imagery is used in numerous scientific and operational applications, including environmental monitoring of above-ground biomass [125], agricultural mapping of oil palm density [126], and flood detection for disaster management [127]. Today, many EO tasks leverage machine learning models and rely on the analysis of specific aspects in the data. Compression techniques should therefore maintain relevant features for downstream task, rather than focusing

only on perceptual reconstruction [34].

Comparison with Natural Images and Entropy Analysis. Unlike natural photos, which prioritize visual appeal and are often post-processed for aesthetic purposes [128], RS data undergoes radiometric, atmospheric, and geometric adjustments to ensure scientific accuracy. Moreover, EO data exclusively captures Earth landscapes, opposed to natural image datasets made up of diverse scenes and objects. To highlight how these differences translate into distinct compression demands, we conducted an entropy analysis of two representative datasets: ImageNet [111] (natural images) and BigEarthNet [112] RGB bands (Sentinel-2 Level 2A satellite images).

Per-image entropy quantifies the average amount of information contained and gives us an indication of the pixel variability within a single image. Our analysis, shown in Fig. 11, involved calculating a pixel value histogram for each image and band, and computing its entropy. We randomly sampled 10,000 images from each dataset for comparison. The results indicate that ImageNet images have an on average higher per-image entropy. These values can be due to the general greater variety in colors and patterns, as well as to post-processing that increases contrast. Sentinel-2 L2A images show lower entropy, despite covering diverse landscapes. Pixel values within individual bands tend to be more concentrated, leading to lower per-image entropy. The very low entropy of some images can be explained by certain scenery classes. For example, sea images often have similar pixel values across the whole

image. This is a property that may be exploited by domain-specific compression algorithms.

These findings underscore the need for compression techniques tailored to the statistical properties of EO data. Compression leverages bias in a dataset, allowing short bit rates to be used for redundant elements in the input data. The design of a compression algorithm is therefore always subject to a fundamental trade-off between broad applicability and data specificity. Conventional compression methods optimized for natural images might not fully exploit the redundancies and correlations prevalent in RS data. Even within RS, the variety of spectral, spatial, temporal, and radiometric resolutions of the instruments complicate the development of a single effective algorithm. Research therefore tends to focus on a specific type of RS data, as we describe in Section III-B.

B. Classification of Compression Methodologies

This section reviews compression approaches for RS data, categorizing them into traditional hand-crafted methods and NC techniques.

1) Traditional Approaches:

Transform Coding methods, such as JPEG and JPEG2000, first convert data into the frequency domain before applying entropy coding. Common transformations include the discrete cosine transform (DCT) [129] and the discrete wavelet transform (DWT) [54]. Their efficiency make transform-based methods widely used in RS applications [130–132]. For example, Hou *et al.* [133] adapt JPEG to detect and simplify cloud features, while Gonzalez-Conejero *et al.* [134] modify JPEG2000 to handle EO image areas without useful information. The CCSDS has developed international compression standards based on JPEG which are commonly used for onboard applications [135–137]. Transform-based methods decorrelate input along the spatial dimension but require extensions to leverage spectral redundancy in multi- and hyperspectral RS data. Markman and Malah [138] extend the DCT to the spectral dimension, while Lim *et al.* [139] and Luigi Dragotti *et al.* [140] apply three-dimensional wavelet transforms using the SPIHT algorithm. Du and Fowler [141] and Du and Fowler

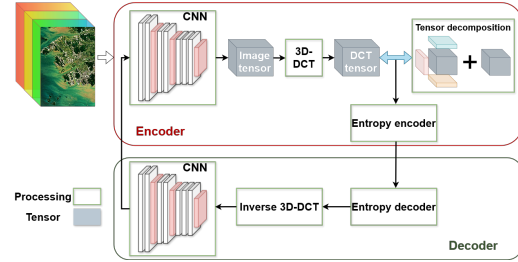


Fig. 12: CNN-based transformation as part of a tensor decomposition framework introduced by Li and Liu [143].

[142] combine JPEG2000 with PCA for spectral decorrelation.

Tensor Decomposition techniques address the multispectral nature of RS data by decomposing multidimensional matrices into low-rank components. Notable methods include the Tucker decomposition [144], which approximates a tensor with factor matrices and weight coefficients as a reduced core tensor. These methods are particularly effective for high-dimensional data like hyperspectral images, achieving high compression rates while preserving multidimensional structures [145–147]. Ongoing research aims to lower the computational complexity of tensor decomposition approaches for practical applications in RS [148–155]. In the context of SAR, raw radar echos are typically compressed onboard using Block Adaptive Quantization (BAQ) [156]. BAQ adapts quantization levels per block to meet bitrate requirements. Extensions of BAQ [157, 158] dynamically adjust bitrates to improve compression rates.

2) *Neural Approaches:* We now introduce NC techniques for RS, summarized in Table III.

Neural transformation. As an early contribution, Li and Liu [143] combine a CNN-based transformation with tensor decomposition (Figure 12). The encoder CNN, optimized for MSE, is combined with a DCT to produce a compact representation, reducing the computational cost of tensor decomposition.

Autoencoders compress data by encoding it into a lower-dimensional latent space and minimizing a reconstruction error, making them a nonlinear trans-

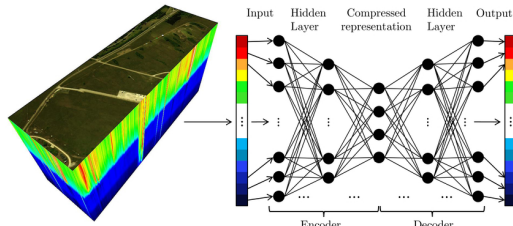


Fig. 13: Autoencoder for hyperspectral data by Kuester *et al.* [110]. Figure taken from the original paper.

form coding method. Yang *et al.* [135] introduce an early autoencoder for satellite imagery, replacing sigmoid activations with a ridgelet transform [159] to improve compression performance. Kuester *et al.* [110] introduce an autoencoder for hyperspectral data by compressing only the spectral component (Figure 13).

Rate-Distortion Autoencoders. Rather than relying solely on dimensionality reduction, rate-distortion autoencoders are optimized end-to-end for bitrate and reconstruction quality. Following Ballé *et al.* [43], several studies have applied CNN-based rate-distortion models to optical satellite imagery [160–163], aerial imagery [90] and SAR data [164–166].

Reduced-complexity Rate-Distortion Autoencoders. CNNs capture spatial features well, and larger kernel sizes enables them to capture information over broader image ranges. However, increasing kernel size also increases model complexity, limiting their suitability for onboard compression applications. Alves de Oliveira *et al.* [160] propose a reduced-complexity VAE that outperforms JPEG2000 [7]. They reduce network parameters and simplify the entropy model by using a parametric estimation of a Laplacian distribution. While this demonstrates that lower-complexity models can compete with the previously mentioned neural methods, their computational costs remain higher than those of traditional onboard techniques. Mijares i Verdú *et al.* [116] adapt a hyperprior VAE for hyperspectral data by clustering spectral bands and applying separate, smaller compression models

to each cluster.

Instrument diversity. Compression approaches within the RS domain must address diverse data types, including modalities, like SAR, which are significantly different from ordinary images in computer vision. For aerial imagery, Zhu *et al.* [90] incorporate radiation calibration into a rate-distortion compression model and exploit interspectral redundancies using 1×1 convolutions. For SAR, compression methods have been developed for different stages of data processing (Figure 14). For onboard compression, [167] extend the BAQ algorithm by a CNN-based optimization of the bitrate used for quantization. Martone *et al.* [168] use backscatter statistics to improve resource allocation and image quality. Other research for SAR focuses on accurate phase and amplitude reconstruction. Maharjan and Li [169] introduce a complex-valued hybrid approach that combines HEVC for encoding with a neural decoder to precisely reconstruct phase and amplitude characteristics. Pilikos *et al.* [165] propose two architectures for complex-valued SAR compression based on Vector Quantized VAEs (VQ-VAE), which use real-valued convolutional layers, while activation functions, batch normalization, and backpropagation are complex-valued. Conversely, [166] use complex-valued convolutional layers to decode the HV polarization.

Architectural adaptations. Many studies build on the hyperprior model [57] but modify the transform architecture. Xiang and Liang [170] integrate attention and long-range convolution to capture spatial redundancy. Kong *et al.* [171] modify the encoder to extract spatial-spectral features at multiple scales and adaptively adjust the weights of the features from different branches of the encoder network. Fu and Du [172] introduce a mixed hyperprior net with two prior models: a transformation-based prior to capture global redundancy and a CNN-based prior to capture local redundancy. Gao *et al.* [173] extend the hyperprior model with an enhanced residual attention module (ERAM) that applies spatial attention to create importance masks for adjusting bit distribution across latent channels. For SAR, Fu *et al.* [174] use multi-layered residual blocks and hyperpriors with local and global context

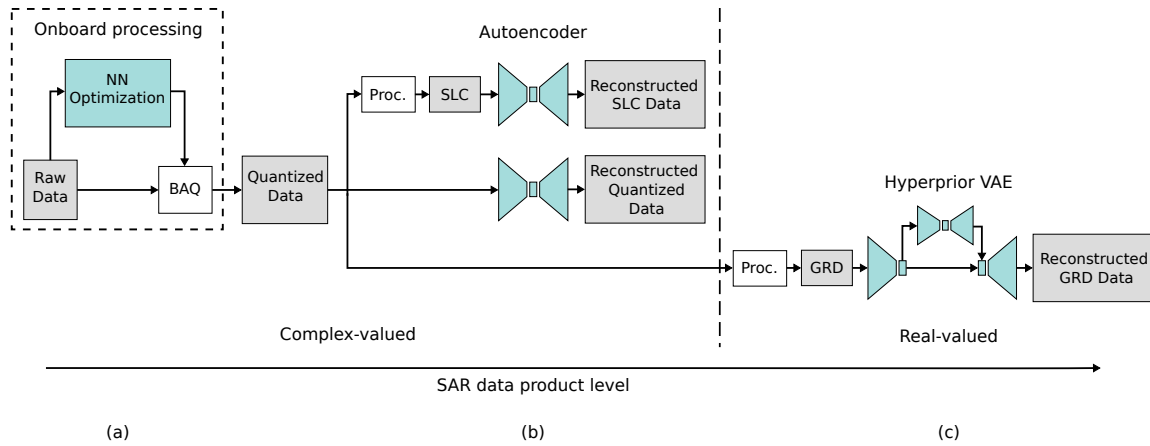


Fig. 14: Overview of the SAR data processing stages where compression methods have been developed. In (a) the raw SAR data is compressed before wireless transmission of the data. Most end-users have access to (b) the compressed data or complex-valued data products calculated from the compressed data, e.g. Single Look Complex (SLC) products, or (c) real-valued data products projected onto an Earth ellipsoid model, e.g. Ground Range Detected (GRD) products.

information, while Di *et al.* [175] utilize pyramidal feature extraction. Their approach involves a single Gaussian hyperprior framework and the pyramidal encoder to capture both coarse and fine structures.

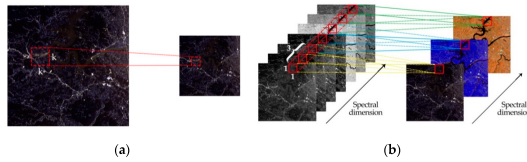


Fig. 15: Spectral convolution taken from Kong *et al.* [161]. (a) 2D convolution; (b) 1D spectral convolution. Figure taken from the original paper.

A common approach in RS compression is to decompose input images into spatial and spectral components. Kong *et al.* [161] propose a feature extraction module that extracts spectral and spatial features separately, using spectral convolution (Figure 15), and fuses them later for further processing. Similarly, Cao *et al.* [162] extract spectral and spatial features separately but without fusing them at a later point. They incorporate Tucker decomposition through tensor layers [163] for better decomposition of the multi-way data representations. Although

not always aimed at reducing complexity, spatial-spectral decomposition can enhance computational efficiency, particularly for datasets with many spectral bands. Besides adaptations to the transform model, novel entropy models in the RS domain, such as Gaussian mixture models (GMM), refine the estimation of latent distributions [119, 176].

Hybrid methods. Several approaches combine non-learnable wavelet transforms with end-to-end compression methods. Anuradha *et al.* [177] combine DWT for spatial-spectral decorrelation with LSTM networks for hyperspectral data. Xiang *et al.* [119] apply a DWT to latent representations, separating high- and low-frequency features. Gaussian mixture models are then used to estimate entropy models for the high- and low-frequency components separately (Figure 16). Xiang and Liang [178] also addresses the challenge of reconstructing high-frequency information in RS images, which often leads to edge-blurred artifacts. They introduce a two-branch architecture that employs a DWT to separate the input data into high-frequency and low-frequency components, which are then processed separately in dedicated sub-networks.

Explicit bitrate allocation. Some approaches

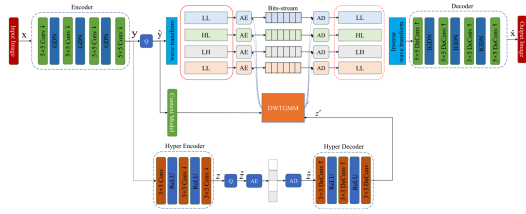


Fig. 16: NC architecture with an incorporated non-learnable wavelet transform introduced by Xiang *et al.* [119] as a Discrete Wavelet Transform-Based Gaussian Mixture Model. Figure taken from the original paper.

add a mechanism to explicitly control the code length allocated to different regions of the input. This is achieved through the introduction of importance maps and attention modules, weighing the compression of certain areas of an input image. Ye *et al.* [179] employ an image segmentation approach to create semantic maps before compression, thereby ensuring enhanced detail fidelity. The compression architecture incorporates an attention mechanism and a rate allocation technique that assigns higher compression rates to regions with smaller-sized details. Deng and Huang [180] incorporate a quality map from a pretrained ViT network to preserve high information content in the regions of interest while reducing redundancy in non-target areas.

Generative Adversarial Networks. GAN-based compression models have shown impressive performance at lower bitrates. Leveraging the VQ-GAN [86] architecture (Figure 17), such models usually consist of autoencoders with GANs serving as decoder modules. The associated adversarial loss is tailored to favor either specialized [181] or generalist [182] compressed representations. Zhao *et al.* [182] optimizes for the visual realism of reconstructed images by including a perceptual similarity term within the adversarial loss of a Conditional GAN [183] decoder. To improve the quality of decompressed edges, textures, or contours Zhao *et al.* [184] propose a Laplacian of Gaussian loss for a symmetric lattice GAN. On the other hand, Li *et al.* [181] focus on generating generalist compressed

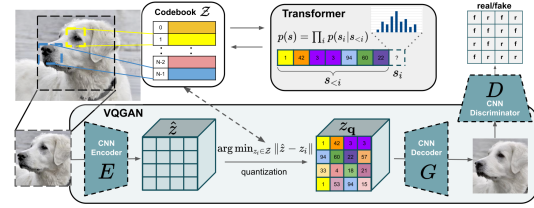


Fig. 17: *Vector-Quantized Generative Adversarial Network (VQ-GAN)* architecture introduced by Esser *et al.* [86] as an extension of the *Vector-Quantized Variational Autoencoder (VQ-VAE)* replacing the pixel-based image reconstruction loss by a discriminator network D . Figure taken from the original paper.

representations. Using Least Squares GANs [185], they reconstruct (dense) low-frequency components from (sparse) high-frequency components of the original images.

Implicit Neural Representations do not rely on autoencoder backbones and have demonstrated their potential in RS [186], outperforming JPEG2000 on both multispectral [187] and hyperspectral [155] datasets. INR-based methods regress the channel values for each pixel of a given image based on corresponding pixel coordinates, or transformations thereof. By optimizing the fidelity of these regressed values, a neural network encodes the implicit mapping between spectral values and pixel locations. The trained weights then undergo quantization and entropy coding, thus serving as a compressed representation of the images. Li *et al.* [187] successfully apply INRs to multi-spectral image compression. They train an MLP with equally sized residual layers to predict pixel values from longitude and latitude coordinates associated with pixel locations. Given the size of the input images and residual blocks, an upper bound to the width of the MLP hidden layers is derived that allows for effective compression. This method matches the quality of reconstruction of JPEG2000 while using half the bits per pixel. A similar approach for hyperspectral imagery is implemented in the FHNerf [155] model, which regresses pixel values from transformed pixel coordinates using Neural

Radiance Fields [65]. The proposed model nearly doubles the reconstruction quality of traditional and autoencoder-based NC at comparable compression ratios.

The main advantage of INR-based compressors is that they are agnostic to certain image features, such as native resolution. In principle, the produced representations are invariant of the scale of the original image, and their size uniquely depends on the architecture of the model performing the regression task. In other words, rather than data compression, methods relying on INRs perform model compression. While these methods stand out as more generalist alternatives to autoencoder-based NC, the latter still achieve higher compression efficiencies for multi-spectral images and are not bounded by the size of the compressor's backbone model.

Feature Compression focuses on compressing representations for downstream tasks rather than reconstructing the input and has seen some preliminary exploration in the RS domain. Furutanpey *et al.* [34] leverage this approach to mitigate the bandwidth bottleneck between satellites and base stations. They design an end-to-end pipeline for onboard feature compression capable of producing task-agnostic features and perceptually similar reconstructions of the input data. Their evaluation on benchmarks for object detection from aerial images demonstrates improved performance compared to neural image codecs and existing neural feature compressors [100, 101]. Gomes and Brunswiler [35] use the same idea tailored to the transmission of features from data centers to end users hosting models for training or inference. They adopt a rate-distortion objective that combines masked auto-encoding as a form of SSL [188] with an entropy penalty to encourage compressible, general-purpose features. They further leverage an existing FM and show that fine-tuning a small portion of the pretrained weights is sufficient to create a general feature compressor for classification and segmentation.

Dictionary Learning involves learning a set of basis elements from the data and representing the data as sparse combinations of these elements, en-

abling efficient compression. In RS, Wu *et al.* [189] propose a double sparsity model for hyperspectral images. Their method, involving entropy coding with Differential Pulse Code Modulation (DPCM) and arithmetic coding, demonstrates superior rate-distortion performance and improved spectral information preservation compared to 3D-SPIHT and JPEG2000. Wang and Celik [190] propose a dictionary learning approach that induces sparse coefficients through online learning. The sparse coefficients are quantized and entropy-coded to generate the final bit stream. Ertem *et al.* [191] also employ dictionary learning to improve hyperspectral image compression. Their method generates superpixel maps for adaptive spatial-spectral representation, computes an optimal dictionary, and determines sparse coefficients using Simultaneous Orthogonal Matching Pursuit (SOMP). Notable innovations include a modified dictionary learning step, an ordering scheme that eliminates the need to transmit the superpixel map as side information, as well as using DPCM to reduce sparse coefficient magnitudes.

C. Summary

The variety in EO instruments produces diverse datasets, resulting in a wide application space for compression techniques. However, this data diversity also leads to a fragmented field. Studies often train and evaluate models on different datasets, which complicates method comparisons. While hand-crafted methods remain prevalent, recent research has shifted towards neural methods that directly optimize a rate-distortion objective. In particular, the hyperprior framework introduced by Ballé *et al.* [57] has been widely adopted and adapted for RS data. The flexibility of this approach in designing synthesis and analysis networks allows research to explore diverse architectures, with most studies emphasizing innovations along the “Transform” axis identified in Section II-B.

Future research directions include exploring data-specific characteristics, such as exploiting temporal correlations inherent in the relatively static nature of consecutive EOs. Recent studies adapt traditional video compression techniques to satellite image sequences. With reference-based coding, historical

TABLE III: Contributions to the field of neural compression for remote sensing described in Section III-B ordered along the axes of the taxonomy described in Section II-B.

Axis	Approach	Papers
Transforms	Complexity Reduction	[116, 160]
	Novel spatial extraction	[90, 170, 171, 173, 175]
	Novel spectral extraction	[110, 171]
	Separate spectral/spatial extraction	[161, 162]
	Incorporate Wavelet Transform	[119, 177, 178]
	Bitrate Allocation	[179, 180]
	Image-specific (INRs)	[155, 187]
Entropy Models	Hyperprior with attention	[173]
	Multiple hyperpriors	[172]
	Split latent space	[119, 162, 178]
Optimization Objectives	Adversarial loss (GANs)	[181, 182, 184]
	Downstream Embedding	[34, 35]

images of the same region are used to compress only the temporal changes instead of individual images [124, 192]. The exploitation of spectral redundancy remains a research focus. For instance, INRs pose an interesting research direction for hyperspectral datasets, which often suffer from limited training samples.

As machine learning becomes increasingly important for processing RS data, it is essential to align compression techniques with the requirements of downstream tasks. Consequently, research is starting to shift from compression strategies optimized for human perception—such as minimizing MSE—toward methods that preserve data integrity for machine processing [34]. Neural feature compression shows promise in two distinct scenarios: the transmission of features from satellites to ground stations, overcoming the data downlink bottleneck, and the transmission of features from data centers to analysts for model training and inference.

Due to the limited resources onboard, only compression methods with low computational and storage complexity can be used. Nevertheless, the onboard application of machine learning represents an important future aspect, for example, to carry out geophysical tasks [193]. With this in mind, the European Space Agency (ESA) launched the Φ -Sat-2 mission to demonstrate onboard AI for various use cases. For instance, the first neural-based onboard

compression using convolutional autoencoders was demonstrated, opening up new opportunities to save bandwidth and storage [194] as well as to support the growing amount of satellite data. Lu *et al.* [195] developed an onboard AI model able to detect bushfire smoke much faster than traditional ground-based processing, which highlights the potential of onboard AI for real-time geophysical applications.

IV. NEURAL COMPRESSION FOR CLIMATE DATA

Earth system models (ESMs) are one of the key tools for understanding the impact of anthropogenic climate change on the Earth. ESMs model the dynamics of the earth's atmosphere on a discretized spherical grid; the horizontal grid spacing in current climate models is usually on the scale of around 100 km. However, with such a coarse resolution many important processes, such as precipitation and deep convection, cannot be explicitly resolved, which motivated the development of the next generation of climate models with grid spacings on the scale of 1–5 km [196–200]. The increased resolution of these models, in turn, leads to a significant increase in the size of datasets produced by ESMs [201, 202]. For example, the recently launched Destination Earth initiative [200] generates around 1 petabyte of data *per day*; making it infeasible to store all the generated data on disk for long time-scales.⁶

⁶<https://stories.ecmwf.int/the-digital-twin-engine/>

With storage costs now making up a significant factor of computing center budgets [203, 204], there is a pressing need for compression algorithms for climate data.

A. Challenges

1) *Data Characteristics*: Data generated by climate simulators has multiple key characteristics that set it apart from other data modalities and emphasize the need for bespoke compression tools and algorithms:

Multidimensional data. The output of models includes multiple variables, e.g. wind, atmospheric pressure, temperature, etc., which are localized in space and time and stored in multidimensional arrays [205, 206]. While natural images and videos commonly only have a small number of channels, i.e. 3 for an RGB image, climate models can have hundreds of different output variables. A key feature of atmospheric data is that there is generally a high correlation in space and time, and between some of the different variables. Additionally, different climate models do not necessarily use the same grid projection. So while the output data is generally saved as a multi-dimensional array, the corresponding coordinate reference system might vary between different climate models [207–209].

Importance of high-frequency signals. The dynamics represented by a climate model are inherently chaotic; accurately modeling the evolution of the dynamics at smaller scales is important for accurately modeling the larger scale dynamics. The representation of variables at small scales also encodes the physics of key climate processes, such as clouds [199, 210]. This makes compressing climate data fundamentally different from compression methods for other datasets such as natural images. For natural images, blurring at smaller scales might be desirable because it does not create images that are visually distinguishable for humans [211]. However, the importance of small-scale features for climate processes means that compression algorithms that overly smooth data over small scales may result in undesirable or unpredictable downstream effects [212].

Extreme events. A key goal for climate modeling is to predict the probability of extreme events such as floods or storms, for which small-scale variability of precipitation is important [213]. It is therefore imperative that the statistics of these extreme events are preserved when compressing the data, which may require additional explicit constraints, due to their unlikely nature in the scope of entire datasets [214].

Lack of quantitative metrics. As outlined in Section II, compression methods are usually evaluated based on how well they reconstruct the input data. This requires a distortion metric to compare the original data, x , and the reconstructed data, x' . However, for climate models classical metrics such as the pixel-wise mean-squared error are often insufficient to capture the structural differences between inputs and reconstructions that scientists are interested in [215–217]. Ideally, any reconstruction should conserve the physical properties of the input in space and time, e.g. individual clouds should have the same mass in the input and reconstructed data and spatio-temporal structures should be preserved. However, developing metrics that capture the distortions relevant for climate data is still an active area of research [204, 218–220].

2) *Data Acquisition and Application*: Modern climate models are executed on the world's largest supercomputers. A single forecast run often requires carefully orchestrating and integrating multiple sub-components, such as ocean and atmospheric models [208, 209, 221]. Even on a supercomputer, completing a single run can take weeks to months. Consequently, data generated from these models is typically produced once and then stored for subsequent access by scientists. However, researchers often need only a subset of the data for their analysis—restricted to a specific time period, geographical region, or selection of variables—so they often want to avoid downloading the entire dataset. Given that data is usually generated only once, it is then logical to invest significant computational resources towards compression if this reduces the bandwidth required for transmitting the data to scientists for further analysis.

B. Classification of Compression Methodologies

1) *Traditional Approaches*: Most existing codecs for multi-dimensional arrays are hand-engineered (in the sense of Section I-B) and employ the transform-based compression approach described in Section II. A key to achieving good compression ratios is to exploit correlations in the data; many codecs divide the input data into sub-blocks⁷ and try to identify correlations within a sub-block. ZFP [222] decorrelates individual sub-blocks using an orthogonal transform. SZ3 [223] uses a spline-based interpolator to identify correlations in a given sub-block. TTHRESH [224] uses a generalization of the singular value decomposition (SVD) to tensors with more than three dimensions to transform the data.

Arguably, an even more simple approach to compression is to reduce the number of bits used for the floating point representation of individual output variables. Most output of climate models is stored in 64-bit precision but to or many variables, the least significant bits in the mantissa of the IEEE floating point representation are effectively random noise, i.e. they are not useful in predicting the future state of the system [225, 226]. One can further formalize this by deriving a measure of the “real information” contained at a given bit location for a given variable [219]. Klöwer *et al.* [219] find that in data from the Copernicus Atmospheric Monitoring Service [227] most variables contain fewer than 7 bits of real information, i.e. the remaining 57 bits in the floating point representation are purely random noise which can be safely discarded. This has been the motivation for a couple of compression schemes [219, 228] which discard a certain number of least significant bits in the mantissa of the floating point representation. The advantage of this approach is that it is complementary to the compression schemes presented in the previous paragraph because it can simply be run as a pre-processing step before passing the dataset to a compressor.

2) *Neural Approaches*: Compared to other data modalities such as images, text, or video there

has been relatively little work on developing NC methodologies for climate data. Most work focuses on adapting existing architectures to the inherently spherical geometry of the data domain.

Implicit neural representations (INRs) have been shown to be able to compress ERA5 temperature data at higher compression ratios than JPEG2000 [231]. Huang and Hoefler [229] demonstrate that representing the data with random Fourier features [232, 233] leads to improved compression ratios. However, these INR codecs can struggle to accurately capture extreme events in the input data. In a case study examining the reconstructed ERA5 geopotential variable during Hurricane Matthew in October 2016, Huang and Hoefler [229] found that the INR reconstructions failed to preserve the extreme values at the center of the hurricane. Further work is needed to address these shortcomings in order to make INR a viable compression method for climate data.

Autoencoders Mirowski *et al.* [214] evaluate a variety of neural architectures for compressing the temperature, pressure, wind velocity, geopotential, and humidity variables in the ERA5 data. The architectures they consider are VQ-VAE, VQ-GAN, factorized prior and hyperprior models (see Section II-C for a description of these models). As a pre-processing step, they use the HEALPix projection [234] to re-grid the data which leads to pixels representing equal area and allows to efficiently compute spherical harmonics transforms. Their experimental results indicate that the hyperprior model gives the best compression results and is able to preserve the power spectrum of the data more accurately than the alternative architectures; even though, the hyperprior model is trained only using a mean squared error reconstruction loss. While the hyperprior model is found to be better at preserving extreme events compared to the INR approach of [229], it does not provide any theoretical guarantees on bounding the reconstruction error compared to traditional approaches such as SZ3. Concurrently, Han *et al.* [230] also use a hyperprior architecture to compress ERA5 data and develop a novel window-based attention mechanism for atmospheric data; they report compression ratios

⁷For a d -dimensional array, each sub-block has size n^d where n is the sub-block size.

TABLE IV: Contributions to the field of neural compression for climate data described in Section IV-B ordered along the axes of the taxonomy described in Section II-B.

Axis	Approach	Papers
Transforms	Pre-process input with HealPIX projection	[214]
	Pre-process input using Random Fourier Features	[229]
Entropy Models	Hyperprior	[214]
	Hyperprior with Attention	[230]
Optimization Objectives	Rate-Distortion and/or Adversarial loss (GANs)	[214, 230]

of around $\sim 300\times$ while Mirowski *et al.* [214] report ratios of $\sim 1000\text{--}3000\times$.

C. Summary

NC methods have shown encouraging results in being able to compress weather and climate data. As highlighted in Table IV, most existing work focuses on adapting architectures developed for the image compression domain and adapt them to the climate domain by adding pre-processing steps to account for the underlying spherical geometry of the input data. While existing work reports impressive compression ratios on the order of $1000\times$, more work is needed to establish trust in these novel neural codecs to ensure they do not erase extreme events such as hurricanes from the input data.

Overall, the use of lossy compression for climate data is relatively under-explored both with traditional and neural approaches. This can be attributed partly due to a lack of trust in lossy compression algorithms in the climate community which tends to have high standards for data integrity [204, 235]. There is a growing body of work that addresses this issue by developing quality metrics that lossy compression algorithms need to pass in order to be suitable for climate data [218, 235–238]. Fundamentally, the goal of these evaluation metrics is to ensure that *lossy compression does not change scientific conclusions* drawn from the data [204]. How to exactly assess this goal is still an area of open research. Suggested quality checks include: asking climate scientists to identify the (lossily) reconstructed members of an ensemble of ESM runs [235]; adapting the SSIM metric for scientific datasets [220]; using the change in real information

content [219] of variables to detect reconstruction artifacts [238]; and defining thresholds on quantitative metrics [237].

However, so far, these quality metrics designed by climate scientists have seen little adoption in the NC literature. Generally, neural approaches tend to mainly evaluate the trained codec based on mean-squared error and by assessing the power spectrum of the original and reconstructed data [214, 229]. To ensure the adoption of neural codecs, it is therefore important that algorithm designers work hand-in-hand with climate scientists to build trust.

V. NEURAL COMPRESSION: IMPLEMENTATION & APPLICATION

A. Neural Compression for Geospatial Analytics Platforms

The exponential growth of data from EO missions has led to significant challenges in transfer, storage, and processing, resulting in substantial resource requirements [109]. Given the broad relevance of EO data across multiple fields [239], innovative real-world compression methods can have a considerable impact. As discussed in Section III, NC can significantly reduce the volume of EO data that needs to be transferred. Furthermore, the sheer scale of data collection means that most of this data will never be analyzed by humans. Instead, automated processing using computer vision algorithms—primarily neural networks—must handle the bulk of the data. Given this context, despite most NC research focusing on human perception, as seen in Sections II-C and III-B, we emphasize the emerging field of feature compression as a promising solution to these challenges.

A setting in which an image's feature vector is compressed instead of the image itself may allow for much greater compression ratios: The neural compressor may deem perceptual information unnecessary when maximizing performance on downstream tasks (e.g., scene classification, semantic segmentation, object detection) instead of reconstruction. Using feature compression, Dubois *et al.* [99] demonstrate a 1000x rate gain compared to JPEG for ImageNet classification adapting a pre-trained FM for compression. While segmentation or detection tasks remain relatively unexplored, even a fraction of this rate gain would be significant. FMs [240], pretrained by SSL, may be key in this context, as they provide generic features, which can be utilized for multiple downstream tasks. A platform like ESA's Copernicus Data Spaces Ecosystem (CDSE), which hosts the data from Copernicus Sentinel missions could compute and host compressed features generated by a FM trained with a compression objective. Customers would request those compressed embeddings, rather than raw data, for their various downstream applications and benefit from reduced i) egress cost, ii) data transfer latency, iii) storage requirements, and lower compute requirements for iv) model training, v) model inference, as well as from fast vi) search in data sets.

Training: In a feature compression system, training datasets for downstream tasks could be transferred as compressed embeddings. This approach reduces the volume of data that must be transferred and stored and allows the client to receive feature vectors directly representing the requested data. As a result, the client is relieved from the need to train a large backbone network as a feature extractor.

Inference: The client's trained downstream model can then use these compressed embeddings for inference. This is particularly advantageous in scenarios requiring real-time user interactions, where latency is primarily dictated by the time needed to transfer input data. It also benefits large-scale inference tasks, such as global-scale analyses, where the reduced costs of data transfer and storage enable more frequent evaluations (e.g., monthly instead of yearly). In both cases, since embeddings are

computed server-side, only a lightweight network is necessary on the client side. This empowers users without access to powerful computational resources to train models and enables edge devices to perform inference.

Data Federation / Data Fusion: Larger downstream models may also benefit from feature compression. The significantly reduced amount of data that must be transferred when working with NC allows for building downstream machine learning models based on fused data, hosted in different data storage facilities, e.g.: fusing optical imagery from Sentinel-2 hosted on the Amazon Web Services (AWS) in Europe (AWS-eu-central-1) with Landsat8 multi-spectral satellite imagery hosted in the United States (AWS-us-west-2), with large spatial and temporal extent Fig. 18).

Search: Similarity search driven by embeddings has demonstrated significant potential in the EO domain, both in academic research [241] and in practical applications. This method enables efficient querying of large datasets by identifying entries that are similar to an input query, bypassing the need for metadata-based searches. Typically, this is achieved using a vector database—a specialized system designed to retrieve similar vectors from vast datasets based on a similarity metric in vector space. Implementing a compressed, server-side store of these embeddings could facilitate the global scalability of this approach over extended periods.

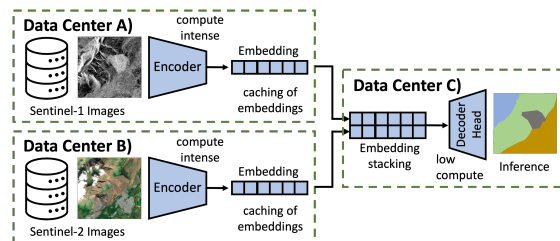


Fig. 18: Concept of data federation through compressed embedding sharing between data centers.

Feature compression offers an effective balance between storage and processing demands: the computationally intensive task of generating embeddings can be handled by high-performance com-

puting (HPC) systems, while lightweight decoding is reserved for end applications. This allows intermediate processing steps to operate on the reduced data size provided by the embeddings. Moreover, using embeddings derived from FMs has shown great promise in few-shot learning [242], which can significantly lower the resources needed for data labeling while maintaining high task accuracy.

While deep NC of EO data addresses the challenges of storage, transfer, and processing, it introduces new challenges that must be managed when working with neurally compressed data:

Tiling: The optimal strategy for tiling the underlying data is unclear. Standard tile sizes in EO (e.g. 110x110 km in Sentinel-2) are generally too large to be used as inputs to neural network models. Additionally, embeddings are taken to represent the entire tile atomically and cannot (in general) be subdivided, making large tile sizes insufficiently granular for many EO tasks. Conversely, smaller tile sizes increase the number of embeddings that must be computed and stored.

Caching / Pre-computing: An additional advantage may arise from intelligently caching compressed embeddings. As the computation of these embeddings is concentrated on the server, previously duplicated computing efforts across distributed machines can be reduced. However, pre-computing embeddings via batch processing can also present difficulties due to tiling and the indivisibility of embeddings, potentially limiting users to requesting data regions that align with the tiling grid. An alternative would be stream processing, which offers greater flexibility but at the expense of efficiency.

Spatio-Temporal extent: The extent of data captured by a single compressed embedding must also be considered in spectral and temporal dimensions. Greater compression may be achieved by jointly compressing more spectral bands and multiple temporally adjacent observations, but this reduces flexibility, as users may only require a subset of the data.

Standards: Additional challenges involve establishing standards and adapting geospatial data platforms. To fully realize the benefits of neurally

compressed EO embeddings, EO cloud platforms need to support their processing. Currently, neither proprietary platforms like Google Earth Engine [243] nor the OpenEO standard [244] support neurally compressed embeddings. Furthermore, an open standard is needed for storing and transferring these compressed embeddings between storage and processing environments. To ensure interoperability, such a format must contain all necessary information for decompression. However, to our knowledge, no such standard currently exists. The Cloud-Native Geospatial Foundation [245] is surveying how the geospatial community is storing embeddings in GeoParquet to potentially develop guidelines or introduce a standard for storing and exchanging embeddings in the future. While this may lead to a solution for storing raw embeddings $\mathbf{z}$, it does not address the challenge of creating a standardized format for storing neurally compressed embeddings. In addition to the compressed data, metadata must be included to provide details such as the FM used to generate the embeddings $\mathbf{z}$ and the geographic area to which the embeddings apply. To ensure that neurally compressed EO data is findable according to the FAIR data principles [246], it should be cataloged in a standardized manner, such as by registering it in a Spatio-Temporal Asset Catalog (STAC) with sufficient metadata.

We therefore argue that existing data catalogs and EO processing platforms must be extended to support the provision and processing of neurally compressed embeddings. Only then can we reap the benefits of using embedding representations over base imagery.

B. Cost- and Energy-Efficiency & Latency

Space Segment Efficiency. Energy efficiency is crucial in RS. An example is the need to maximize the operational lifespan of nanosatellites, ensure effective data transfer within limited downlink windows, and reduce operational costs [123]. Efficient onboard data processing reduces the volume of data transfers, mitigating bandwidth constraints. Advances in *Orbital Edge Computing* and neural feature compression may enable satellites to handle large data volumes without excessive energy

use, enhancing overall system efficiency. This is essential for the sustainability and effectiveness of satellite constellations in capturing and transmitting valuable EO data.

Ground Segment Efficiency. However, the approaches reviewed also have a significant impact for the ground operation of the data infrastructure. In this section, we aim to give back-of-the-envelope estimations that quantitatively assess the role that NC can have in geospatial analytics. A concise system analysis of the information and communication technology infrastructure as well as a detailed scenario analysis are left out of the scope of this review. However, we aim at providing a first orientation with a simple model calculation. The demonstrative case we study is the Copernicus program, for which the Sentinel satellites are the main data sources. We investigate the potential benefits of Copernicus data products leveraging neural (feature) compression on the server side, before transmission to clients. The focus of our analysis is energy efficiency, with aspects of cost-efficiency and latency also included.

Copernicus Data Volumes and Transfer Cost. According to the Copernicus Data Dashboard⁸, at the time of writing, the total volume of data products is growing by 759 TB per month, with 6.2 PB of data products downloaded in the same period. We use these figures and estimate approximate yearly values of 10 and 100 PB respectively, rounded up to account for expected growth. Price indications for data transport out of cloud storage (egress costs) are given on the website of AWS⁹. While the detailed pricing depends on the site of the host and consumer, 20 USD/TB is a realistic lower bound for 2024 when expecting a high quality of service. Hence, a gross data transfer cost of 2,000,000 USD/year per data product download is a reasonable estimate.

Energy Spent on Transfer. The energy footprint of data transfer is very difficult to assess. For example, a meta-study by Aslan *et al.* [247] summarizes 14 studies that deviate by a factor of ten or more, even after adjusting the system boundaries. The web

page *wholegraindigital* expands on the challenge of defining suitable system boundaries¹⁰. The authors challenge the idea of a single metric measuring the energy of data transfer and point out that Aslan *et al.* [247] only evaluate the usage of a subsystem. With this in mind, as a starting point, we pick the number of 0.01 kWh/GB, carefully following the extrapolation in Fig. 3 of Aslan *et al.* [247] but correcting upwards. With this assumption, the annual energy cost of data transfer sums up to about 1 GWh/year.

Compute Demand for Neural Compression. NC could reduce the data transfer burden. However, the compression is associated with energy consumption as well. We propose a simple order-of-magnitude estimate to assess the consumption that transfers the insights from computer vision to RS using BigEarthNet as an intermediate, where multispectral data is available in a similar format to natural images [112]. Typical convolutional encoder networks from computer vision may require up to several tens of billions of floating point operations, or GigaFLOPs, for processing an RGB image of the characteristic size 224x224 that has been abundantly used in computer vision datasets [111, 248].

For transformer architectures, the operation count can be as high as hundreds of GigaFLOPs per image. For simplicity, we assume here that compressing a multispectral image with resolution 120x120 is comparable to processing a 224x224 RGB image. For encoding the entire BigEarthNet-S1 archive consisting of 590,326 non-overlapping image patches with a total volume of 66 GB, assuming 100 GigaFLOPs per image, this adds up to 6.6×10^{16} FLOPs in total, or about 10^{15} FLOPs/GB.

Time and Energy Demand for Neural Compression. The GPU that currently dominates AI compute centers, NVIDIA's A100 GPUs, can, according to Kesselheim *et al.* [249], with moderate optimization, sustain 50% of their nominal performance using fp16 accuracy throughout the ML workloads. Hence, they can achieve approximately 150 TeraFLOPs per second at a power consumption

⁸<https://dashboard.dataspace.copernicus.eu>, accessed on 2024/06/27

⁹https://aws.amazon.com/s3/pricing/?trk=ap_card

¹⁰<https://www.wholegraindigital.com/blog/website-energy-consumption/>

of 400 W¹¹. Based on our previous assumptions, and adding an overhead of 50% for server operations and cooling, we obtain a processing time of seven seconds and an energy consumption of about 1 Wh per GB. Scaling up to the yearly data generation of about 10 PB per year, this amounts a total energy requirement of approximately 0.011 GWh/year. The total processing time adds up to approximately two years, so with a single commercial eight-GPU server the continuous provision of compressed data products can be realized.

Copernicus Data		
Data volume to process	10	PB/year
Downloaded data volume	100	PB/year
Energy of data transfer	1	GWh/year
Egress cost	2Mio	USD/year
Cost of Compression		
Compute time on A100	1.89	years
Energy for compression	10	MWh/year
Compression to transfer energy ratio	0.01	

TABLE V: Key parameters and results of the analysis comparing energy needed for data compression relative to data transfer.

Energy Demand for Neural Compression compared to Data Transfer. Comparing the potential energy savings and the compression, despite all uncertainties, it seems apparent that a one-off compression of the data would be almost negligible with an estimated three orders of magnitude difference from the transfer consumption and thus, the utilization of compression is highly beneficial. The key parameters and results of this investigation are depicted in Table V.

Embedding vs. Image Reconstruction. As a final factor, it is important to assess the consumer side. Here we distinguish two scenarios. In scenario (a) the consumer performs an ML operation directly on the transferred embeddings; in (b) the consumer reconstructs the data for other downstream tasks. In scenario (a) typically the user will save energy as the compact compressed representation is potentially even better suited for this class of tasks. In scenario (b) we must consider the energy con-

sumption of the reconstruction. In many encoder-decoder architectures, the computational efforts of both are balanced. In this case, this would in turn produce a computation effort of approximately 10^{15} FLOPs/GB. It is important to consider with which computational devices (CPUs, older models of GPUs, etc.) and with which expertise the decompression is performed. Under non-ideal conditions, the reconstruction could offset the savings of the efficient transfer entirely. Encouragingly, projects such as `llama.cpp`¹² show that techniques such as model quantization allow for even computationally demanding ML models to be executed on a wide variety of hardware systems efficiently.

Neural Compression Benefit Scenarios. Finally, we would like to share two estimates demonstrating the possible latency improvements when employing neural feature compression. Considering the results reviewed in Sections II and III, we assume a neural feature compression system may be able to achieve a 10x improvement in compression ratio over JPEG2000. For our first example, we consider a researcher performing a spatiotemporal analysis over 10 years worth of multispectral imagery across all of Germany. For May 2024, all Sentinel-2 images available for the entire country require 470GB (L1C data product) when stored in JPEG2000 format. This extrapolates to an estimated volume of 56TB for one decade. With an internet download speed of 100MBit/s, the data is available with a delay of about 52 days, posing a significant obstacle to the progress of research. Neural feature compression may reduce the time to only about a week. For a second example, a researcher may want to create a mosaic of land cover information from Sentinel-2 imagery given a single timestamp. Assuming the global land mass covers an area of approximately 148 940 000 km², and given a single Sentinel-2 tile covers about 10 000 km², that amounts to roughly 15,000 tiles. With one tile consuming 0.8GB, that amounts to 12TB of data downloaded to generate a landcover product using JPEG200. With the same internet download speed from the previous example, we obtain a latency of 11 days, compared to about

¹¹<https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf>

¹²<https://github.com/ggerganov/llama.cpp>

1 day when neural feature compression is applied.

Summary. This analysis of the status quo of the data product generation and consumption of the Copernicus program shows tremendous potential for savings in cost and energy scale. Furthermore, we highlight how a reduction in latency can enable data-intensive applications for consumers. While the challenging number of variables involved in the analysis requires us to rely on coarse estimates and assumptions at times, we believe these figures are still illustrative of the potential impact of neural (feature) compression.

Future Directions. Future projections are extremely challenging. NVIDIA's Road Map presentation at Computex¹³ indicates optimism about future performance and energy efficiency gains that make compute-heavy approaches more attractive. Comparably, the cost of data transfer decrease as well. However the complex interplay of efficiency and demand with focus on data transfer energy demand is explained in Koomey and Masanet [250]. The increasing energy efficiency can lead to reduced energy consumption, but a more accessible resource can generate increasing demand that overcompensates the energy savings.

C. Democratization for Applications

Beyond Compression. As discussed in the previous section, the optimized compression of the massive raster data generated by EO systems and climate simulators show potential for reducing the energy cost and transmission latency both for data-distributing platforms and their end users. As a result, stakeholders with limited bandwidth may access scientific data previously out of reach for their resources. Moreover, when applied to embeddings generated by large pretrained models, NC would also permit downstream users with modest compute and deep learning expertise to benefit from expressive feature representations without the need for training a backbone from scratch on their end. Here, we propose to illustrate our point with four example applications that could directly benefit from such neurally compressed data.

¹³<https://blogs.nvidia.com/blog/computex-2024-jensen-huang/>

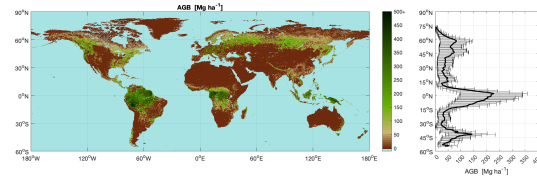


Fig. 19: Aboveground biomass map from the Climate Change Initiative (CCI) Biomass project [251]. Despite global coverage, this product has a 100 m ground sampling resolution and is only available for 2017, 2018, and 2020. This is limiting for applications needing to monitor the evolution of vegetation structure at higher spatio-temporal resolution. Using neurally compressed satellite imagery or features would allow the computation and distribution of more frequent, higher-resolution, global vegetation structure products.

1) Global Vegetation Structure Analysis:

Worldwide mapping of vegetation properties is of prime importance for understanding the global carbon cycle [252], the impact of human activities on carbon emissions [253], and the study of ecosystem services [254]. The accurate and frequent mapping of a small set of vegetation structure indicators such as canopy height (CH) and aboveground biomass (AGB) is key to the study of terrestrial ecosystem functions [255, 256].

The traditional protocol for estimating such indicators requires in-situ—sometimes destructive—manual measurement surveys. Due to the poor spatio-temporal scalability of this approach, much research effort has been invested in characterising vegetation structure from RS data with terrestrial laser scanning (TLS) and aerial laser scanning (ALS) [257]. While ALS provides accurate, dense, very high-resolution data, acquisition campaigns remain costly, limited to regional scales, with revisit rates of several years. The ultimate need to scale vegetation mapping to global scale with revisit rates below one year and low-cost data hence calls for space-borne data. Ideal satellite observations for global forest analysis need to capture vegetation properties at high spatial resolution with high revisit rate, and be freely available. Several works

have proposed to map forest structure from time-series of NASA/USGS Landsat or ESA Sentinel-1/2 acquisitions [258]. Recently, combining spaceborne LiDAR measurements from the NASA GEDI mission [259] with Sentinel imagery has shown great potential for regressing forest biophysical variables like AGB or CH at a global scale and 10 m resolution [260].

Still, Lang *et al.* [260] find that the prediction of a single, global map for the year of 2020 requires extensive computational power. In order to cover the entire landmass of the Earth (excluding Antarctica), a total of ~ 160 terabytes of Sentinel-2 image data need to be downloaded. Running the model on these images takes $\sim 27,000$ GPU-hours (~ 3 GPU years) of computation time, parallelized on a high-performance cluster to obtain the global map in *ten days* real time. Yet, the breakdown of the entire process reveals that more than half of the time is spent downloading and moving the data around.

Besides, the rise of SSL leading to the current emergence of RS FMs [16, 18, 19, 107] renders possible the distribution of expressive feature representations directly usable for downstream vegetation-related tasks [261] without the need for the compute or AI expertise required to train the corresponding deep learning architecture.

Consequently, a pipeline capable of efficiently and accurately transmitting neurally-compressed sensor data or pretrained feature representations would allow producing and frequently updating vegetation structure maps. By lowering the compute, bandwidth, and AI skills required for using deep learning models to regress vegetation structure variables from RS data, more stakeholders may take part in the production and analysis of such products. This would in turn benefit crucial applications such as ecosystem protection and global carbon cycle monitoring. What is more, new use cases may also emerge from the facilitated access to global vegetation structures. For instance, numerous industrial actors are in need for tools for monitoring deforestation-free supply chains without investing in large-scale data storage, computer infrastructure, nor deep learning knowledge. A typical example are companies depending on commodities sourced

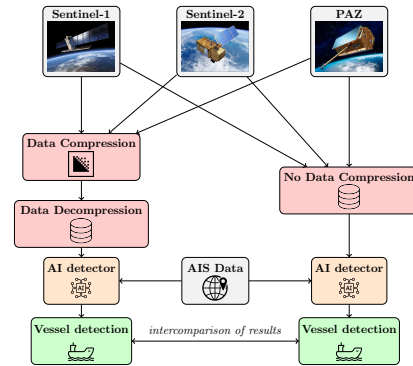


Fig. 20: Workflow for ship detection using satellite imagery and artificial intelligence.

in the tropics such as palm oil or cocoa [262, 263].

2) Ship Detection for Maritime Awareness:

Ship detection is an important aspect of maritime awareness, as ships often carry valuable cargo and pose a potential threat to populations and infrastructure. There are various methods for detecting ships, including SAR [264], optical modalities [265] and Automatic identification system (AIS) [266]. EO data allow ship traffic monitoring and the identification of potential security threats on large areas and the support of AIS data provides a technology used for maritime safety and security in near real-time to identify and track vessels. Receiving timely, reliable, and meaningful information is therefore crucial. In the last years, AI and ML have been used to detect, identify, and classify vessels in an automatic way [267]. Vessel identification could benefit from neurally-compressed RS images or corresponding pretrained features in order to:

- Compress the images to improve data transfer latency and facilitate access to relevant sources and collateral data (e.g. AIS).
- Support the creation tools for ship and port monitoring with minimal data labeling.
- Support the fusion of GeoData with AIS data for anomaly detection of ship movements.

3) **Climate and Air Pollution Prediction:** As described in Section IV, high-resolution climate models are able to resolve key small-scale phenomena such as clouds and ocean eddies [198,

199]. An additional advantage of the increased resolution of modern climate models is that their generated data is now at the same resolution as the observations from RS devices such as geostationary satellites [199]. However, the sheer volume of data generated poses challenges for full scientific exploitation, as the datasets are often unwieldy for efficient analysis and distribution.

The ability to compress the data into embeddings would significantly broaden the access to these datasets and enable new workflows. For example, the potential use cases for the generated embeddings include:

Training cloud classification and air pollution prediction models directly in the embedding space, circumventing the need for complex and computationally expensive image processing methods. This includes identifying and tracking convective storms in the embedding space.

Detecting extreme events by modeling the distribution of embeddings and detecting out-of-distribution samples directly in the embedded space.

Comparing the outputs of climate model simulations with observational data. Meaningful embeddings make it possible to compute statistics about the occurrence of individual cloud types (e.g. deep convection and shallow convection) which is more difficult in the raw data space [268].

While the use of embeddings has shown some promising early applications [268–271], their adoption is still in its infancy. It is important that any analysis run on the embeddings produced by a model lead to the same scientific conclusions as the analysis run on the full dataset. Hence, any model for generating embeddings should be developed in collaboration with domain experts to ensure that the generated embeddings are fit for purpose.

4) *Early Crop Stress and Yield Prediction:*

European agriculture is continuously affected by an increasing frequency of weather extremes [272, 273], which are expected to increase in magnitude and frequency in the near future. How crops are affected by adverse weather conditions strongly depends on the crop's development stage. Systems for timely monitoring of crop phenology are necessary to understand and assess the impact of climate

change on crops [274]. The Sentinel missions [118] contribute significantly to agricultural monitoring with its high temporal and spatial resolution. Despite the development of crop maps and crop yield forecasting activities at the European scale [275], integrating EO and climatological data is needed to capture the effect of increasing weather extremes on crops.

In particular, the early prediction of crop stress or crop yields at field, national, or continental level benefits from Sentinel-1/2 time series. Satellite time series also proved to be a valuable resource to improve crop type classification [276] by capturing the dynamic changes in spectral and temporal signatures of crops during the growing season [277]. Comparatively, methods based on single-date imagery fail to accurately capture variations in phenology, biomass accumulation, and the effects of local conditions.

While crop-related tasks have proven to benefit from multi-modal, multi-date satellite imagery, mobilizing the necessary data and running models on it requires significant computational resources. An efficient compression pipeline would allow the distribution of raw imagery or embeddings to stakeholders currently hindered by bandwidth and hardware requirements. Such pipeline would support a range of actors in the agricultural community: farmers and agricultural organizations (e.g., improved monitoring/forecasting of field damage assessment), the public sector responsible for governing the transition of agriculture, the private sector, including agricultural technology and machinery industries, seed companies and agribusiness retailers, the agrochemical industry, and the insurance sector for risk management, and environmental agencies conducting crop forecasting activities.

VI. PERSPECTIVES & RECOMMENDATIONS

We conclude our literature survey on NC for geospatial analytics by a summary presented through Fig. 21. Novel methodologies to compress EO and ESM data need to cover a wide range of use cases—from single-image compression to embedding long time series, while incorporating information from a plethora of sensors and simulated

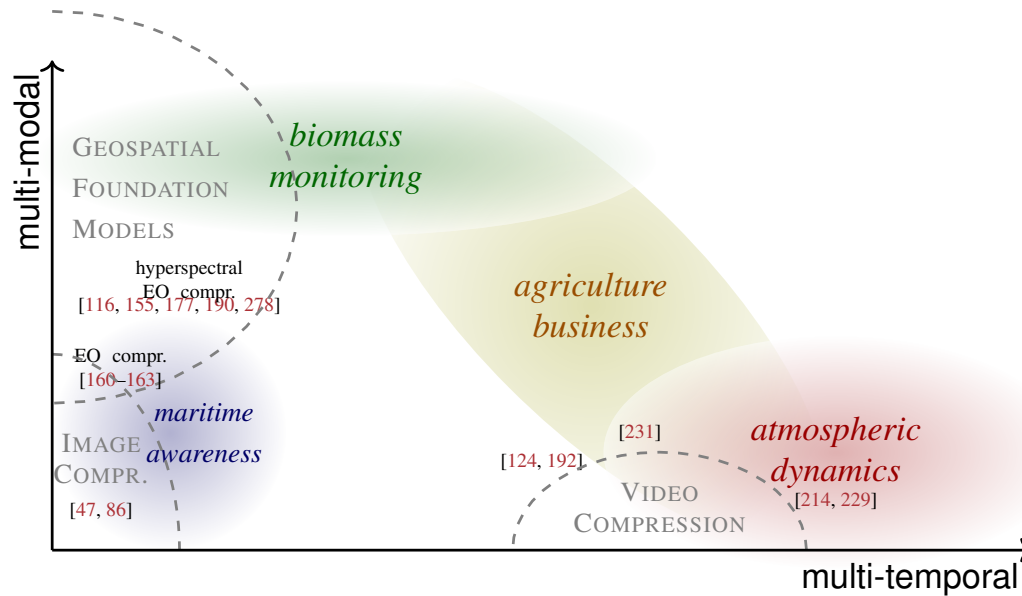


Fig. 21: We represent downstream applications from the perspective of conceptual dimensions relevant for NC methodologies: multi-temporality and multi-modality. References to existing neural data compression literature are positioned with respect to these concepts. A gap in the literature can be observed for the compression of multimodal and multi-temporal data, showing potential for several downstream applications.

physical quantities. However, existing (neural) compression algorithms only partially suit such needs. Image compression offers techniques to compress single-timestamp and single-modality data. Recent developments in FMs work towards joint representation learning of a variety of remote sensors. Video compression provides concepts to summarize time series of images as relevant to ESM applications. However, those algorithms currently lag support for multi-modal inputs.

Earth Observation. Given radar, LiDAR, and multi-spectral sensors operate on various bands of the electromagnetic spectrum, Hyperspectral EO data compression offers a direction toward multi-modal compression. However, a clear deficiency in the current research in the field is the lack of an established methodology to quantitatively compare methods. Datasets meant as a benchmark for learned compression for the wide range of existing

RS modalities are scarce or not used widely enough to enable systematized comparisons. The availability of standard training datasets, and perhaps more importantly, the alignment of the research community on standard evaluation datasets for different EO modalities is critical in enabling improvements in methodology in the field.

Further development of the transform f for NC to better suit different modalities within RS data seems bound to continue to be a fruitful research direction. However, it should be noted that this direction poses the risk of incentivizing continuous small adaptations to methods proposed in the field of natural image compression with limited innovation regarding RS compression. On the other hand, other differentiating data characteristics in this domain seem relatively underexplored as of yet regarding their integration into NC methodologies. For instance, RS data is very rich in metadata. Specifically, geolocation and time of capture may

be informative.

While neural video compression has been successfully employed in natural videos, a major difficulty is its usual reliance on *optical flow* fields and their compression, which proves challenging in scenes with fast motion in uncorrelated directions. The adaptation of these techniques to EO, where such optical flow fields are mostly absent, has the potential to yield great compression ratios where the goal is to transmit a temporal sequence of samples.

Earth System Modeling. Compared to EO, NC for climate model data has received relatively little attention in the academic literature. Existing approaches for compressing the outputs of climate simulators mainly rely on hand-engineered transform coding schemes [219, 222–224, 228]. Data-driven NC offers an attractive alternative [214, 229, 230] but more work is needed to rigorously evaluate their impact on the scientific integrity of the datasets.

A major obstacle to designing and evaluating compression schemes is the lack of agreed-upon quantitative metrics that can be used to reliably assess whether lossy compression schemes preserve all the relevant aspects of the data for climate analysis [204, 235, 237]. Hence, future work should not only focus on the development of new compression schemes, but also new metrics designed along with domain experts to meaningfully evaluate and compare different codecs.

Foundational Models for Geospatial Analytics. A desirable property of foundational models that is currently underexplored for NC would be to tightly integrate well-calibrated uncertainties by design. Model outputs with well-calibrated uncertainties could ease integration not only into downstream tasks based on deep learning but also, and more importantly, become a natural interface to Bayesian methods [279, 280], mechanistic modeling [281], and the existing rich statistics toolbox including significance tests [282]. Furthermore, well-calibrated uncertainties can work as a natural link to physics-based forward simulations in computational science [283], e.g. to tightly integrate radiative transfer models with learning-based approaches in RS [284].

Computing uncertainties along with model outputs would also act as a natural early alert if a given foundational model would be applied to new data far away from the original training distribution. In that way, FMs capable of uncertainty prediction could, for instance, identify strategical training samples within an active learning setting.

Resulting Research Recommendations. To sum up, we list our main research recommendations to advance the domain of NC in the EO and ESM domain.

- 1) Extend NC for temporal and multi-modal EO and ESM data.
- 2) Establish metrics to evaluate utility of NC results for specific applications.
- 3) Compile methodologies and data sets to benchmark the progress of NC algorithms.
- 4) Take advantage of FM development for NC.
- 5) Establish guard rails to estimate compression quality in the context of distribution shift of input data.

Advancing and solving these main issues will be key to establish NC as a widely accepted methodology, disrupting the EO and ESM domain with efficiency in data storage, transfer and computation.

ACKNOWLEDGMENT

This research is carried out as part of the Embed2Scale project and is co-funded by the EU Horizon Europe program under Grant Agreement No. 101131841. Additional funding for this project has been provided by the Swiss State Secretariat for Education, Research and Innovation (SERI) and UK Research and Innovation (UKRI).

REFERENCES

- [1] Copernicus, *Observer: Copernicus climate and atmosphere services provide 2021 climate insights at global and european level*, 2022.
- [2] J. Clauson et al., *Earth observing sensing satellites online compendium: U.s. geological survey digital data*, 2024.
- [3] M. Czerkawski, M. Kluczek, and J. S. Bojanowski, “Global and dense embeddings of earth: Major tom floating in the latent space,” *arXiv preprint arXiv:2412.05600*, 2024.
- [4] *Certain data included herein are derived from clarivate web of science*. 2024. [Online]. Available: <https://webofscience.com>.

- [5] D. Vasisht, J. Shenoy, and R. Chandra, "L2d2: Low latency distributed downlink for leo satellites," *ACM SIGCOMM Conference*, 2021.
- [6] Citation report graphic is derived from clarivate web of science. 2024. [Online]. Available: <https://webofscience.com>.
- [7] P.-S. Yeh *et al.*, "The new CCSDS image compression recommendation," *IEEE Aerospace Conference*, 2005.
- [8] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *NeurIPS*, 2018.
- [9] F. Mentzer *et al.*, "Vct: A video compression transformer," *NeurIPS*, 2022.
- [10] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.
- [11] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," *NeurIPS*, 1994.
- [12] S. Lu *et al.*, "Ai foundation models in remote sensing: A survey," *arXiv preprint arXiv:2408.03464*, 2024.
- [13] S. K. Mukkavilli *et al.*, "Ai foundation models for weather and climate: Applications, design, and implementation," *arXiv preprint arXiv:2309.10808*, 2023.
- [14] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 4, pp. 213–247, 2022.
- [15] R. Bommasani *et al.*, "On the Opportunities and Risks of Foundation Models," *arXiv preprint arXiv:2108.07258*, 2021.
- [16] J. Jakubik *et al.*, "Foundation models for generalist geospatial artificial intelligence," *arXiv preprint arXiv:2310.18660*, 2023.
- [17] C. Lessig, I. Luise, B. Gong, M. Langguth, S. Stadler, and M. Schultz, "AtmoRep: A stochastic model of atmosphere dynamics using large scale representation learning," *arXiv preprint arXiv:2308.13280*, 2023.
- [18] X. Sun *et al.*, "RingMo: A remote sensing foundation model with masked image modeling," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
- [19] F. Liu *et al.*, "RemoteCLIP: A vision language foundation model for remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.
- [20] D. Wang *et al.*, "Advancing plain vision transformer toward remote sensing foundation model," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
- [21] X. Guo *et al.*, "Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery," *CVPR*, 2024.
- [22] D. Hong *et al.*, "Spectralgpt: Spectral foundation model," *arXiv preprint arXiv:2311.07113*, 2023.
- [23] C. Bodnar *et al.*, "Aurora: A foundation model of the atmosphere," *Tech. Rep.*, 2024.
- [24] N. A. A. Braham, C. M. Albrecht, J. Mairal, J. Chanussot, Y. Wang, and X. X. Zhu, "Spectralearth: Training hyperspectral foundation models at scale," *arXiv preprint arXiv:2408.08447*, 2024.
- [25] Y. Yang, S. Mandt, L. Theis, *et al.*, "An introduction to neural data compression," *Foundations and Trends® in Computer Graphics and Vision*, 2023.
- [26] X. Zhang *et al.*, "Boosting neural representations for videos with a conditional decoder," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.18152>.
- [27] H. Kim, M. Bauer, L. Theis, J. R. Schwarz, and E. Dupont, "C3: High-performance and low-complexity neural compression from a single image or video," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [Online]. Available: <https://arxiv.org/abs/2312.02753>.
- [28] Z. Duan, M. Lu, J. Yang, J. He, Z. Ma, and F. Zhu, "Towards backward-compatible continual learning of image compression," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.18862>.
- [29] Z. Zhang, H. Wang, Z. Chen, and S. Liu, "Learned lossless image compression based on bit plane slicing," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024/papers/Zhang_Learned_Lossless_Image_Compression_based_on_Bit_Plane_Slicing_CVPR_2024_paper.pdf.
- [30] A. Khoshkhahtinat, A. Zafari, P. M. Mehta, and N. M. Nasrabadi, "Laplacian-guided entropy model in neural codec with blur-dissipated synthesis," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [Online]. Available: <https://arxiv.org/abs/2403.16258>.
- [31] J. Li, B. Li, and Y. Lu, "Neural video compression with feature modulation," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.17414>.
- [32] Y. Wang, N. A. A. Braham, Z. Xiong, C. Liu, C. M. Albrecht, and X. X. Zhu, "Ssl4eo-s12: A large-scale multimodal, multitemporal dataset for self-supervised learning in earth observation [software and data sets]," *IEEE Geoscience and Remote Sensing Magazine*, 2023.
- [33] W. Li, H. Lee, S. Wang, C.-Y. Hsu, and S. T. Arundel, "Assessment of IBM and NASA's geospatial foundation model in flood inundation mapping," *arXiv preprint arXiv:2309.14500*, 2023.
- [34] A. Furutanpey, Q. Zhang, P. Raith, T. Pfandzelter, S. Wang, and S. Dustdar, "Fool: Addressing the downlink bottleneck in satellite computing with neural feature compression," *arXiv preprint arXiv:2403.16677*, 2024.
- [35] C. Gomes and T. Brunschweiler, "Compressed multi-task embeddings for data-efficient downstream training and inference in earth observation," *arXiv preprint arXiv:2403.17886*, 2024.
- [36] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485–532, 2020.

- [37] K. Brandenburg and G. Stoll, "Iso-mpeg-1 audio: A generic standard for coding of high-quality digital audio," *Journal of the Audio Engineering Society*, 1994.
- [38] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd. Wiley, 2010.
- [39] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012.
- [40] G. K. Wallace, "The jpeg still picture compression standard," *Commun. ACM*, 1991.
- [41] Sonehara, Kawato, Miyake, and Nakane, "Image data compression using a neural network model," *International 1989 Joint Conference on Neural Networks*, 1989.
- [42] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *ICLR*, 2022.
- [43] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end Optimized Image Compression," *ICLR*, 2016.
- [44] S. Mallat, *A wavelet tour of signal processing*. 1999.
- [45] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, 1948.
- [46] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *Machine learning for data science handbook: data mining and knowledge discovery handbook*, 2023.
- [47] A. Van Den Oord, O. Vinyals, et al., "Neural discrete representation learning," *NeurIPS*, 2017.
- [48] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, 1952.
- [49] J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM Journal of Research and Development*, 1979.
- [50] G. E. Blelloch et al., "Introduction to data compression,"
- [51] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, 2004.
- [52] V. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, 2001.
- [53] R. N. Bracewell and R. N. Bracewell, *The Fourier transform and its applications*. 1986.
- [54] I. Daubechies, *Ten lectures on wavelets*. 1992.
- [55] J. Ballé et al., "Nonlinear transform coding," *IEEE Journal of Selected Topics in Signal Processing*, 2020.
- [56] T. Bird, J. Ballé, S. Singh, and P. A. Chou, "3d scene compression through entropy penalized neural representation functions," *2021 Picture Coding Symposium (PCS)*, 2021.
- [57] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *ICLR*, 2018.
- [58] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," *International Conference on Artificial Neural Networks*, 2011.
- [59] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CVPR*, 2016.
- [61] G. Toderici et al., "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.
- [62] Y. Zhu, Y. Yang, and T. Cohen, "Transformer-based transform coding," *ICLR*, 2022.
- [63] H. Li, S. Li, W. Dai, C. Li, J. Zou, and H. Xiong, "Frequency-aware transformer for learned image compression," *ICLR*, 2024.
- [64] R. Yang and S. Mandt, "Lossy image compression with conditional diffusion models," *NeurIPS*, 2024.
- [65] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, 2021.
- [66] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," *CVPR*, 2019.
- [67] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," *CVPR*, 2019.
- [68] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," *CVPR*, 2019.
- [69] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *NeurIPS*, 2020.
- [70] Y. Du, Y. Zhang, H.-X. Yu, J. B. Tenenbaum, and J. Wu, "Neural radiance flow for 4d view synthesis and video processing," *ICCV*, 2021.
- [71] E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet, "Coin: Compression with implicit neural representations," *arXiv preprint arXiv:2103.03123*, 2021.
- [72] Y. Strümpfer, J. Postels, R. Yang, L. V. Gool, and F. Tombari, "Implicit neural representations for image compression," *ECCV*, 2022.
- [73] H. Chen, B. He, H. Wang, Y. Ren, S. N. Lim, and A. Shrivastava, "Nerv: Neural representations for videos," *NeurIPS*, 2021.
- [74] C. Gomes, R. Azevedo, and C. Schroers, "Video compression with entropy-constrained neural representations," *CVPR*, 2023.
- [75] Z. Guo, G. Flamich, J. He, Z. Chen, and J. M. Hernández-Lobato, "Compression with bayesian implicit neural representations," *NeurIPS*, 2023.
- [76] J. He, G. Flamich, Z. Guo, and J. M. Hernández-Lobato, "Recombiner: Robust and enhanced compression with bayesian implicit neural representations," *ICLR*, 2024.
- [77] T. Pham and S. Mandt, "Neural nerf compression," *ICML*, 2024.
- [78] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. 2006.
- [79] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," *2016 Picture Coding Symposium (PCS)*, 2016.
- [80] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic

- neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.
- [81] K. Tsubota and K. Aizawa, "Comprehensive comparisons of uniform quantizers for deep image compression," *2021 IEEE International Conference on Image Processing (ICIP)*, 2021.
- [82] G. Toderici *et al.*, "Full Resolution Image Compression with Recurrent Neural Networks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [83] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," *CVPR*, 2018.
- [84] E. Agustsson *et al.*, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *NeurIPS*, 2017.
- [85] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," *CVPR*, 2018.
- [86] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," *CVPR*, 2021.
- [87] L. Yu *et al.*, "Language model beats diffusion-tokenizer is key to visual generation," *arXiv preprint arXiv:2310.05737*, 2023.
- [88] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, 2018.
- [89] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," *CVPR*, 2018.
- [90] M. Zhu, G. Li, and W. Zhang, "Research on UAV remote sensing multispectral image compression based on CNN," *International Conference on Geology, Mapping and Remote Sensing*, 2022.
- [91] Y. Qian, M. Lin, X. Sun, Z. Tan, and R. Jin, "Entroformer: A transformer-based entropy model for learned image compression," *arXiv preprint arXiv:2202.05492*, 2022.
- [92] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," *International Conference on Image Processing*, 2020.
- [93] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *ICLR*, 2024.
- [94] G. Delétang *et al.*, "Language modeling is compression," *arXiv preprint arXiv:2309.10668*, 2023.
- [95] W. Jiang, J. Yang, Y. Zhai, P. Ning, F. Gao, and R. Wang, "MLIC: Multi-reference entropy model for learned image compression," *ACM International Conference on Multimedia*, 2023.
- [96] G. Zhang, J. Qian, J. Chen, and A. Khisti, "Universal rate-distortion-perception representations for lossy compression," *NeurIPS*, 2021.
- [97] E. Agustsson, D. Minnen, G. Toderici, and F. Mentzer, "Multi-realism image compression with a conditional generator," *CVPR*, 2023.
- [98] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ballé, A. Shrivastava, and G. Toderici, "End-to-end learning of compressible features," *2020 IEEE International Conference on Image Processing (ICIP)*, 2020.
- [99] Y. Dubois, B. Bloem-Reddy, K. Ullrich, and C. J. Maddison, "Lossy compression for lossless prediction," *NeurIPS*, 2021.
- [100] Y. Matsubara, R. Yang, M. Levorato, and S. Mandt, "Supervised compression for resource-constrained edge computing systems," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.
- [101] A. Furtuapey, P. Raith, and S. Dustdar, "Frankensplit: Efficient neural feature compression with shallow variational bottleneck injection for mobile edge computing," *IEEE Transactions on Mobile Computing*, 2024.
- [102] C. M. Bishop, "Latent variable models," in 1998.
- [103] F. Rottensteiner *et al.*, "The isprs benchmark on urban object classification and 3d building reconstruction," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012.
- [104] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," *ICML*, 2021.
- [105] M. Caron *et al.*, "Emerging properties in self-supervised vision transformers," *ICCV*, 2021.
- [106] Y. Cong *et al.*, "SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery," *NeurIPS*, 2022.
- [107] M. J. Smith, L. Fleming, and J. Geach, "Earthpt: A foundation model for earth observation," *NeurIPS Workshop on Tackling Climate Change with Machine Learning*, 2023.
- [108] H. Guo, Z. Liu, H. Jiang, C. Wang, J. Liu, and D. Liang, "Big earth data: A new challenge and opportunity for digital earth's development," *International Journal of Digital Earth*, 2017.
- [109] R. Wilkinson *et al.*, "Environmental impacts of earth observation data in the constellation and cloud computing era," *Science of The Total Environment*, 2024.
- [110] J. Kuester, W. Gross, and W. Middelmann, "An approach to near-lossless hyperspectral data compression using deep autoencoder," *Proceedings of SPIE - The International Society for Optical Engineering*, 2020.
- [111] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *CVPR*, 2009.
- [112] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," *IGARSS*, 2019.
- [113] A. Vali, S. Comai, and M. Matteucci, "Deep Learning for Land Use and Land Cover Classification Based on Hyperspectral and Multispectral Earth Observation Data: A Review," *Remote Sensing*, 2020.
- [114] S. Bajwa, P. Bajcsy, P. Groves, and L. Tian, "Hyperspectral image data mining for band selection in agricultural applications," *Transactions of the ASAE. American Society of Agricultural Engineers*, 2004.
- [115] S.-E. Qian, "Hyperspectral Satellites, Evolution, and Development History," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021.
- [116] S. Mijares i Verdú, J. Ballé, V. Laparra, J. Bartrina-Rapesta, M. Hernández-Cabrero, and J. Serra-Sagristà, "A Scalable Reduced-Complexity Compress-

- sion of Hyperspectral Remote Sensing Images Using Deep Learning,” *Remote Sensing*, 2023.
- [117] M. Zribi *et al.*, “Analysis of L-Band SAR Data for Soil Moisture Estimations over Agricultural Areas in the Tropics,” *Remote Sensing*, 2019.
- [118] M. Berger, J. Moreno, J. A. Johannessen, P. F. Levelt, and R. F. Hanssen, “Esa’s sentinel missions in support of earth system science,” *Remote sensing of environment*, 2012.
- [119] S. Xiang, Q. Liang, and L. Fang, “Discrete wavelet transform-based gaussian mixture model for remote sensing image compression,” *IEEE Transactions on Geoscience and Remote Sensing*, 2023.
- [120] L. Miller, C. Pelletier, and G. I. Webb, “Deep Learning for Satellite Image Time Series Analysis: A Review,” *IEEE Geoscience and Remote Sensing Magazine*, 2024.
- [121] P. Liu, “A survey of remote-sensing big data,” *Frontiers in Environmental Science*, 2015.
- [122] K. Lieberman, J. Diffenderfer, C. Godfrey, and B. Kailkhura, *Neural image compression: Generalization, robustness, and spectral biases*, 2023. arXiv: 2307.08657 [eess.IV]. [Online]. Available: <https://arxiv.org/abs/2307.08657>.
- [123] A. Furutanpey, Q. Zhang, P. Raith, T. Pfandzelter, S. Wang, and S. Dustdar, “Fool: Addressing the downlink bottleneck in satellite computing with neural feature compression,” *arXiv preprint arXiv:2403.16677*, 2024.
- [124] K. Du, Y. Cheng, P. Olsen, S. Noghabi, R. Chandra, and J. Jiang, “Earth+: On-board satellite imagery compression leveraging historical earth observations,” *arXiv preprint arXiv:2403.11434*, 2024.
- [125] Y. Li, M. Li, C. Li, and Z. Liu, “Forest aboveground biomass estimation using landsat 8 and sentinel-1a data with machine learning algorithms,” *Scientific Reports*, 2020.
- [126] A. C. Rodríguez, S. D’Aronco, K. Schindler, and J. D. Wegner, “Mapping oil palm density at country scale: An active learning approach,” *Remote Sensing of Environment*, 2021.
- [127] Copernicus, *Copernicus european flood awareness system*, 2024.
- [128] R. Ramanath, W. Snyder, Y. Yoo, and M. Drew, “Color image processing pipeline,” *IEEE Signal Processing Magazine*, vol. 22, Jan. 2005. [Online]. Available: <http://ieeexplore.ieee.org/document/1407713/>.
- [129] N. Ahmed, T. Natarajan, and K. Rao, “Discrete cosine transform,” *IEEE Transactions on Computers*, 1974.
- [130] H. Lee, N. Younan, and R. King, “Hyperspectral image cube compression combining JPEG-2000 and spectral decorrelation,” *IGARSS*, 2002.
- [131] G. Yu, T. Vladimirova, and M. N. Sweeting, “Image compression systems on board satellites,” *Acta Astronautica*, 2009.
- [132] I. Hacıhaliloglu and M. Karta, “DCT and DWT based image compression in remote sensing images,” *IEEE Antennas and Propagation Society Symposium*, 2004.
- [133] P. Hou, M. Petrou, C. Underwood, and A. Hojjatoleslami, “Improving JPEG performance in conjunction with cloud editing for remote sensing applications,” *IEEE Transactions on Geoscience and Remote Sensing*, 2000.
- [134] J. Gonzalez-Conejero, J. Bartrina-Rapesta, and J. Serra-Sagrista, “JPEG2000 encoding of remote sensing multispectral images with no-data regions,” *IEEE Geoscience and Remote Sensing Letters*, 2010.
- [135] S. Yang, M. Wang, and L. Jiao, “Compression of remote sensing images based on ridgelet and neural network,” *Advances in Neural Networks*, 2005.
- [136] E. Machairas and N. Kranitis, “A 13.3 gbps 9/7m discrete wavelet transform for CCSDS 122.0-b-1 image data compression on a space-grade SRAM FPGA,” *Electronics*, 2020.
- [137] F. Garcia-Vilchez and J. Serra-Sagrista, “Extending the CCSDS recommendation for image data compression for remote sensing scenarios,” *IEEE Transactions on Geoscience and Remote Sensing*, 2009.
- [138] D. Markman and D. Malah, “Hyperspectral image coding using 3d transforms,” *International Conference on Image Processing*, 2001.
- [139] S. Lim, K. Sohn, and C. Lee, “Compression for hyperspectral images using three dimensional wavelet transform,” *IGARSS*, 2001.
- [140] P. Luigi Dragotti, G. Poggi, and A. Ragozini, “Compression of multispectral images by three-dimensional SPIHT algorithm,” *IEEE Transactions on Geoscience and Remote Sensing*, 2000.
- [141] Q. Du and J. E. Fowler, “Hyperspectral image compression using JPEG2000 and principal component analysis,” *IEEE Geoscience and Remote Sensing Letters*, 2007.
- [142] Q. Du and J. E. Fowler, “Low-complexity principal component analysis for hyperspectral image compression,” *The International Journal of High Performance Computing Applications*, 2008.
- [143] J. Li and Z. Liu, “Multispectral transforms using convolution neural networks for remote sensing multispectral image compression,” *Remote Sensing*, 2019.
- [144] L. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, 1966.
- [145] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, “Compression of hyperspectral remote sensing images by tensor approach,” *Neurocomputing special issue Advances in Self-Organizing Maps*, 2015.
- [146] M. Zhang, B. Du, L. Zhang, and X. Li, “A low-rank tensor decomposition based hyperspectral image compression algorithm,” 2016.
- [147] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao, “PLTD: Patch-based low-rank tensor decomposition for hyperspectral images,” *IEEE Transactions on Multimedia*, 2017.
- [148] N. Li and B. Li, “Tensor completion for on-board compression of hyperspectral images,” *IEEE International Conference on Image Processing*, 2010.
- [149] A. Karami, M. Yazdi, A. Zolghadri, and A. Zolghadri, “Hyperspectral image compression based on tucker decomposition and discrete cosine transform,” (2010).
- [150] A. Karami, M. Yazdi, and G. Mercier, “Hyperspectral image compression based on tucker decomposition and wavelet transform,” *2011 3rd Workshop on Hyperspec-*

- tral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2011.
- [151] A. Karami, M. Yazdi, and G. Mercier, "Compression of hyperspectral images using discrete wavelet transform and Tucker decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2012.
 - [152] N. D. Sidiropoulos and A. Kyriillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Processing Letters*, 2012.
 - [153] J. Li, F. Xing, and Z. You, "Compression of multispectral images with comparatively few bands using post-transform Tucker decomposition," *Mathematical Problems in Engineering*, 2014.
 - [154] J. Li and Z. Liu, "Compression of hyper-spectral images using an accelerated nonnegative tensor decomposition," *Open Physics*, 2017.
 - [155] L. Zhang, T. Pan, J. Liu, and L. Han, "Compressing hyperspectral images into multilayer perceptrons using fast-time hyperspectral neural radiance fields," *IEEE Geoscience and Remote Sensing Letters*, 2024.
 - [156] R. Kwok and W. Johnson, "Block adaptive quantization of magellan sar data," *IEEE Transactions on Geoscience and Remote Sensing*, 1989.
 - [157] E. Attema *et al.*, "Flexible dynamic block adaptive quantization for sentinel-1 sar missions," *IEEE Geoscience and Remote Sensing Letters*, 2010.
 - [158] M. Martone, "Onboard quantization for interferometric and multichannel synthetic aperture radar (sar) systems," Ph.D. dissertation, Karlsruhe Institut für Technologie (KIT), 2019.
 - [159] E. J. Candès and D. L. Donoho, "Ridgelets: A key to higher-dimensional intermittency?" *Philosophical Transactions of the Royal Society of London*, 1999.
 - [160] V. Alves de Oliveira *et al.*, "Reduced-complexity end-to-end variational autoencoder for on board satellite image compression," *Remote Sensing*, 2021.
 - [161] F. Kong, K. Hu, Y. Li, D. Li, and S. Zhao, "Spectral-spatial feature partitioned extraction based on CNN for multispectral image compression," *Remote Sensing*, 2021.
 - [162] T. Cao, N. Zhang, S. Zhao, K. Hu, and K. Wang, "Spectral-spatial feature completely separated extraction with tensor CNN for multispectral image compression," *Lecture Notes in Electrical Engineering*, 2022.
 - [163] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.
 - [164] Q. Xu *et al.*, "Synthetic aperture radar image compression based on a variational autoencoder," *IEEE Geoscience and Remote Sensing Letters*, 2022.
 - [165] G. Pilikos, M. Azcueta, R. Camarero, and N. Floury, "Raw data compression for synthetic aperture radar using deep learning," *International Workshop on On-board Payload Data Compression*, 2022.
 - [166] R. M. Asiyabi, A. Anghel, P. Rizzoli, M. Martone, and M. Datcu, "Complex-valued autoencoder for multipolarization slc sar data compression with side information," *IGARSS*, 2023.
 - [167] N. Gollin, M. Martone, G. Krieger, and P. Rizzoli, "AI-based performance-optimized quantization for future sar systems," *IGARSS*, 2023.
 - [168] M. Martone, N. Gollin, P. Rizzoli, and G. Krieger, "Performance-optimized quantization for sar and insar applications," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
 - [169] P. Maharjan and Z. Li, "Complex-valued sar image compression: A novel approach for amplitude and phase recovery," *International Conference on Visual Communications and Image Processing*, 2023.
 - [170] S. Xiang and Q. Liang, "Remote sensing image compression with long-range convolution and improved non-local attention model," *Signal Processing*, 2023.
 - [171] F. Kong, S. Zhao, Y. Li, and D. Li, "End-to-end multispectral image compression framework based on adaptive multiscale feature extraction," *Journal of Electronic Imaging*, 2021.
 - [172] C. Fu and B. Du, "Remote sensing image compression based on the multiple prior information," *Remote Sensing*, 2023.
 - [173] J. Gao, Q. Teng, X. He, Z. Chen, and C. Ren, "Mixed entropy model enhanced residual attention network for remote sensing image compression," *Neural Processing Letters*, 2023.
 - [174] C. Fu, B. Du, and L. Zhang, "Sar image compression based on multi-resblock and global context," *IEEE Geoscience and Remote Sensing Letters*, 2023.
 - [175] Z. Di, X. Chen, Q. Wu, J. Shi, Q. Feng, and Y. Fan, "Learned compression framework with pyramidal features and quality enhancement for SAR images," *IEEE Geoscience and Remote Sensing Letters*, 2022.
 - [176] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," *CVPR*, 2020.
 - [177] D. Anuradha, G. Sekhar, A. Mishra, P. Thapar, Y. El-Ebiary, and M. Syamala, "Efficient compression for remote sensing: Multispectral transform and deep recurrent neural networks for lossless hyper-spectral image," *International Journal of Advanced Computer Science and Applications*, 2024.
 - [178] S. Xiang and Q. Liang, "Remote sensing image compression based on high-frequency and low-frequency components," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.
 - [179] W. Ye, W. Lei, W. Zhang, T. Yu, and X. Feng, "GF-SCompNet: Remote sensing image compression network based on global feature-assisted segmentation," *Multimedia Tools and Applications*, 2024.
 - [180] J. Deng and L. Huang, "Synthetic aperture radar image compression based on low-frequency rejection and quality map guidance," *Remote Sensing*, 2024.
 - [181] J. Li, Y. Ye, and B. Liu, "An intelligent image compression method based on generative adversarial networks for satellites," *International Symposium on Computer Engineering and Intelligent Communications*, 2023.
 - [182] S. Zhao, S. Yang, Z. Liu, Z. Feng, and X. Liu, "Sparse flow adversarial model for robust image compression," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

- [183] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
- [184] S. Zhao, S. Yang, J. Gu, Z. Liu, and Z. Feng, "Symmetrical lattice generative adversarial network for remote sensing images compression," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021.
- [185] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *ICCV*, 2017.
- [186] Y. Strümpfer, J. Postels, R. Yang, L. V. Gool, and F. Tombari, "Implicit neural representations for image compression," *ECCV*, 2022.
- [187] X. Li, B. Sun, J. Liao, and X. Zhao, "Remote sensing image compression method based on implicit neural representation," *International Conference on Computing and Pattern Recognition*, 2023.
- [188] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *CVPR*, 2022.
- [189] Q. Wu, R. Zhang, and F. Wang, "Hyperspectral data compression using double sparsity model," *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2015.
- [190] H. Wang and T. Celik, "Sparse representation-based hyperspectral data processing: Lossy compression," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017.
- [191] A. Ertem, A. C. Karaca, O. Urhan, and M. K. Güllü, "Superpixel based compression of hyperspectral image with modified dictionary and sparse representation," *International Journal of Remote Sensing*, 2020.
- [192] X. Wang, R. Hu, Z. Wang, and J. Xiao, "Virtual background reference frame based satellite video coding," *IEEE Signal Processing Letters*, 2018.
- [193] G. Meoni, R. Prete, F. Serva, A. Beusscher, O. Colin, and N. Longépé, "Unlocking the use of raw multispectral earth observation imagery for onboard artificial intelligence," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.
- [194] G. Guerri, F. Del Frate, and G. Schiavon, "Artificial intelligence based on-board image compression for the ϕ -sat-2 mission," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023.
- [195] S. Lu *et al.*, "Onboard AI for Fire Smoke Detection Using Hyperspectral Imagery: An Emulation for the Upcoming Kanyini Hyperscout-2 Mission," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, 2024.
- [196] B. Stevens and S. Bony, "What are climate models missing?" *science*, 2013.
- [197] T. Palmer, "Climate forecasting: Build high-resolution global climate models," *Nature*, 2014.
- [198] T. Schneider *et al.*, "Climate goals and computing the future of clouds," *Nature Climate Change*, 2017.
- [199] B. Stevens *et al.*, "Dyamond: The dynamics of the atmospheric general circulation modeled on non-hydrostatic domains," *Progress in Earth and Planetary Science*, 2019.
- [200] P. Bauer, B. Stevens, and W. Hazeleger, "A digital twin of earth for the green transition," *Nature Climate Change*, 2021.
- [201] J. E. Kay *et al.*, "The community earth system model (cesm) large ensemble project: A community resource for studying climate change in the presence of internal climate variability," *Bulletin of the American Meteorological Society*, 2015.
- [202] H. Hersbach *et al.*, "The era5 global reanalysis," *Quarterly Journal of the Royal Meteorological Society*, 2020.
- [203] J. M. Kunkel, M. Kuhn, and T. Ludwig, "Exascale storage systems: An analytical study of expenses," *Supercomputing frontiers and innovations*, 2014.
- [204] F. Cappello, S. Di, and A. M. Gok, "Fulfilling the promises of lossy compression for scientific applications," *Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI: 17th Smoky Mountains Computational Sciences and Engineering Conference, SMC 2020, Oak Ridge, TN, USA, August 26-28, 2020, Revised Selected Papers 17*, 2020.
- [205] B. Eaton *et al.*, "Netcdf climate and forecast (cf) metadata conventions," URL: <http://cfconventions.org/Data/cf-conventions/cf-conventions-1.8/cf-conventions.pdf>, 2003.
- [206] D. Cherian *et al.*, *cf_xarray*, 2023. [Online]. Available: <https://github.com/xarray-contrib/cf-xarray>.
- [207] C. D. Roberts, R. Senan, F. Molteni, S. Boussetta, M. Mayer, and S. P. Keeley, "Climate model configurations of the ecwf integrated forecasting system (ecmf-ifs cycle 43r1) for highresmp," *Geoscientific model development*, 2018.
- [208] G. Danabasoglu *et al.*, "The community earth system model version 2 (cesm2)," *Journal of Advances in Modeling Earth Systems*, 2020.
- [209] J. H. Jungclaus *et al.*, "The icon earth system model version 1.0," *Journal of Advances in Modeling Earth Systems*, 2022.
- [210] C. Schär *et al.*, "Kilometer-scale climate models: Prospects and challenges," *Bulletin of the American Meteorological Society*, 2020.
- [211] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, 2004.
- [212] T. Palmer, A. Döring, and G. Seregin, "The real butterfly effect," *Nonlinearity*, 2014.
- [213] T. Palmer and B. Stevens, "The scientific challenge of understanding and estimating climate change," *Proceedings of the National Academy of Sciences*, 2019.
- [214] P. Mirowski *et al.*, "Neural compression of atmospheric states," *arXiv preprint arXiv:2407.11666*, 2024.
- [215] S. Rasp *et al.*, "Weatherbench 2: A benchmark for the next generation of data-driven global weather models," *Journal of Advances in Modeling Earth Systems*, 2024.
- [216] G. Mandorli and C. J. Stubenrauch, "Assessment of object-based indices to identify convective organization," *Geoscientific Model Development*, 2024.
- [217] L. J. Freischem, P. Weiss, H. M. Christensen, and P. Stier, "Multifractal analysis for evaluating the repre-

- sensation of clouds in global kilometer-scale models,” *Geophysical Research Letters*, 2024.
- [218] A. H. Baker, H. Xu, D. M. Hammerling, S. Li, and J. P. Clyne, “Toward a multi-method approach: Lossy data compression for climate simulation data,” *High Performance Computing*, 2017.
- [219] M. Klöwer, M. Razinger, J. J. Dominguez, P. D. Düben, and T. N. Palmer, “Compressing atmospheric data into its real information content,” *Nature Computational Science*, 2021.
- [220] A. H. Baker, A. Pinard, and D. M. Hammerling, “On a structural similarity index approach for floating-point data,” *IEEE Transactions on Visualization and Computer Graphics*, 2023.
- [221] T. Mauritsen *et al.*, “Developments in the mpi-m earth system model version 1.2 (mpi-esml. 2) and its response to increasing co₂,” *Journal of Advances in Modeling Earth Systems*, 2019.
- [222] P. Lindstrom, “Fixed-rate compressed floating-point arrays,” *IEEE transactions on visualization and computer graphics*, 2014.
- [223] X. Liang *et al.*, “Sz3: A modular framework for composing prediction-based error-bounded lossy compressors,” *IEEE Transactions on Big Data*, 2022.
- [224] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola, “Tthresh: Tensor compression for multidimensional visual data,” *IEEE transactions on visualization and computer graphics*, 2019.
- [225] F. Váña *et al.*, “Single precision in weather forecasting models: An evaluation with the ifs,” *Monthly Weather Review*, 2017.
- [226] O. Tintó Prims *et al.*, “How to use mixed precision in ocean models: Exploring a potential reduction of numerical precision in nemo 4.0 and roms 3.6,” *Geoscientific Model Development*, 2019.
- [227] A. Inness *et al.*, “The cams reanalysis of atmospheric composition,” *Atmospheric Chemistry and Physics*, 2019.
- [228] C. S. Zender, “Bit grooming: Statistically accurate precision-preserving quantization with compression, evaluated in the netcdf operators (nco, v4. 4.8+),” *Geoscientific Model Development*, 2016.
- [229] L. Huang and T. Hoefer, “Compressing multidimensional weather and climate data into neural networks,” *ICLR*, 2023.
- [230] T. Han, S. Guo, W. Xu, L. Bai, *et al.*, “Cra5: Extreme compression of era5 for portable global climate and weather research via an efficient variational transformer,” *arXiv preprint arXiv:2405.03376*, 2024.
- [231] E. Dupont, H. Loya, M. Alizadeh, A. Golinski, Y. W. Teh, and A. Doucet, “Coin++: Neural compression across modalities,” *Transactions on Machine Learning Research*, 2022.
- [232] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” *NeurIPS*, 2007.
- [233] M. Tancik *et al.*, “Fourier features let networks learn high frequency functions in low dimensional domains,” *NeurIPS*, 2020.
- [234] K. M. Gorski *et al.*, “Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere,” *The Astrophysical Journal*, 2005.
- [235] A. H. Baker *et al.*, “Evaluating lossy data compression on climate simulation data within a large ensemble,” *Geoscientific Model Development*, 2016.
- [236] A. Poppick, J. Nardi, N. Feldman, A. Baker, and D. Hammerling, “A statistical analysis of compressed climate model data,” *Proc. DRBSD*, 2018.
- [237] R. Underwood, J. Bessac, S. Di, and F. Cappello, “Understanding the effects of modern compressors on the community earth science model,” *International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD)*, 2022.
- [238] H. Sather, A. Pinard, A. H. Baker, and D. M. Hammerling, “What can real information content tell us about compressing climate model data?” *International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD)*, 2022.
- [239] P. Kansakar and F. Hossain, “A review of applications of satellite earth observation data for global societal benefit and stewardship of planet earth,” *Space Policy*, 2016.
- [240] S. Lu *et al.*, “Ai foundation models in remote sensing: A survey,” *arXiv preprint arXiv:2408.03464*, 2024.
- [241] B. Blumenstiel, V. Moor, R. Kienzler, and T. Brunschweiler, “Multi-spectral remote sensing image retrieval using geospatial foundation models,” *arXiv preprint arXiv:2403.02059*, 2024.
- [242] M. Allen *et al.*, “Fewshot learning on global multimodal embeddings for earth observation tasks,” *arXiv preprint arXiv:2310.00119*, 2023.
- [243] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, “Google earth engine: Planetary-scale geospatial analysis for everyone,” *Remote Sensing of Environment*, 2017.
- [244] M. Schramm *et al.*, “The openeo api—harmonising the use of earth observation cloud services using virtual data cube functionalities,” *Remote Sensing*, 2021.
- [245] Cloud-Native Geospatial Foundation, *Survey for current practices storing embeddings in geoparquet*, 2024. [Online]. Available: <https://github.com/cloudnativegeo/geo-embeddings-survey>.
- [246] M. D. Wilkinson *et al.*, “The fair guiding principles for scientific data management and stewardship,” *Scientific data*, 2016.
- [247] J. Aslan, K. Mayers, J. G. Koomey, and C. France, “Electricity intensity of internet data transmission: Untangling the estimates,” *Journal of industrial ecology*, 2018.
- [248] Z. Liu *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows,” *ICCV*, 2021.
- [249] S. Kesselheim *et al.*, “Juwels booster—a supercomputer for large-scale ai research,” *High Performance Computing: ISC High Performance Digital 2021 International Workshops*, 2021.
- [250] J. Koomey and E. Masanet, “Does not compute: Avoiding pitfalls assessing the internet’s energy and carbon impacts,” *Joule*, 2021.
- [251] M. Santoro and O. Cartus, “Esa biomass climate change initiative (biomass cci): Global datasets of forest above-ground biomass for the years 2010, 2017 and

- 2018, v2,” *Centre for Environmental Data Analysis*, 2021.
- [252] P. De Frenne *et al.*, “Global buffering of temperatures under forest canopies,” *Nature Ecology & Evolution*, 2019.
- [253] N. T. Hoang and K. Kanemoto, “Mapping the deforestation footprint of nations reveals growing threat to tropical forests,” *Nature Ecology & Evolution*, 2021.
- [254] P. Manning *et al.*, “Redefining ecosystem multifunctionality,” *Nature ecology & evolution*, 2018.
- [255] W. Jetz *et al.*, “Essential biodiversity variables for mapping and monitoring species populations,” *Nature ecology & evolution*, 2019.
- [256] M. Migliavacca *et al.*, “The three major axes of terrestrial ecosystem function,” *Nature*, 2021.
- [257] B. Brede *et al.*, “Non-destructive estimation of individual tree biomass: Allometric models, terrestrial and uav laser scanning,” *Remote Sensing of Environment*, 2022.
- [258] P. Potapov *et al.*, “Mapping global forest canopy height through integration of gedi and landsat data,” *Remote Sensing of Environment*, 2021.
- [259] R. Dubayah *et al.*, “The global ecosystem dynamics investigation: High-resolution laser ranging of the earth’s forests and topography,” *Science of remote sensing*, 2020.
- [260] N. Lang, W. Jetz, K. Schindler, and J. D. Wegner, “A high-resolution canopy height model of the earth,” *Nature Ecology & Evolution*, 2023.
- [261] J. Tolan *et al.*, “Very high resolution canopy height maps from rgb imagery using self-supervised vision transformer and convolutional decoder trained on aerial lidar,” *Remote Sensing of Environment*, 2024.
- [262] C. Renier, M. Vandromme, P. Meyfroidt, V. Ribeiro, N. Kalischek, and E. K. Zu Ermgassen, “Transparency, traceability and deforestation in the ivorian cocoa supply chain,” *Environmental Research Letters*, 2023.
- [263] N. Kalischek *et al.*, “Cocoa plantations are associated with deforestation in côte d’ivoire and ghana,” *Nature Food*, 2023.
- [264] G. Margarit, J. A. Barba Milanés, and A. Tabasco, “Operational ship monitoring system based on synthetic aperture radar processing,” *Remote Sensing*, 2009.
- [265] C. Zhu, H. Zhou, R. Wang, and J. Guo, “A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features,” *IEEE Transactions on geoscience and remote sensing*, 2010.
- [266] R. Pelich, N. Longépé, G. Mercier, G. Hajdich, and R. Garelo, “Ais-based evaluation of target detectors and sar sensors characteristics for maritime surveillance,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2014.
- [267] J. Tang, C. Deng, G.-B. Huang, and B. Zhao, “Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine,” *IEEE transactions on geoscience and remote sensing*, 2014.
- [268] G. Mooers *et al.*, “Comparing storm resolving models and climates via unsupervised machine learning,” *Scientific Reports*, 2023.
- [269] T. Kurihana, E. J. Moyer, and I. T. Foster, “Aicca: Ai-driven cloud classification atlas,” *Remote Sensing*, 2022.
- [270] L. Denby, “Discovering the Importance of Mesoscale Cloud Organization Through Unsupervised Classification,” *Geophysical Research Letters*, 2020.
- [271] L. Denby, “Charting the Realms of Mesoscale Cloud Organisation using Unsupervised Learning,” *arXiv preprint arXiv:2309.08567*, 2023.
- [272] D. Beillouin, B. Schauburger, A. Bastos, P. Ciais, and D. Makowski, “Impact of extreme weather conditions on european crop production in 2018,” *Philosophical Transactions of the Royal Society B*, 2020.
- [273] J. Hristov *et al.*, “Analysis of climate change impacts on eu agriculture by 2050,” *Publications Office of the European Union, Luxembourg, Luxembourg*, 2020.
- [274] M. A. White, F. Hoffman, W. W. Hargrove, and R. R. Nemani, “A global framework for monitoring phenological responses to climate change,” *Geophysical Research Letters*, 2005.
- [275] M. Van der Velde and L. Nisini, “Performance of the mars-crop yield forecasting system for the european union: Assessing accuracy, in-season, and year-to-year improvements from 1993 to 2015,” *Agricultural Systems*, 2019.
- [276] F. Vuolo, M. Neuwirth, M. Immitzer, C. Atzberger, and W.-T. Ng, “How much does multi-temporal sentinel-2 data improve crop type classification?” *International Journal of Applied Earth Observation and Geoinformation*, 2018.
- [277] M. Russwurm and M. Korner, “Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images,” *CVPR Workshop*, 2017.
- [278] M. H. P. Fuchs and B. Demir, “Hyspecnet-11k: A large-scale hyperspectral dataset for benchmarking learning-based hyperspectral image compression methods,” *IGARSS*, 2023.
- [279] A. G. Wilson, “The case for bayesian deep learning,” *arXiv preprint arXiv:2001.10995*, 2020.
- [280] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. 2006.
- [281] A. Kraslawski and I. Turunen, *23rd European Symposium on Computer Aided Process Engineering*. 2013.
- [282] E. L. Lehmann, J. P. Romano, and G. Casella, *Testing statistical hypotheses*. 1986.
- [283] M. Reichstein *et al.*, “Deep learning and process understanding for data-driven earth system science,” *Nature*, 2019.
- [284] Y. Zérah, S. Valero, and J. Inglada, “Physics-driven probabilistic deep learning for the inversion of physical models with application to phenological parameter retrieval from satellite times series,” *IEEE Transactions on Geoscience and Remote Sensing*, 2023.