EMBED 2SCALE

Ref. Ares(2024)4688163 - 28/06/2024

Earth Observation & Weather Data Federation with AI Embeddings

D2.1:

Report on State-of-the-art in AI compressors

A Review

Revision: v 0.1

D2.1: Report on State-of-the-art in AI Compressors

A review

Work package	WP2
Task	1
Due date	June 30, 2024
Submission date	June 28, 2024
Deliverable lead	DLR (Conrad M Albrecht)
Version	0.1
Authors (random order)	Carlos Gomes, Matej Batič, Tim Reichelt, Isabelle Wittmann, Conrad M Albrecht, Stefano Maurogiovanni, Jan Dirk Wegner, Gabriele Cavallaro, Johannes Jakubik, Jonas Hurst, Philip Stier, Rocco Sedona, Thomas Brunschwiler, Stefan Kesselheim, Damien Robert, Kennedy Adriko, Miguel A. Belenguer-Plomer, Michele Lazzarini
Reviewers	
Abstract	This report provides a comprehensive literature review of compression algorithms based on artificial neural networks, termed “Neural Compression”. Beyond generic methodologies, we present and classify works from the domain of Earth observation, and outline the opportunities for neural compression for Earth system modeling. We put the literature into the context of Embed2Scale’s agenda wrt. the downstream applications of global biomass estimation, agriculture monitoring, maritime awareness, and climate modeling to identify future directions of research and innovation for the project.
Keywords	neural compression, Earth observation, Earth system modelling, geospatial foundation models, energy-efficient deep learning, federated geospatial data platforms

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	June 18, 2024	initial draft	- **DLR**: Conrad M Albrecht - **IBM**: Carlos Gomes, Isabelle Wittmann, Johannes Jakubik, Thomas Brunschwiler - **UZH**: Damien Robert, Jan D Wegner - **UOXF**: Tim Reichelt, Philip Stier - **SIN**: Matej Batič - **FZJ**: Stefano Maurogiovanni, Kennedy Adriko, Rocci Sedona, Stefan Kesselheim, Gabriele Cavallaro - **WWU**: Jonas Hurst - **SATCEN**: Miguel A Belenguer-Plomer, Michele Lazzarini

Grant Agreement No: 101131841 | **Topic:** HORIZON-EUSPA-2022-SPACE-02-55
Call: HORIZON-EUSPA-2022-SPACE | **Type of action:** HORIZON-RIA

DISCLAIMER

Funded by the European Union
EUSPA European Union Agency for the Space Programme
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI
UK Research and Innovation

Embed2Scale (Earth Observation & Weather Data Federation With AI Embeddings) project is funded by the EU's Horizon Europe program under Grant Agreement number 101131841. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. This work has also received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI) and the UK Research and Innovation (UKRI).

COPYRIGHT NOTICE

Project funded by the European Commission in the Horizon Europe Programme	
Nature of the deliverable*:	R
Dissemination Level	
SEN	Sensitive, limited under the conditions of the Grant Agreement **note**: basis of a scientific publication in preparation for submission to IEEE GRSM

* *R: Document, report (excluding the periodic and final reports)*
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
DATA: Data sets, microdata, etc.
DMP: Data management plan
ETHICS: Deliverables related to ethics issues.
SECURITY: Deliverables related to security issues
OTHER: Software, technical diagram, algorithms, models, etc.

Neural Compression for Geospatial Analytics: A Review

Carlos Gomes*[1], Matej Batič[3], Tim Reichelt[5], Isabelle Wittmann[1], Conrad M Albrecht[2], Stefano Maurogiovanni[4], Jan Dirk Wegner[6], Gabriele Cavallaro[4], Jonas Hurst[7], Philip Stier[5], Rocco Sedona[4], Thomas Brunschwiler[1], Stefan Kesselheim[4], Damien Robert[6], Kennedy Adriko[4], Miguel A. Belenguer-Plomer[8], and Michele Lazzarini[8]

[1]IBM Research - Europe, Switzerland
[2]German Aerospace Center, Germany
[3]Sinergise Solutions, Slovenia
[4]Forschungszentrum Jülich, Germany
[5]Oxford University, United Kingdom
[6]University of Zurich, Switzerland
[7]Münster University, Germany
[8]European Union Satellite Centre, Spain

Friday 28th June, 2024

*Leading author. All other authors in random order according to Python code snippet `import random; a=['MB','TR','TB','CMA',...]; random.shuffle(a); a=['CG']+a`.

Abstract

Over the past decads 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 issues, operating at high resolutions in space and time, producing petabytes of data per simulated day. The problem of how to efficiently store and compress data has long been studied. In recent years, approaches combining deep learning and information theory for data compression have shown great promise, spawning the field of neural compression. The advent of self-supervised learning (SSL) and foundation models (FM) boosted progress in methodologies to efficiently distill representations from vast amounts of unlabeled data. EO and ESMs provide a promising playground for neural compression, with a wealth of unlabeled data readily available.

In this review, we share an overview of current developments in the field of neural compression applied to geospatial analytics. We introduce the main concepts in neural compression and its seminal works in traditional applications to image and video compression domains. We then review works that have applied neural compression to a wide variety of EO modalities. An additional section covers the currently sparse efforts to utilize neural compression for ESMs. In addition, we connect these developments of neural compression for EO to foundation models to highlight 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.

Contents

1 Introduction **4**
1.1 Motivation & Approach . . . 4
1.2 Traditional vs. Neural Compression . . . 6
1.3 Foundation Models for Remote Sensing & Atmospheric Modelling . . . 8

2 (Lossy) Neural Compression **9**
2.1 Background . . . 9
2.1.1 Lossy Compression . . . 9
2.1.2 Transform coding . . . 10
2.2 Compression Taxonomy . . . 13
2.3 Methods in Neural Compression . . . 14
2.3.1 Transforms . . . 14
2.3.2 Quantization Strategies . . . 15
2.3.3 Entropy Models . . . 18
2.3.4 Optimization Objectives . . . 19

3 Neural Compression for Remote Sensing **21**
3.1 Challenges . . . 21
3.1.1 Data Characteristics . . . 21
3.1.2 Data acquisition and application . . . 23
3.2 Classification of Compression Methodologies . . . 24
3.2.1 Traditional approaches . . . 24
3.2.2 Neural approaches . . . 25
3.2.3 Dictionary learning . . . 31
3.3 Summary . . . 32

4 Neural Compression for Climate Data **33**
4.1 Challenges . . . 33
4.1.1 Data Characteristics . . . 33
4.1.2 Data Acquisition and Application . . . 34
4.2 Classification of Compression Methodologies . . . 34

5 Neural Compression: Implementation & Application **36**
5.1 Neural Compression for Geospatial Analytics Platforms . . . 36
5.2 Cost- and Energy-Efficiency & Latency . . . 38
5.3 Democratization for Applications . . . 40
5.3.1 Global Vegetation Structure Analysis . . . 41
5.3.2 Ship Detection for Maritime Awareness . . . 42
5.3.3 Climate and Air Pollution Prediction . . . 42
5.3.4 Early Crop Stress and Yield Prediction . . . 43

6 Perspectives & Recommendations **44**

Figure 1: Domain-specific shares in publications in neural compression methodologies from years 2000 through 2024, source: query to the *Web of Science* [200].

1 Introduction

1.1 Motivation & Approach

Earth Observation (EO) is the process of capturing data about the Earth's surface and atmosphere, carried out through instruments on board satellites, airborne vehicles, or ground stations. The instruments used for collection vary considerably in their specifications, capturing different aspects of the Earth using different spectral regions and resolutions, Ground Sample Distances (GSMs), and revisit times, amongst many other factors. Due to their constant operation and wide coverage, the bulk of this data is produced by satellites, with the Copernicus system alone delivering a reported 16 terabytes of data per day [156]. 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 [201]. Additionally, newer satellite constellations are equipped with more powerful sensors, capable of capturing higher spatial and spectral resolution imagery, leading to a further increase in data volume.

Earth System Models (ESMs) simulate processes of the Earth system to predict future climate. While they generate their own data, they rely on earth observation data to constrain their parameters. These systems also produce large volumes of data, and, driven by the need for higher resolutions to resolve and predict increasingly complex phenomena, these volumes are certain to increase with next-generation ESMs.

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. However, the computational cost of running them prohibits repeated computation, requiring the complete outputs of their runs to be stored. EO, on the other hand, is crucial not only in assessing the

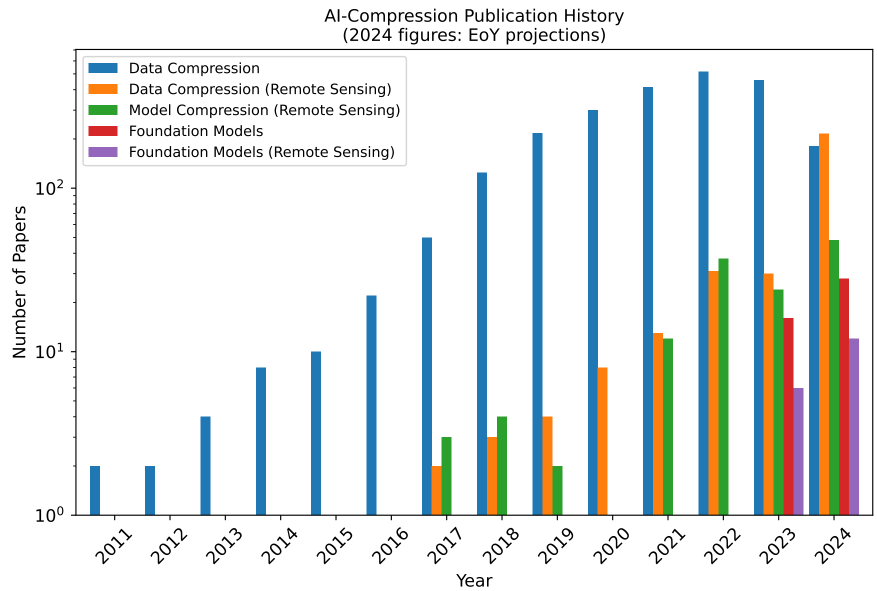

Figure 2: Literature on neural compression summarized for the past 15 years. We plot separate bars for remote sensing and recent developments in foundation model methodology. The data for 2024 are linear projections from March to the end of the year. Data source: queries to the *Web of Science* [199].

current state of the Earth but also in keeping a record of the past state, with datasets such as ERA5 [110] spanning decades. To best utilize this 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. The first relates to the limited bandwidth between satellites and base stations, required for transferring the observations for storage and analysis on the ground. This is a well-known problem in the community, referred to as the data downlink bottleneck [148]. The second relates to how such a volume of data can be stored on a physical medium, or transferred through a network, typically from data centers to different research institutions distributed globally.

This growth in volume demands new approaches to efficiently store only those aspects of the data that are required for their reconstruction or usage for geospatial analytics. Under this setting, in this review, we motivate the need for further research into *Neural compression* for EO and ESMs. Neural compression uses deep neural networks to perform data compression. It seeks to learn from datasets how to identify and efficiently store those critical aspects of the data, discarding unimportant or repeated information. It has seen success across several fields such as image compression [89], video compression [162], and audio compression [149], outperforming traditional hand-designed compression algorithms. These algorithms are mostly based on autoencoders [14] and do not require labels. However, they do rely on large datasets which representatively sample the underlying probability distribution of the data. This creates an inviting environment for adapting and applying neural compression techniques to EO and ESM. In this domain application, *Neural Compression for Geospatial Analytics* embraces computational algorithms employing artificial neural networks to reduce the storage required to digitize geospatial data while comprehending its information content.

Fig. 2 reveals the growth in popularity of neural compression in research publications. It also demonstrates that applying neural compression to this domain is not a new idea, as we see an increase in those publications, although with some delay. However, Fig. 1 illustrates that it is still a relatively unexplored topic in remote sensing, making up only 3 percent of publications in neural compression methodologies.

Foundation Models (FMs), large pre-trained neural networks that seek to provide embedding spaces that can be leveraged for multiple downstream tasks in a domain, have on the other hand been more quickly adopted in remote sensing domains. They share similarities with neural compression, with both being trained on very large unlabeled datasets to extract fundamental features from data. We frequently refer to FMs throughout this work, diving into their application to the geospatial domain in Section 1.3 and discussing their combination with neural compression in order to be used as compressed feature generators in Section 2 and Section 3.

The goal of this work is to introduce the topic of neural compression to a technical reader who is a newcomer to the field and explore its application to geospatial analytics. Section 2 will provide the relevant background as well as a review of state-of-the-art methodologies in neural compression. Section 3 continues by exploring neural compression in EO whereas Section 4 is dedicated to the compression of the outputs from ESMs, with both sections laying out the challenges, imposed by each domain, in applying neural compression. Section 5 focuses on how neural compressors for geospatial analytics may be integrated into geospatial data platforms, and, by way of example, discusses 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 6 highlighting relevant future directions for the field of neural compression for geospatial analytics.

1.2 Traditional vs. Neural Compression

Compression algorithms aim to encode a signal in as few bits–or symbols–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–or *codecs*–consist of a pipeline of components that have been engineered by hand 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 applications of 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 [13], H.264 [40], HEVC [47] or JPEG [11], to name only a few.

Learning-based methods, including artificial neural networks, have been explored for compression since at least the 1980s [10]. However, with the recent emergence of deep learning, promising results [166, 63] led to a growth of research in the field of *neural data compression*. The main premise of neural compression 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 just individual components are replaced, but rather the whole pipeline, leading to a codec that is learned fully end-to-end.

Two main benefits emerge from learned approaches. The first is the reduction in expert hours required for elaborating algorithms, relying on data-driven processes to determine the transformations applied to the data. This is particularly relevant for emergent data types, or those without extremely widespread use, such as scientific data, where it may not be economically viable to develop custom new codecs. By modifying the loss function, the codec can be explicitly trained to 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 these pipelines, compression algorithms can essentially achieve their goal in two main ways, both of which deep learning proposes to improve:

1. **Cleverly encoding the signal using fewer symbols.** Let us consider the simple scenario of a system that can be in one of three states A, B, or C with probabilities 0.9, 0.09, and 0.01 respectively. If we wish to transmit information about the state of the system across a network using the symbols 0, 10, and 110 (these are examples of prefix-free codes, meaning a string of such symbols can be decoded unambiguously without the need for special delimiting symbols), we can immediately see it makes sense to assign the shortest code, 0, to the most frequently occurring state A.

   We could try to further improve our encoder if we could leverage the symbols we have thus far decoded as context. For instance, let us assume we know that after the sequence A B, C always occurs. Then, we can potentially omit it completely from our encoded message, since the decoder will know after decoding A and then B, C must follow.

   However, no matter how clever our encoding scheme is, there is a fundamental limit to the minimum number of symbols we must transmit to recover a given signal, dependent on its information content. In Section 2, we formalize this notion of *information*, but it is ultimately dependent on the probability distribution of all messages we are interested in encoding. Having an accurate model of this distribution, in particular one that takes into account the context surrounding a given symbol, turns out to be 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 optimal encoding schemes.

2. **Allowing for some loss of information.** This differentiates codecs into *lossy* and *lossless* compression algorithms. 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.

   A common application of this idea makes use of spectral analysis. Let us consider compressing an audio signal. To do this, we can perform a Fourier transform, obtaining the spectral composition of the signal. If we were to transmit the result of the Fourier Transform, the receiver would be able to perfectly reconstruct the original audio signal by performing the Inverse Fourier Transform. However, we may decide that, for our human listeners, we can get away with dropping all frequencies above a certain threshold in our signal, thus reducing the amount of information we need to transmit. While that part of the signal is certainly now lost to the receiver, they may not notice or indeed may not mind.

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, more optimal trade-offs between message length and reconstruction quality can be discovered.

1.3 Foundation Models for Remote Sensing & Atmospheric Modelling

The emergence of foundation models represents a fundamental paradigm shift in deep learning. This paradigm shift has primarily resulted from three factors: (1) the availability of vast amounts of *unlabeled data*, (2) the emergence of self-supervision, a concept that allows deep learning models to learn from unlabeled data, and (3) a significant increase of computational power that allowed to train self-supervised models on vast amounts of unlabeled data at scale [125]. 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 foundation models.

While the adoption of foundation models began with natural language and images, several additional domains are characterized by significant available data volumes that have remained largely unlabeled due to a lack of scaling in the human annotation processes. Two of these domains are remote sensing of the earth's surface (e.g., satellite imagery) and atmospheric modeling. In both, data generation has accelerated significantly with the availability of sensors, satellite missions, and ground measurements. Human annotation of this data to enable task-specific supervised deep learning is therefore infeasible. Thus, self-supervision is of significant interest in remote sensing of the earth's surface and atmosphere, resulting in a recent push towards foundation models for remote sensing [181, 184, 165, 210, 168, 209, 179, 198]. Such task-agnostic models are meant to serve as a foundation that can be fine-tuned to a range of downstream applications with reduced computational effort compared to supervised deep learning, an improved generalization behavior, and a natural compression of the raw data into a latent space representation.

Empirical evidence demonstrates several improved capabilities of foundation models for remote sensing and atmospheric modeling compared to supervised deep learning models. For example, recent work has shown a significant acceleration in solving tasks in remote sensing based on pre-trained large-scale self-supervised models (e.g., [165]). In addition, the amount of required task-specific data (typically human-annotated) to finetune pre-trained foundation models can be significantly reduced [181]. This means foundation models are more data-efficient when applied to specific tasks than other supervised deep learning models. Furthermore, recent research demonstrated that foundation models for remote sensing 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 pretraining data compared to other supervised deep learning approaches [186]. Finally, based on the task-agnostic nature of the pretraining, the same foundation model can be re-used and finetuned to diverse tasks in earth observation and atmospheric modeling (e.g., [184]).

Despite various benefits resulting from foundation models for remote sensing of land and atmosphere, several challenges remain—especially regarding their significant data consumption, creating significant data transmission and storage bottlenecks. While foundation models 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 neural compression of embeddings particularly interesting [206, 208], especially in an upcoming era of large growth in data generation.

2 (Lossy) Neural Compression

In this section, we begin with an introduction to lossy compression, presenting neural compression as an extension of transform coding. We then provide an overview of the neural compression literature, proposing a taxonomy for navigating the field and detailing 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. [195].

2.1 Background

2.1.1 Lossy Compression

Lossy compression considers the scenario where an imperfect signal reconstruction is acceptable. More concretely, let S be a random variable, known as a source, producing symbols which we concatenate into strings $\mathbf{x} := (x_1, x_2, \ldots, x_n)$ from some alphabet $\mathcal{X}$. We often refer to $\mathbf{x}$ as the signal, or message, to be compressed. For example, in image compression, the message $\mathbf{x}$ would be a 3-dimensional tensor representing an image, with each symbol being an integer corresponding to a pixel value.

In order to do this, we require an encoder e that maps $\mathbf{x}$ to a string of symbols $\hat{\mathbf{z}} := (z_1, z_2, \ldots, z_m)$ from a different alphabet $\mathcal{Z}$. Additionally, a decoder d seeks to recover $\mathbf{x}$ 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 such schemes are *Huffman Coding* [3] or *Arithmetic Coding* [7]. Intuitively, ϕ may assign shorter binary codes to commonly occurring symbols or groups of symbols, to reduce the length of the encoded message without losing information.

Collectively, a concrete e, d and ϕ specify a *codec*. The goal of e is to minimize the length of the new string $\phi(\mathbf{z})$, known as the *code-length*, and expressed in bits, while minimizing the loss of information in the reconstruction from the decoder $\mathbf{x}'$. Abstracted in e is the mechanism by which information loss is introduced in the system as part of a quantization step. We discuss different quantization methods in further detail in Section 2.3.2. Therefore, lossy compression inherently involves a trade-off between compression and distortion.

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

$$\min \lambda D + R \ . \tag{1}$$

The two terms in this expression are the rate R, the expected number of bits required to transmit a data point, and the distortion D, the expected error between a data point and its reconstruction, with λ controlling the trade-off between both. Codecs with different trade-offs between R and D can be seen as optimizing for different values of λ. Plotting R against D as λ is varied yields a *Rate–Distortion plot*, which characterizes the performance of a lossy compression method and enables it to be compared to others. An example is shown in Figure 3.

We will now further concretize both of these terms, constructing a loss function from Expression 1. For the rate term R, we rely on Shannon's source coding theorem [2], which provides us with a lower bound on the number of bits required to losslessly encode $\hat{\mathbf{z}}$ as

$$-\log_2 p(\hat{\mathbf{z}}) \ , \tag{2}$$

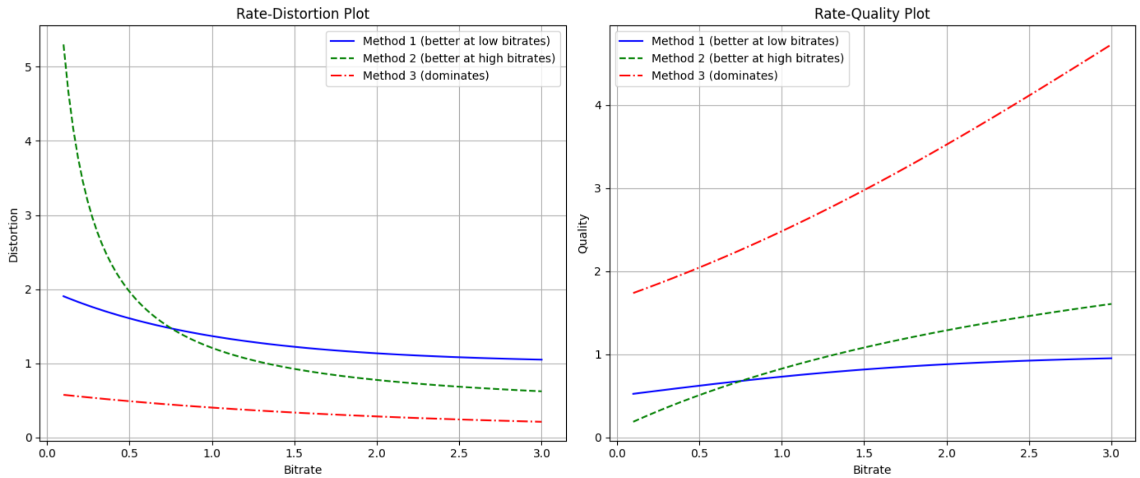

Figure 3: **Left:** Toy Rate–Distortion plot. The method in blue performs better than the one in green for low bitrates whereas the method in red outperforms them both across all bitrates. **Right:** Distortion is often replaced with some quality measure (e.g. PSNR) in which case we obtain a Rate-Quality plot.

where p is the probability mass function of the distribution of $\hat{\mathbf{z}}$. The expectation of this quantity over p

$$\mathbb{E}[-\log_2 p(\hat{\mathbf{z}})] \ , \tag{3}$$

is known as the entropy of the distribution and characterizes how difficult it is to compress samples drawn from it.

Taking the value from Expression 2 as R abstracts us from any particular encoding scheme that may be used. While this value is only the best achievable rate, and not the real number of bits achieved by the algorithm (known as the *operational rate*), in practice, encoding schemes such as arithmetic coding [7] can achieve operational rates very close to this lower bound [216], making this a good approximation which will also be useful to enable calculating gradients with respect to this loss. However, this estimate is very dependent on the form we determine for p.

The distortion term D relies on an underlying error function $\rho(\mathbf{x}, \mathbf{x}')$. ρ can be any error measure that appropriately captures distortion between the input and reconstruction for the domain in question. Typical examples for image and video compression are the mean squared error or the structural similarity index measure (SSIM) [24].

2.1.2 Transform coding

The neural compression methods explored in this section can be seen as non-linear, learned variations [106] of the transform coding paradigm [19]. Transform coding is a core idea behind modern codecs, including traditional ones such as JPEG or HEVC. It extends the previous encoder by adding a first step: applying a transform on the raw data before quantizing and entropy coding it. This transform aims to map the data to a space where it can be more easily compressed.

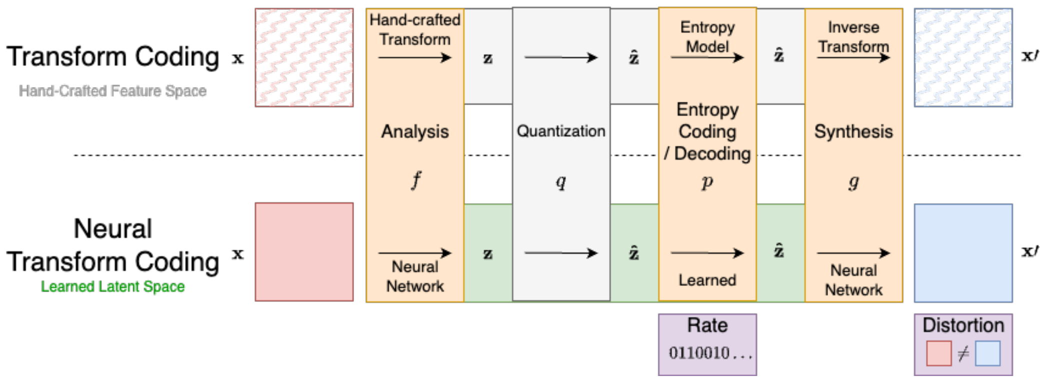

Figure 4: Illustration of the transform coding framework and its adaptation to neural transform coding. Components within orange boxes are replaced with learned counterparts. Terms within purple boxes are used in the loss.

Neural transform coding is a natural extension of transform coding, where hand-crafted components in the pipeline can be replaced by artificial neural networks trained on large datasets. We now go through these steps, illustrated in Figure 4, identifying these extensions:

- **Transform $f(\mathbf{x}) = \mathbf{z}$.** Often, the input data $\mathbf{x}$ is expressed as a vector whose coordinates are correlated. In natural images, for instance, adjacent pixels tend to have similar values. These correlations are redundancies in the input signal: knowing part of the signal allows guessing the remainder without actually seeing it. Hence, discarding such correlations is desirable in a compression framework. In practice, a *transform* f is used to map the data into another representation $\mathbf{z} = f(\mathbf{x})$ whose coordinates are less correlated, and ideally independent. This operation aims at facilitating the quantization and entropy modeling steps that follow. The dimensions of $\mathbf{z}$ will often take continuous values, as is the case with the Fourier Transform [8] or the Discrete Wavelet Transform [12] (used in JPEG).

  In traditional compression literature, f is a hand-crafted, linear, fully-invertible mapping and is referred to as the *analysis transform*. In neural compression, f is an artificial neural network trained to map $\mathbf{x}$ to an *embedding* $\mathbf{z}$ in a continuous latent space. Somewhat misleadingly, the neural network f may also be called the *encoder network*. The ability to learn complex non-linear transforms directly from dataset statistics puts neural compression approaches at a great advantage compared to handcrafted methods. Figure 5 illustrates this with a toy example.

- **Quantization $q(\mathbf{z}) = \hat{\mathbf{z}}$.** Since the output of the transform is embedded in a continuous latent space, it must necessarily be quantized, as data from a continuous source requires an infinite number of bits to be losslessly compressed. Beyond its necessity in this case, this discretization is where the information loss is introduced in the compression process, and is thus also the mechanism by which rate is traded for distortion, controlled by the transform and the quantization method. A neural network transform f can learn how to warp the embedding space to effectively manipulate which information is lost through quantization. By quantization, we broadly mean any mapping from a continuous to a discrete and countable set. The chosen quantization method q is applied to $\mathbf{z}$, with $q(\mathbf{z}) = \hat{\mathbf{z}}$.

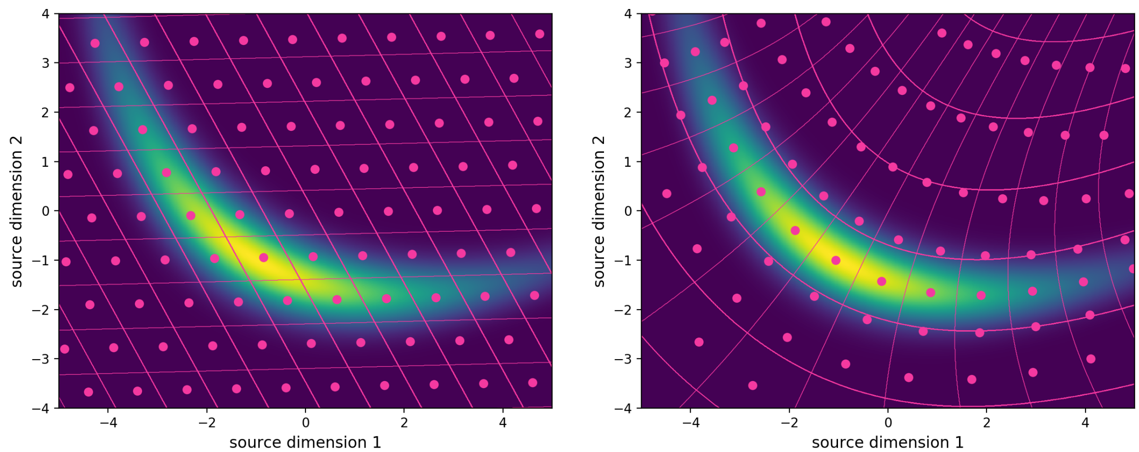

Figure 5: Linear transform code (left), and nonlinear transform code (right) of a banana-shaped source distribution, both obtained by empirically minimizing the rate–distortion Lagrangian. Lines represent quantization bin boundaries, while dots indicate code vectors. While LTC is limited to lattice quantization, NTC can more closely adapt to the source, leading to better compression performance [...]. Figure and caption taken from [106]

- **Entropy (de)coding.** Once the discretized representation of the data $\hat{\mathbf{z}}$ is obtained, it can be losslessly compressed through entropy coding, mapping symbols into codes whose length reflects their likelihood given the previous symbols. Assuming an efficient encoding scheme, such as arithmetic coding, the efficacy of this compression is determined by the *entropy model* p'. Since the underlying data distribution p is unknown, we can only model it. An accurate approximation p' is critical for the entropy code to correctly assign shorter codes to more frequently occurring symbols and minimize the average length of the compressed representations. How well p' approaches p determines how close to the lower bound in Expression 2 the length of the final encoding can get, with an exact match when p' exactly models p. In neural compression, the entropy model also takes on an additional role during training, where it is used to provide differentiable estimates of the cost in bits of encoding a batch. This quantity is incorporated into the loss of the model, enabling an end-to-end rate–distortion optimization. On the receiver side, the reverse process is applied to recover $\hat{\mathbf{z}}$.
- **Inverse Transform $g(\hat{\mathbf{z}}) = \mathbf{x}'$.** To reconstruct the data, the inverse transform g is applied. In the neural compression case, we usually to not have access to an analytical expression of the inverse of the encoder f. Instead, a second neural network is employed to learn to reconstruct the original data given the quantized representation. g is often referred to as the *synthesis transform* or, in the case of a neural network, the *decoder*. Finally, this gives us $g(\hat{\mathbf{z}}) = \mathbf{x}'$, the reconstruction of $\mathbf{x}$ with some loss.

Neural compression aims to optimize over the Lagrangian from Expression 1 end-to-end using the machinery of deep learning. By setting the loss function to the Lagrangian, defining f and g as deep neural networks, and flexibly modeling p (often also through neural networks), neural compression can learn non-linear functions for the transforms and complex models for p, with superior rate–distortion performance compared to traditional compressors, which has been demonstrated for tasks such as image [89], video [162], audio [149], and 3D scene compression [124].

Two important details, which we postpone to Section 2.3.3, are the concrete form of the quantization and how a continuous model p can be used to fit the resulting discrete distribution.

We can now write out the full form of the loss for a single input $\mathbf{x}$ as

$$L(\mathbf{x}) = \lambda \cdot \underbrace{\rho\Big(\mathbf{x},\ g\big(q(f(\mathbf{x}))\big)\Big)}_{D} \overbrace{-\ \log_2 p'\Big(q\big(f(\mathbf{x})\big)\Big)}^{R}\ . \tag{4}$$

Let us analyze each term individually. For the distortion term D, we see that only f and g, the encoder and decoder networks, take part in it. The gradients of this loss will push the encoder network to produce representations that are robust to quantization such that they are capable of being reconstructed by the decoder into the original input as accurately as possible.

For the rate term R, we see that only the entropy model and the encoder network take part in it. From the perspective of the encoder network, it may minimize this term by minimizing the entropy of the distribution of symbols it produces, guided by the entropy model p'. In other words, the encoder network is encouraged to produce compressible representations $\mathbf{z}$.

From the perspective of the entropy model p, it is helpful to realize that this entropy term takes the same form as the cross-entropy loss one would use to fit a model p to a distribution through samples $\hat{\mathbf{z}}$. Furthermore, the cross-entropy is precisely the expected amount of bits required to entropy code $\hat{\mathbf{z}}$ when using the approximated distribution p' and is minimized when p' matches p. Thus, to minimize this term, p will aim to assign high probability to $\hat{\mathbf{z}}$ to minimize the loss, under the constraint that it must be a valid probability density function.

Since p only participates in the loss through this 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 $\hat{\mathbf{z}}$.

2.2 Compression Taxonomy

We begin by broadly categorizing approaches to compression. The first categorization is between lossy and lossless compression, where the characterizing decision is whether some loss of information is acceptable, usually through the introduction of quantization. We continue by further dividing methods into those that are explicitly engineered and those that are learned from data. The compression algorithms in most widespread use today fall into the category of explicitly engineered, as is the case for JPEG [11], MP3 [13], or HEVC [47].

Due to the breadth of the field, we limit the scope of this section to those methods we believe are set to have a greater impact on research in the near future. First, while there is some work on lossless neural compression [98, 116, 104], we consider only lossy neural compression methods. This is necessary to provide compression ratios that can compete with currently employed standards, such as JPEG, and thus offer viable alternatives when deployed across EO applications at scale.

Second, we focus on learned compression methods, which have consistently been shown to outperform their hand-designed counterparts in rate–distortion metrics.

With the framework exposed in this chapter in mind, we identify the four main axes of variation in such works:

- **Transforms.** This axis encompasses all aspects of the encoder and decoder models, chiefly determined by their architecture.
- **Quantization strategies.** This axis defines how continuous data representations are discretized, and how this discrete step can be made compatible with an end-to-end learning process.
- **Entropy models.** This axis is determined by the chosen set of assumptions in modeling the distribution of the transformed data (namely the independence or conditional independence relations among different dimensions) together with the implementation used to concretize these assumptions.
- **Optimization objectives.** Neural compression approaches are trained end-to-end using a rate–distortion loss. The final axis represents the variability in optimization objectives, namely concerning the chosen distortion measure.

2.3 Methods in Neural Compression

Table 1: Collection of neural compression papers in this chapter, aligned by contributions along the axes described in Section 2.2.

Axis	Approach	Papers
Transforms	CNN	[63, 82, 166, 89]
	RNN	[59]
	Transformer	[170]
	INR	[124, 130, 127, 164, 178]
Quantization Strategies	Scalar Uniform Quantization	[62, 166]
	Binarization	[59, 78, 86]
	Vector Quantization	[69, 88]
	Weight Quantization	[124, 178]
Entropy Models	Fully Factorized	[62, 82]
	Hyperprior	[82]
	Autoregressive	[82]
Optimization Objectives	Rate-Distortion-X	[150, 172]
	Downstream Embedding	[118, 129]
	Split Computing	[161, 205]

2.3.1 Transforms

In the learned image and video compression domains, the synthesis and analysis transforms are most often concretized as two halves of a deep convolutional auto-encoder [43] as popularized by Ballé et al. [63] and Theis et al. [166].

In these works, the general structure of the encoder follows that of a typical deep convolutional architecture, with a repeating pattern of convolutional layers and non-linear activations gradually reducing the spatial dimensions of the feature maps while increasing the number of channels (otherwise known as the embedding dimension). As its counterpart, the decoder also employs a deep convolutional architecture to gradually spatially upsample the feature maps while reducing the number of channels to recover the original input. This architecture, as per Ballé et al. [82], is shown in Fig. 6. Many improvements to these transforms have been inherited from architectural innovations in the fields of deep learning and computer vision, for instance, the integration of mechanisms such as attention [51] or residual connections [64] into the network.

Enabled by the general formulation of neural transform coding, as neural compression has continued to evolve, researchers have explored alternative architectures beyond traditional convolutional networks. Indeed, unstructured data compression has leveraged fully connected feed-forward neural networks [106] and earlier works employed recurrent neural networks (RNNs) [59] as encoder and decoder architectures. More recent works have also explored the use of transformers [171].

A very different paradigm that has also emerged is that of *Implicit Neural Representations* (INRs). Popularized in large part through their use in Nerf [141] for 3D scene representation, INRs have shown promise as an alternative way of representing and storing 3D geometry [100, 99, 94], audio [119], images [119], video [128], amongst others. INRs aim to represent any signal as an implicitly defined function. For instance, we may represent an image as a function $f(x, y) : \mathbb{R}^2 \to \mathbb{R}^3$ 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 network's weights. This representation allows for the leveraging of model compression literature to achieve general signal compression. It has successfully been employed for compressing 3D scenes [124], images [130, 164] and videos [127, 178], although their use together with an entropy penalty is still not ubiquitous, with many methods relying instead on more typical model compression techniques.

Somewhat informally, we propose that this can also be seen 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 network itself. The remaining 3 axes are then still fully applicable to INRs. INRs show competitive performance in 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 neural compression 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 compression [127, 178]. However, the lengthy training process required for compressing each sample makes INRs impractical for deployment in many real-world applications.

2.3.2 Quantization Strategies

It can be shown that the optimal rate for a given distortion can be achieved through vector quantization [29]. 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.

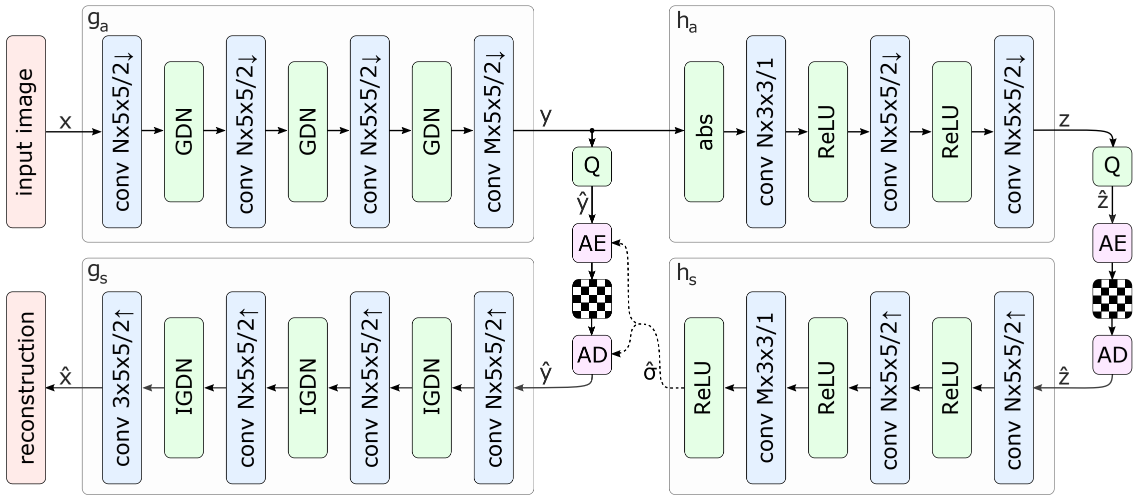

Figure 6: Network architecture of the hyperprior model [82]. The left side shows an image autoencoder architecture, the right side corresponds to the autoencoder implementing the hyperprior. The factorized-prior model uses the identical architecture for the analysis and synthesis transforms g_a and g_s. 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. [82].

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. [62] and Theis et al. [166], where each dimension of the transformed data is independently quantized. In practical terms, this quantization usually consists of rounding each element of the transformed input 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 [195]. Its effectiveness despite its simplicity is enabled by the flexible non-linear transforms which can in essence warp this grid as desired. The two approaches are illustrated in Figure 7 The main obstacle introduced by the quantization step is that of non-differentiability. To carry out the end-to-end optimization, it is necessary to backpropagate gradients through the whole network. However, the quantization step will have a gradient of 0 almost everywhere, preventing any components before it from receiving gradients necessary for optimization. The two main techniques to address non-differentiability are:

- **Straight Through Estimator (STE)**. The STE [48] proposes to backpropagate through non-differentiable components by treating them as if they were the identity function during the backpropagation process, fixing their gradient to 1, thus allowing gradients to pass through unchanged. Theis et al. [166] propose to apply this to the quantization function for end-to-end training. Through the forward process, the quantization is unchanged.

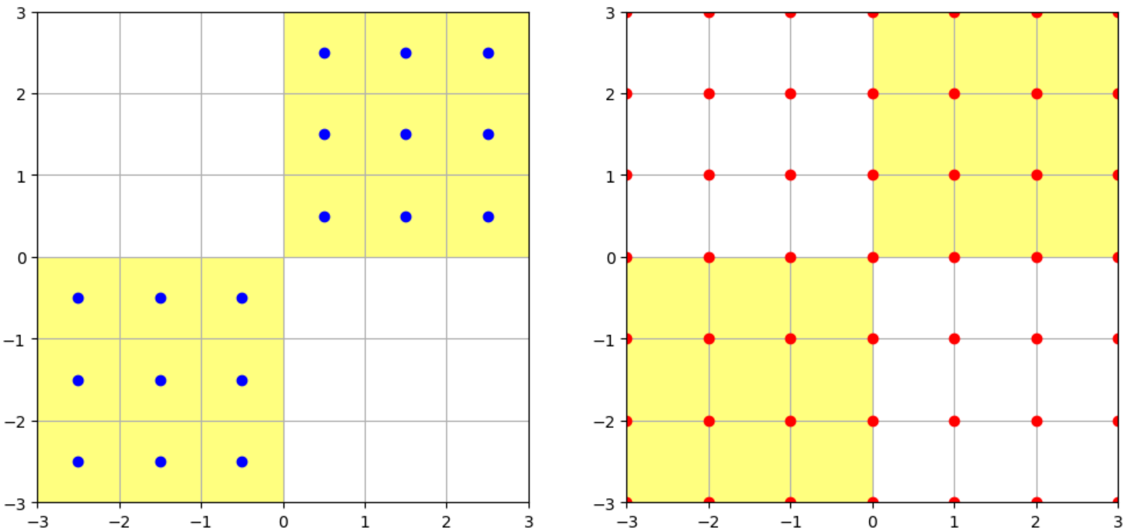

Figure 7: Illustration of vector vs scalar quantization for a given data distribution, shown in yellow. **Left:** Vector quantization can efficiently cover the space by freely building a codebook of vectors, shown in blue. **Right:** Scalar quantization quantizes each dimension individually. This potentially 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.

- **Uniform Noise**. Ballé et al. [62] propose the replacement of quantization during training with additive uniform noise with the same width as the quantization bins. In the case of rounding to the nearest integer, this has the range of $[-0.5, 0.5]$.

Empirically, the combination of both of these methods seems to be optimal for training [147], with the STE used for calculating the distortion term and additive uniform noise used for the entropy term.

Other forms of quantization have also been explored, despite being less popular. Early works in the field employed binarization as a form of quantization, reducing every element of $\mathbf{z}$ to one of two values [59, 78, 86]. Vector quantization (VQ) has also been successfully employed in neural transform coding [69, 88] 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) [131] for vector quantization and binarization to make this quantization more computationally feasible [196]. 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 [84] 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 feeding the learnt (quantized) embeddings and their codebook to 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.

In the case of INRs, quantization often inherits from general neural network compression, involving strategies such as weight quantization or pruning [130, 127]. These can be applied after the optimization process or, often achieving better results, throughout it. Quantization-aware training [85] can be used to obtain an INR that is more robust to the error introduced by quantization, and finetuning after pruning can reduce its effect on distortion [127].

2.3.3 Entropy Models

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

a) the estimation of the rate during training to be used in the loss and

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 in this process, while *a)* additionally requiring differentiability to enable end-to-end training. To optimize a differentiable model over the discrete $\hat{\mathbf{z}}$, the most common approach is to employ uniform quantization and define p in terms of an underlying continuous density p' [195].

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

For each entropy modeling method, the assumptions imposed on $p(\hat{\mathbf{z}})$ *a priori* to make the above integral tractable and thus enable both *a)* and *b)* are the main factors determining the possible architectures for the entropy model and have a great impact on the final performance.

Fully factorized model: one of the stronger simplifying assumptions we can make is that each element of $p(\hat{\mathbf{z}})$ is independent, enabling a fully factorized model. The model for each independent marginal can be implemented with varying degrees of complexity, from a simple parametric distribution such as a Gaussian or Laplacian to a neural network modeling the c.d.f. of the distribution [82].

Hyperprior model: Despite its simplicity, the assumption of full independence is most likely too strong. A typical way to model dependence between variables is to instead consider them conditionally independent given some other latent variable [15]. Ballé et al. [82] use this technique to extend their fully-factorized model to a latent variable model, known as the *hyperprior* approach.

$$\mathbf{z} \sim p(\mathbf{z}|\hat{\mathbf{h}}) \tag{5}$$

$$\mathbf{h} \sim p(\mathbf{h}) \tag{6}$$

Following this approach, the hyperprior $p(\mathbf{h})$ can be modeled in the same fully factorized way as $p(\mathbf{z})$ before. $\mathbf{z}$ is now modelled given the quantized hyperprior $p(\mathbf{z}|\hat{\mathbf{h}})$ using a 0-mean gaussian whose standard deviation is generated from $\hat{\mathbf{h}}$ for each dimension of $\mathbf{z}$. While the new latent variable becomes "side-information" which must also be compressed and transmitted, its size is negligible compared to $\hat{\mathbf{z}}$, and the added flexibility in the model tends to result in significantly improved R-D performance. Intuitively, this additional information about the particular $\hat{\mathbf{z}}$ that is being transmitted allows the receiver to adapt its entropy model.

Autoregressive and transformer-based models: more complex models can also be used, enabling more sophisticated dependencies between each element of $\hat{\mathbf{z}}$ and the context that preceded

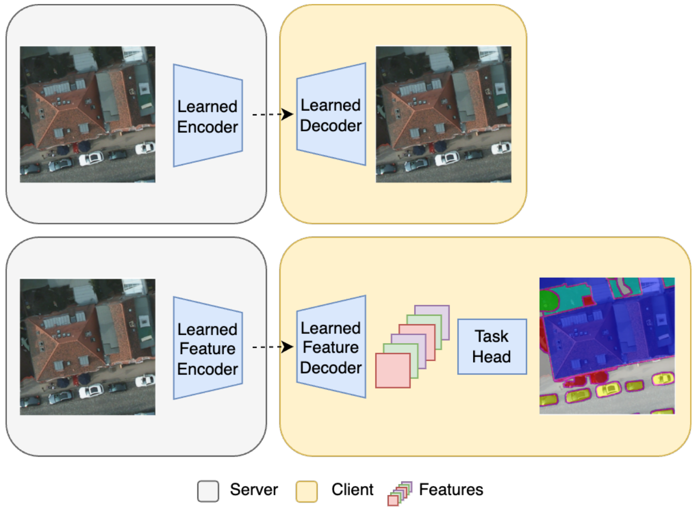

Figure 8: Illustration of feature compression. **Top:** the usual scenario where a client wishes to reconstruct the input compressed by the server. **Bottom:** the server sends a compressed feature representation of the input, which the client can directly feed to a task-specific head (e.g. semantic segmentation in this case)

it at the cost of computational complexity, for instance in the case of autoregressive models [117]. Transformers have also been successfully employed for modeling complex relationships in the latent space of $\hat{\mathbf{z}}$, particularly in video compression. Mentzer et al. [162] can greatly simplify video compression pipelines by relying on the modeling power of a transformer model. More recently, the growth in research in large language models (LLMs) has also led to their exploration for compression, highlighting the known connection between generative modeling and compression. LLMs have been explored for lossless [174] as well as lossy [196] text, image, and video compression, showing promising results in leveraging their predictive power and large context modeling for next-generation compression algorithms.

INRs: Entropy-based end-to-end optimization has also been applied to good effect in INRs. In this case, the weights of the network themselves are treated as $\mathbf{z}$, and their distribution, after quantization, is modeled. This has shown competitive results for 3D scene [124] and video compression [178].

2.3.4 Optimization Objectives

Traditionally, most works are concerned with distortion as measured by human perception, exploring a variety of loss functions that attempt to best approximate this idea, most commonly MSE, SSIM, or a combination of the two. Identifying loss functions that act as accurate proxies to human perception is an active field of research not only within neural compression but also for generative visual models in general.

Recent works have explored new trade-offs that can be navigated by reinterpreting what is meant by distortion. Examples include the introduction of rate–distortion–perception [150] and rate–distortion–realism [172] trade-offs.

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. If the end consumer of the data is an algorithm (e.g. a neural network), rather than a human, is compression designed for human perception the optimal choice? From this perspective, recent works propose reframing distortion from an algorithmic point of view. As illustrated in Fig. 8, 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 [118]. 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 self-supervised learning [208] to identify which aspects of the data may safely be lost during compression without affecting downstream performance [129]. A similar idea has been applied to transfer data in the reverse direction, from an edge device collecting data to a powerful server where that data can be analyzed, in a paradigm known as *Split Computing*. Matsubara et al. [161] combine *Knowledge Distillation* with neural compression 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. [205] 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 architectures, showing showing improved R-D performance.

Indeed, while foundation models for vision have been shown to generate embeddings that can be leveraged for several tasks [143, 126, 192], the dimensions of their output feature space can 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 foundation models increasingly become prevalent in EO [181, 192, 155, 191].

3 Neural Compression for Remote Sensing

Remote sensing technologies are evolving and the number of earth observation satellites in space is rising, with more launched in 2023 than any year before [201]. This enables the acquisition of satellite images with increasing spatial resolution, broader spectral bands, higher temporal frequency, and higher radiometric resolution [73, 212]. The increase in data volumes has made the transmission, storage, and processing of satellite imagery on a large scale an increasingly challenging task. The study of image compression commonly focuses on natural images, which presents challenges for the application of established compression approaches in the context of different domains. This necessitates the development of data compression methods that are tailored to the specific requirements of remote sensing data. The following chapter examines these domain-specific requirements and challenges in Section 3.1. Subsequently, an overview of compression research for remote sensing images is presented in Section 3.2 and a discussion of future developments in Section 3.3.

3.1 Challenges

To outline the domain-specific challenges and opportunities for compression, we analyze the distinctive data characteristics of remote sensing data in comparison to natural images. Following this, constraints imposed by the data acquisition and usage are discussed.

3.1.1 Data Characteristics

The resolution at which remote sensing images are recorded are defining characteristics that distinguish them from natural images and differentiate various types of remote sensing data. This distinction is for example evident in comparison to RGB images, and has implications for the applicability of existing architectures as the computational complexity and training behavior of neural approaches can vary significantly [113]. Conversely, it also presents opportunities for compression due to correlations along these dimensions.

- **Spectral resolution**. Remote sensing images typically comprise multiple spectral bands that extend beyond the visible range, including infrared and ultraviolet. This allows for a comprehensive analysis of surface and atmospheric conditions. Multispectral data encompasses a limited number of broad spectral bands, whereas hyperspectral data collects measurements from a multitude of spectral bands (up to hundreds) that span narrow wavebands, increasing the input data size and computational complexity of processing the data. Narrower wavebands due to multi- or hyperspectral data cause adjacent spectral bands to be more correlated.

- **Spatial resolution**. The spatial resolution is defined as the size of a pixel, which represents the distance between the measurement points of a sensor. Compared to other domains, remote sensing data generally has a lower spatial resolution, as it covers large geographical areas from a great distance. Consequently, individual pixels can contain highly relevant information for downstream tasks, and images exhibit complex textures with rich information [194, 170]. At the same time, earth observations capture specific landscapes, thereby exhibiting spatial redundancy. This includes both local texture information and global feature distribution [177, 175]. Image data is accompanied by metadata on geographic coordinates, elevation, and sensor angles.

- **Temporal resolution**. Most satellites capture data of a specific geographic region at regular intervals, resulting in a time series of images of a certain location. This temporal resolution allows to track dynamic processes and environmental changes. Successive images show temporal correlations due to slowly changing landscape structures but are also subject to seasonal and weather-dependent changes. Data used in general compression research differs from this as approaches are either built for image datasets, which share no temporal structure between images, or as part of video compression research, where data frames are much closer together than for a time series of satellite images.

- **Radiometric resolution**. Radiometric resolution refers to the ability of a sensor to measure the intensity of radiation reflected from an observed object within a specific wavelength range. For example, satellite data from the Sentinel series uses sensors with a 12-bit resolution, which is higher than the radiometric resolution of natural photos often having a 8-bit range. This high radiometric resolution provides more precise measurements but also leads to a larger input alphabet, which can increase the complexity for compression algorithms. This is due to the necessity of encoding a greater number of distinct values and the fact that larger alphabets can reduce the probability of redundancy in the data. As a result, it becomes more challenging to utilize patterns for efficient compression. Larger alphabet sizes are further linked to a larger input data size and higher entropy, both of which impact the achievable compressed data size.

Remote sensing data is recorded for various purposes and with instruments that vary in spectral, spatial, temporal, and radiometric resolution. This makes it challenging to define a typical remote sensing image, as different datasets can vary greatly in these dimensions. Additionally, there may be differences that are specific to the sensor or post-processing applied, which affects the brightness and contrast of images, for example. The degree and steps of post-processing can also be an important factor that distinguishes remote sensing data from natural imagery. As part of a comparison, we analyzed the image entropy of Sentinel-2 satellite and ImageNet data. For each image, the distribution of the measured pixel values is recorded and the image-specific entropy calculated, not taking the pixel positions into account. If we compare this at dataset level, it provides information on whether a dataset consists of images with low, fluctuating or high entropies. For this purpose, we compute the histogram of pixel values for each image individually, given by:

$$p_{\text{img}}(x_i) = \frac{\text{Number of pixels with intensity } x_i}{\text{Total number of pixels in image}} \tag{7}$$

The distributions obtained are used to calculate the entropy, resulting in an individual entropy value for each image.

$$H_{\text{img}}(X) = -\sum_{i=1}^{n} p_{\text{img}}(x_i) \log p_{\text{img}}(x_i) \tag{8}$$

The results in Figure 9 show that the majority of natural photos have a very high entropy. ImageNet is composed of a wide variety of scenes and motives, this analysis shows that the pixel values within the images are also very diverse. This may be due to a generally greater variety of colors and patterns, but can also be the result of more extensive post-processing of the photos, which leads to higher-contrast colors. Satellite images, on the other hand, have a wider spread and on average a lower entropy per image across the RGB bands. The low entropy of some images in the satellite dataset can be explained by images of certain landscape scenes. For example, an image of the

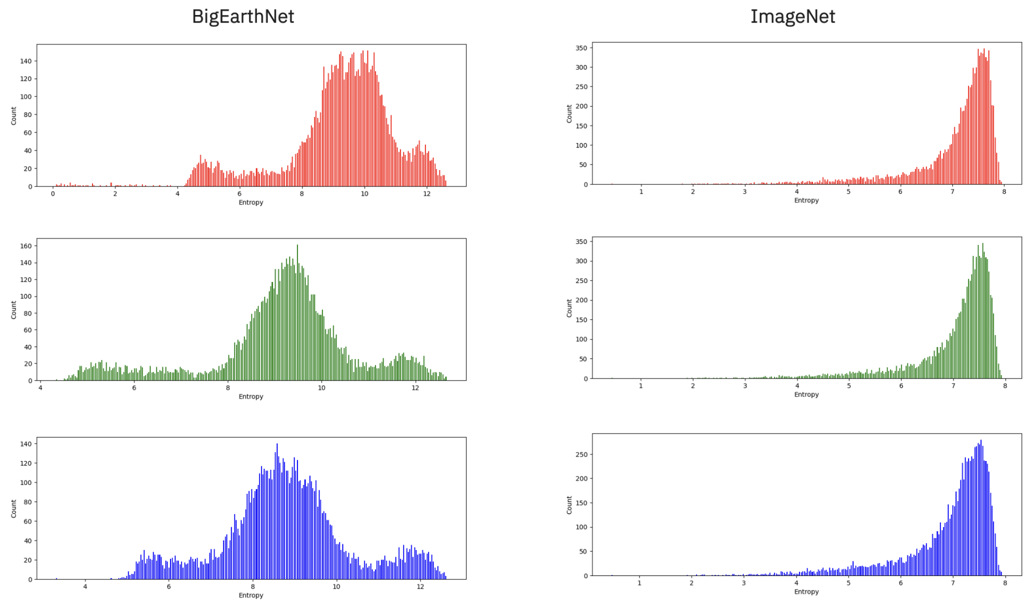

Figure 9: Histograms of per image entropy values for RGB bands. **Left**: 10000 randomly sampled Sentinel-2 images from BigEarthNet. **Right**: 10000 randomly sampled images from ImageNet.

sea has a very concentrated probability distribution of pixel values. Those data subsets of specific scenery potentially have lower entropy and therefore rather homogeneous input pixel values. This is a property that may be exploited by domain specific compression algorithms.

Compression techniques leverage bias in a data set, 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. A general compression algorithm is designed for a broad spectrum of data and can be used for a wide range of images, but often falls short in performance because specific properties are not present throughout the data. A dataset-specific algorithm, on the other hand, can be fine-tuned to the particular patterns within a dataset, such as recurring landscape features in satellite imagery, resulting in higher compression rates. The differences between natural photos and satellite data thus make it difficult to use models from the natural photo domain for remote sensing. Even within remote sensing, 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 remote sensing data (See Section 3.2).

3.1.2 Data acquisition and application

In addition to underlying differences in data dimensions and characteristics, requirements for compression of remote sensing data are caused by the manner of data collection, processing and use.

- **Acquisition**. Remote sensing is based on the collection of data by airborne or satellite-based sensors, which is then transmitted to a ground station on Earth. Especially with satellite data, the problem of transmitting the data to Earth, known as the data downlink bottleneck, is currently a limiting factor in the data collection pipeline [148]. To increase the amount of data that can be transmitted, satellite images are already compressed onboard to reduce their size. However, due to the limited resources on board, only compression methods with low computational and storage complexity can be used. For onboard satellite compression, the Consultative Committee for Space Data Systems (CCSDS) recommends the use of a discrete wavelet transform based on the ground-based JPEG2000 standard [27]. The computational requirements of CCSDS have been significantly reduced compared to JPEG2000, considering the limitations in the image processing units of satellites. Neural compression methods, which are successfully used for lossy image compression, can outperform conventional transform coding methods in terms of the trade-off between rate and distortion (as explored in Section 2). However, this comes at the cost of higher computational complexity and therefore limits their usability for on-board satellite applications.

- **Application**. Satellite data is crucial for a variety of applications like environmental monitoring of above ground biomass [114], agriculture mapping of oil palm density [144] or in disaster management for flood extent mapping [202]. All of these rely on the analysis of specific aspects and features in the data, which leads to diverse requirements for data processing and compression. For remote sensing data, compression applied to data before it is distributed is generally carried out at a high bit rate for data integrity purposes, with a tendency towards almost lossless compression. This should achieve a balance between compression and preserving information crucial for downstream applications. Nowadays many of these process input data with machine learning models. As a result retention of details for machine processing becomes more and more important. Recent research in the field of lossy image compression focuses primarily on optimizing for human perception, for example in terms of MSE or PSNR, which might not align with the distortion picked up by downstram procsessing algorithms.

3.2 Classification of Compression Methodologies

In this section, we discuss approaches to the compression of remote sensing data with traditional hand-crafted methods, the main algorithms used operationally for compression in the domain currently. We then discuss current work in adapting neural compression approaches to satellite imagery.

3.2.1 Traditional approaches

Most traditional methods such as JPEG and JPEG2000 are based on the framework of transform coding. As introduced in Section 2.1.2, these methods transform the original data into a different domain, usually the frequency domain, which enables more efficient coding. Transformations used are for example the Fourier transform [8], the Karhunen-Loeve transform (KLT) [1, 6], the discrete cosine transform (DCT) [5], and the discrete wavelet transform (DWT) [12]. These transformations decorrelate the dimensions of the data, making it easier to compress. Due to their efficiency and speed, transform-based methods are also widely used in remote sensing [22, 36, 23]. Among the earlier studies in optical satellite data compression, Hou et al. [17] propose a variant of the JPEG coding scheme that explicitly detects clouds to simplify the compression of cloud features. Gonzalez-Conejero et al. [38] compresses remote sensing data by adapting JPEG2000 to input data containing unimportant areas such as non-data regions. Based on JPEG, the Consultative Committee for Space

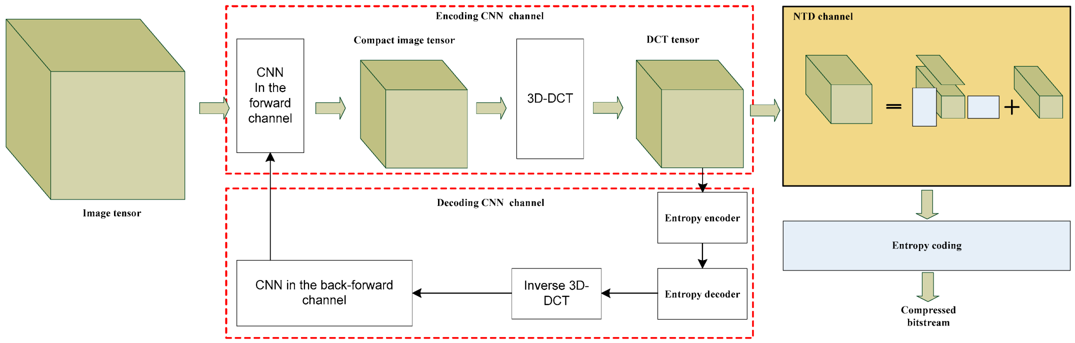

Figure 10: CNN-based transformation as part of a tensor decomposition framework introduced by Li et al. [97]. Figure taken from the original paper.

Data Systems (CCSDS) designed international remote sensing image compression standards which are commonly used for onboard compression [26, 115, 34].

Transform-based compression methods for natural images focus on decorrelating the input along the spatial dimensions. Remote sensing data, however, often contains an increased number of spectral bands. To exploit this, handcrafted compression approaches have been extended to explicitly decorrelate along the spectral dimension, leveraging the redundancies therein. Markman et al. [21] extend the DCT to a third dimension for hyperspectral data. Similarly, Lim et al. [20] and Luigi Dragotti et al. [18] apply a three-dimensional wavelet transform to hyperspectral images using the three-dimensional version of the SPIHT algorithm. Du et al. [30] and Du et al. [32] combine JPEG2000 with PCA for spectral decorrelation.

Other approaches that take into account the multispectral character of the data are *tensor decomposition* techniques, which deal with multidimensional matrices, or tensors, by decomposing them into a sum of simpler low-rank components. Important methods include the Tucker decomposition [4], which approximates a tensor by a set of factor matrices and weight coefficients in the form of a reduced core tensor. The column vectors of the factor matrices represent the set of basis functions onto which the data is projected, thus defining the mapping between original and compressed data. These methods are particularly effective for high-dimensional data such as hyperspectral images. In contrast to transform-based methods, tensor decomposition is more complex but can achieve high compression rates for multispectral data [61, 68, 71] while preserving their multidimensional structure. As a result, they are being explored for compressing multispectral remote sensing images [39, 37, 42, 45, 46, 215]. Consequent work aims to lower the complexity of tensor decomposition approaches for better applicability to the domain. Li et al. [75] applies Tucker decomposition to multispectral images with comparatively few bands in a post-transform domain and Li et al. [52] accelerates the application of nonnegative tensor decomposition for hyperspectral image compression by a pairwise multilevel grouping approach.

3.2.2 Neural approaches

We now introduce different approaches to neural compression in the domain, an overview can be seen in Table 2.

Table 2: Contributions to the field of neural compression for remote sensing described in Section 3.2 ordered along the axes of the taxonomy described in Section 2.2.

Axis	Approach	Papers
Transforms	Complexity Reduction	[122, 188]
	Novel spatial extraction	[135, 157, 170, 193, 177]
	Novel spectral extraction	[135, 113]
	Separate spectral/spatial extraction	[136, 154]
	Incorporate Wavelet Transform	[194, 213, 197]
	Bitrate Allocation	[214, 203]
	Image-specific (INRs)	[187, 215]
Entropy Models	Hyperprior with attention	[177]
	Multiple hyperpriors	[175]
	Split latent space	[154, 194, 213]
Optimization Objectives	Adversarial loss (GANs)	[151, 185]
	Downstream Embedding	[206, 208]

Neural transformation. Li et al. [97] is one of the earlier papers that uses a neural component as part of a traditional method and combines CNNs with tensor decomposition as seen in Figure 10. An encoder CNN plus a DCT is used to obtain a more compact representation of the input data so that the non-negative tensor decomposition (NTD) is less expensive. The CNNs are trained on the MSE of reconstructed images.

Autoencoder. An autoencoder represents the simplest form of a neural compression architecture. It compresses data by encoding it into a latent space with reduced dimensions, via a bottleneck layer in the neural network. The network is trained on reconstruction errors between the input and output. One of the earlier contributions applied to satellite imagery is Yang et al. [26], which employs a three-layer autoencoder architecture. The activation function within the bottleneck layer is modified to a ridgelet transform [16], which results in better performance than that achieved by classic neural networks with sigmoid activation. Kuester et al. [113] were among the first to use an autoencoder for hyperspectral satellite imagery. To cope with the high spectral resolution, the spatial component is neglected and only the spectral component is compressed with a simple autoencoder network with linear layers as displayed in Figure 11.

Rate–Distortion Autoencoders. Rather than relying only on dimensionality reduction, these autoencoders are trained on a rate–distortion objective. Most current lossy neural compression methods are based on this idea, leveraging the paradigm of end-to-end learned nonlinear transform coding introduced in Section 2.1.2. In image compression, the networks are typically CNNs trained end-to-end with a rate–distortion objective. Following the seminal work of Ballé et al. [63], which demonstrates that end-to-end learned neural compression approaches can outperform hand-crafted methods, notable progress has been made applying these techniques to remote sensing data. Various approaches demonstrate the application of end-to-end learned architectures in the areas of optical satellite imagery [122, 136, 154, 83], aerial imagery [170] and Synthetic Aperture Radar (SAR) data [169]. CNNs are suitable for capturing spatial features. By increasing the size of the convolutional kernels, information over large ranges can also be captured well, which plays an essential role in the

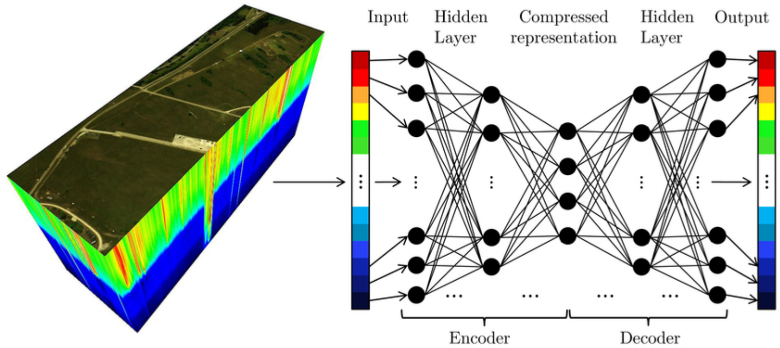

Figure 11: Autoencoder for hyperspectral data by Kuester et al. [113]. Figure taken from the original paper.

extraction of representations of images. However, the use of large convolutional kernels leads to an increase in the complexity of the model.

Reduced–complexity Rate–Distortion Autoencoders. For on-board deployment of neural compression algorithms, the computational complexity must be kept within the limits of the available hardware. Research in remote sensing compression thus focuses on adapting architectures to reduce computational complexity.

Alves de Oliveira et al. [122] propose a reduced-complexity variational autoencoder that outperforms traditional transform coding methods such as JPEG2000 [27]. By reducing the number of network parameters and simplifying the entropy model, they design a lower complexity compression architecture that maintains competitive performance with other neural models. The fitting of hyperpriors or nonparametric distributions is replaced by a parametric estimation of a Laplacian distribution that effectively represents the embedded satellite images. Despite these advances, these autoencoders still incur a high computational cost compared to traditional methods. Mijares i Verdú et al. [188] extends the hyperprior VAE model to hyperspectral data by clustering the input bands of an image into groups of three, whereby a separate neural model is applied to each group, which reduces the computational complexity.

Instrument diversity. Different works have also explored how neural compression can be tailored to the wide variety of data within the remote sensing domain. For aerial imagery, Zhu et al. [170] adapt an end-to-end multispectral compression method based on CNNs for UAV data. Their approach incorporates radiation calibration to achieve a more even distribution of input data and integrates 1×1 convolutions to better utilize interspectral redundancies. In the field of SAR data, Xu et al. [169] proposed a CNN model for end-to-end SAR image compression based on a variational autoencoder with a hyperprior. Di et al. [157] proposed to utilize pyramidal feature extraction to better capture the redundancy between image pixels. Their approach for compressing SAR images involves a single Gaussian hyperprior framework and the pyramidal encoder to capture both coarse and fine structures, along with a quality enhancement module based on a dense residual network.

Architectural adaptations. Many works build on the hyperprior model [82] through architectural adaptations to the neural networks in the transform step. Xiang et al. [193] incorporate long-range convolution and attention mechanisms to identify spatially redundant features. Their

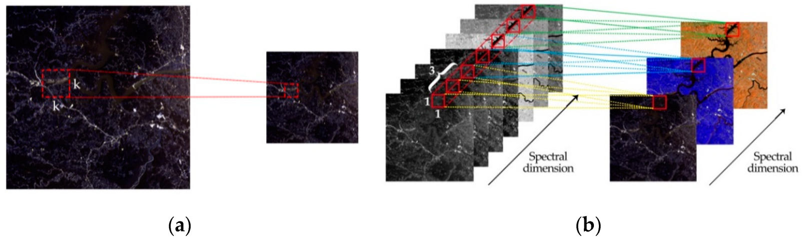

Figure 12: Spectral convolution taken from Kong et al. [136]. (a) 2D convolution; (b) 1D spectral convolution. Figure taken from the original paper.

improved non-local attention model reduces computational complexity while outperforming standard neural methods. Kong et al. [135] also modify the encoder to extract spatial-spectral features at multiple scales and adaptively adjust the weights of the features from the different branches of the encoder network. To leverage both local and non-local redundant features, Fu et al. [175] propose a mixed hyperprior net. This is based on a hyperprior VAE but employs two prior models: a transformation-based prior to capture global redundancy and a CNN-based prior to capture local redundancy. The model is applied to high-resolution remote sensing images (HRRSI) and enables high-resolution reconstruction images even with high compression rates.

Gao et al. [177] also extend the hyperprior model by implementing an enhanced residual attention module (ERAM) in the primary and hypercoder. This module applies a spatial attention mechanism to generate importance masks for potential features, later used to adjust the bit distribution over latent channels. This is motivated by an analysis showing the varying importance of latent features to the distortion measure. Their approach additionally uses a discrete Laplacian mixture entropy model instead of a piecewise linear function.

Another common strategy is to decompose the spatial and spectral components in the model. This contrasts with the common framework of neural compression approaches [82], where dependencies between the image channels are not explicitly considered. Kong et al. [136] propose adaptations to treat spectral and spatial dimensions separately. They introduce an encoder with a feature extraction module divided into two parallel parts and extract spectral and spatial features independently, fusing them later for further processing. Figure 12 shows spectral convolution which is used for the spectral feature extraction module. This method outperforms neural compression with unified extraction and hand-crafted methods on Landsat satellite images. Similarly, Cao et al. [154] handle the extraction of spectral and spatial features independently. In contrast to [136], spatial and spectral features are not fused for further processing but are extracted completely separately. They incorporate Tucker decomposition through tensor layers [83] for better decomposition of the multi-way data representations. While not always motivated by complexity reduction, spatial-spectral decomposition can also lead to more computationally efficient models, particularly when dealing with a large number of spectral bands.

Hybrid methods. Some works combine the wavelet transform from traditional compression with neural architectures. Anuradha et al. [197] employ the DWT for spatial-spectral decorrelation in combination with LSTM networks for hyperspectral image compression. In order to more effectively differentiate between high-frequency and low-frequency information, Xiang et al. [194] build a neural compression model with a DWT module and Gaussian mixture models for entropy estimation shown in Figure 13. The DWT is employed to transform the latent representations of

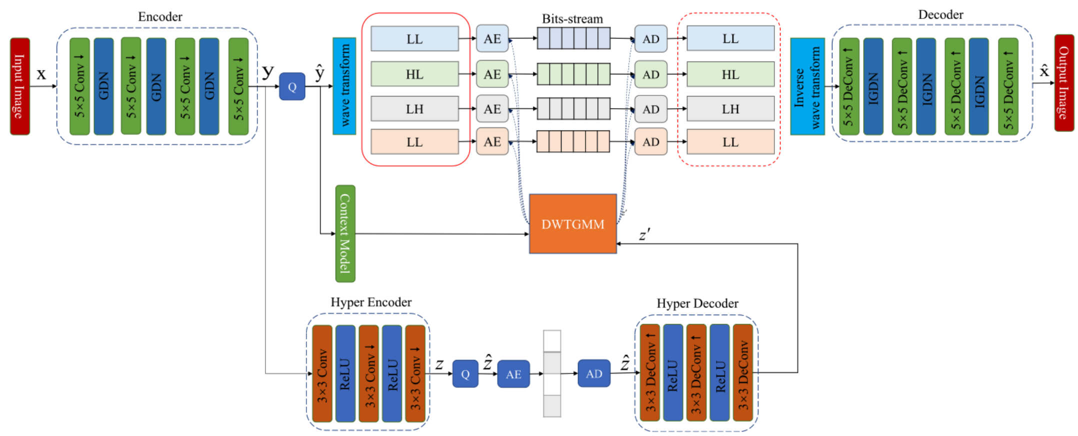

Figure 13: Neural compression architecture with an incorporated non-learnable wavelet transform introduced by Xiang et al. [194] as a Discrete Wavelet Transform-Based Gaussian Mixture Model. Figure taken from the original paper.

a neural encoder into multiple sparse representations, for which separate probability distributions are then estimated using a hyperprior and context model. This work thus combines an end-to-end image compression scheme with a non-learnable transformation component.

Xiang et al. [213] also addresses the challenge of reconstructing high-frequency information in remote sensing images, which often leads to edge-blurred artifacts. It introduces 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. As in the previous work, Gaussian mixture models are used to estimate entropy models for the high- and low-frequency components separately.

Explicit bitrate allocation. Some approaches add a mechanism for the neural codec 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, which can be used to weigh the compression of certain areas of an input image. Ye et al. [214] use this technique to address the challenge of preserving fine details in remote sensing images at high compression rates within a VAE hyperprior model. Their method employs 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 et al. [203] integrate a quality map into a hyperprior VAE architecture. This is designed to maintain a high information content in the regions of interest while reducing redundancy in non-target areas. Rate allocation, guided by a quality map created with a pre-trained Vision Transformer (ViT) network, serves to minimize redundancy and effectively balance focused and unfocused regions.

Generative Adversarial Networks. GANs have also been used for compression of satellite imagery. GAN compression models have shown impressive performance in the low-bitrate regime.

Leveraging the VQGAN [131] architecture, cf. Figure 14, such models usually consist of autoencoders employing GANs as decoder modules, and tailoring the associated adversarial loss to favor specialized [185] or generalist [151] compressed representations.

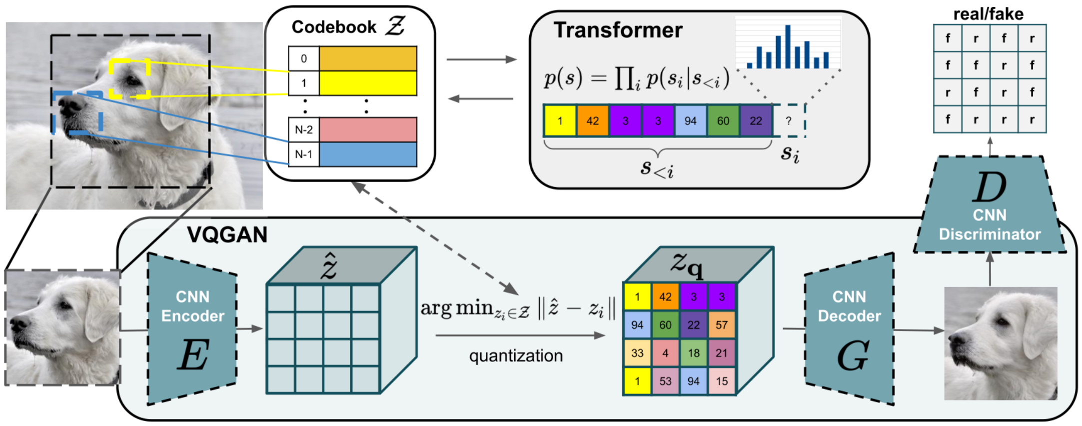

Figure 14: *Vector-Quantized Generative Adversarial Network* (VQ-GAN) architecture introduced by Esser et al. [131] as extension of the *Vector-Quantized Variational Autoencoder* (VQ-VAE) substituting the pixel-based image reconstruction loss by a discriminator network D. Figure taken from the original paper.

Zhao et al. [151] optimizes for the visual realism of reconstructed images by including a perceptual similarity term within the adversarial loss of a Conditional GAN [54] decoder. On the other hand, Li et al. [185] focuses on the generation of generalist compressed representations using Least Squares GANs [76] to reconstruct (dense) low-frequency components from (sparse) high-frequency components of the same original images.

Implicit Neural Representations. Alternative approaches to neural image compression exist that do not rely on autoencoder backbones. In this regard, methods based on Implicit Neural Representations (INRs) proved capable of outperforming JPEG2000 on both multispectral [187] and hyperspectral [215] datasets.

In general, INR-based methods aim at regressing the channel values for each pixel of a given image from the corresponding pixel coordinates, or transformations thereof. By optimizing for the fidelity of the regressed channel values, a neural network is trained that encodes the implicit mapping between the spectral values and the location (or information associated with it) of each pixel. The set of trained weights then undergoes quantization and entropy coding, thus acting as a compressed representation for images in the training dataset.

Li et al. [187] successfully apply INRs to multi-spectral image compression. In this work, the authors train an MLP with equally sized residual layers to predict pixel values from longitude and latitude coordinates respectively 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. Based on that upper bound authors show their methods match the quality of reconstruction of JPEG2000 with half the number of bits encoding each pixel.

A similar approach to hyperspectral imagery is implemented within the FHNeRF [215] model. It aims at regressing hyperspectral pixel values from transformed pixel coordinates using Neural Radiance Fields [141]. The proposed model almost doubles the reconstruction quality of traditional and autoencoder-based neural compressors at comparable compression ratios.

Arguably, the main advantage of INR-based compressors is that they are agnostic to some inherent features of the original images, 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 neural compressors, 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. The idea of compression feature representations of data suitable for being processed for downstream tasks by algorithms, rather than for reconstruction, has also seen some preliminary exploration in earth observation. Furutanpey et al. [206] leverage this approach to mitigate the bandwidth bottleneck between satellites and base stations. They design an end-to-end pipeline for on-board feature compression capable of producing task-agnostic features as well as perceptually similar reconstructions of the input data. Their evaluation on benchmarks for object detection from aerial images shows improved performance compared to neural image codecs and existing neural feature compressors [161, 205].

Gomes et al. [208] use the same idea tailored to the transmission of features from data centers to end users hosting models for training or inference. They use a rate–distortion objective that combines masked auto-encoding as a form of self-supervision [159] with an entropy penalty to encourage compressible, general-purpose features. They further leverage an existing foundation model and show that fine-tuning a small portion of the pre-trained weights with this objective is sufficient to turn it into a general feature compressor for classification and

3.2.3 Dictionary learning

Dictionary learning approaches remain popular in the remote-sensing compression literature. The approach involves learning a set of basis elements (a dictionary) from the data, and then representing data as sparse combinations of these elements, enabling efficient compression.

In Earth observation, Wu et al. [60] propose a hyperspectral image compression method based on the double sparsity model, which efficiently captures and sparsifies signals using a learned sparse dictionary. Their method, which involves sparse and entropy coding with Differential Pulse Code Modulation (DPCM) and arithmetic codec, demonstrates superior rate–distortion performance and better spectral information preservation than 3D-SPIHT and JPEG2000. Similarly, dictionary learning is at the core of the work by Wang et al. [80], which utilizes a lossy hyperspectral data compression method with sparse representation by learning a dictionary that induces sparsity in the coefficient vectors of input signals. The approach leverages the energy compaction feature of sparse coefficients within a lossy compression framework, demonstrating competitive performance with state-of-the-art methods like JPEG2000 and 3D-SPIHT. Dictionaries exploiting spectral and spatial correlations are trained using online dictionary learning, and hyperspectral data is represented via sparse coding with these learned dictionaries. The sparse coefficients are then quantized and entropy-coded to form the final bit stream. The framework supports using a base dictionary trained offline or updating the dictionary for enhanced adaptivity, with different compression levels achieved by allowing varying numbers of non-zero coefficients.

Ertem et al. [109] also employ dictionary learning to improve hyperspectral image compression. This method generates superpixel maps for adaptive spatial–spectral representation, computes an optimal dictionary, and determines sparse coefficients using Simultaneous Orthogonal Matching Pursuit (SOMP). Innovations include a modified dictionary learning step, an ordering scheme that eliminates the need to send the superpixel map as side information, and using DPCM to reduce sparse coefficient magnitudes. Their method was compared with others, including JPEG2000, on datasets like Indian Pines and Washington DC Mall, demonstrating better performance in quality metrics and anomaly-preserving performance.

3.3 Summary

To conclude this section, we summarize the current trends in neural compression for satellite imagery.Despite the persisting popularity of dictionary learning, recent work seems to favor building on neural methods that directly optimize a rate–distortion objective, in particular the hyperprior approach introduced by Ballé et al. [82]. The flexibility of this approach concerning the form of the synthesis and analysis networks has enabled researchers to easily experiment with different architectures, with the "Transform" axis identified in Section 2.2 being the target of most investigations.

Motivated by the relatively static nature of consecutive observations of the earth's surface for most geographical locations, Wang et al. [92] use ideas from traditional video compression to compress sequences of satellite images. Recent work from Du et al. [204] has shown potential in exploiting the same characteristic through reference-based coding, where for a given earth observation from a satellite, a historical observation from the same region is used as a reference, and only the difference must be transmitted.

INRs also pose an interesting research direction, in particular for hyperspectral data, usually consisting of a very large volume of highly correlated signals for which large training datasets are typically less available than for multispectral or RGB data. The variety of earth observation instruments leads to a very wide application space for compression. This results in different works training and evaluating on different datasets, rendering comparisons between methods extremely difficult.

Finally, recent work in neural feature compression shows promise in earth observation 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.

4 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; individual grid cells in current climate models are usually on the scale of around 100 kilometers (kms). However, with such a coarse resolution many important processes, such as precipitation and deep convection, cannot be fully resolved which motivated the development of the next generation of climate models with grid cells on the scale of 1-5 kms [50, 55, 77, 102, 123]. The increased resolution of these models also implies that they produce tremendous amounts of data [58, 110]. For example, the recently launched Destination Earth initiative generates around 1 petabyte of data *per day*.[1] The wealth of data generated by these models leads to new operational constraints for climate and weather prediction; it is often no longer feasible to store all the generated data on disk.

4.1 Challenges

4.1.1 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 speed, geopotential, temperature, etc., which are localized in space and time and are stored in multidimensional arrays. 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, time, and between the different variables. Furthermore, while the output is generally saved in multidimensional arrays in data formats such as NetCDF and Zarr, different climate models can use different grid projections for modeling purposes.

- **Small spatial scales.** Small scales are important (this is why we need km-scale models in the first place). 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. However, the climate system is inherently chaotic. Hence, accurately modeling the evolution of the dynamics at smaller scales is important for accurately modeling the larger scales dynamics. A key motivation for using km-scale models in the first place is to be able to model smaller scale phenomena. Hence, compression algorithms that overly smooth data over small scales may result in undesirable or unpredictable downstream effects.

- **Extreme events.** A key goal for climate modeling is to predict the probability of extreme events such as floods or storms. 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.

- **Lack of quantitative metrics.** As outlined in Section 2, compression methods are usually evaluated based on how well they reconstruct the input data. This requires a quantitative

[1] https://stories.ecmwf.int/the-digital-twin-engine/

measure to compare the original data, $\mathbf{x}$, and the reconstructed data, $\mathbf{x}'$. However, in climate sciences, there is a general lack of quantitative metrics that can do this comparison. Classical metrics such as the pixel-wise mean-squared error are often insufficient to capture the structural differences between inputs and reconstructions that climate scientists are interested in. Ideally, any reconstruction should conserve the physical properties of the input, e.g. individual clouds should have the same mass in the input data and the reconstruction and they should be arranged in a physically consistent way. However, having metrics that can easily and cheaply capture this is still an active research area.

4.1.2 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. Despite the power of supercomputers, completing a single run can take several days. Consequently, data generated from these models is typically produced once and then stored for subsequent access by scientists. Researchers often need only a subset of the data, such as specific time periods or particular variables, for their analysis 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 in compressing it if this reduces the bandwidth required for transmitting the data to scientists for further analysis.

4.2 Classification of Compression Methodologies

For climate data, lossless compression algorithms do not provide compression ratios that meaningfully reduce the disk space of the data [167]. Hence, most work investigates the opportunities of using lossy compression algorithms. Most climate data, as we outlined above, is stored in multi-dimensional arrays, hence efficient compression algorithms need to exploit the correlations between the different dimensions of the data.

Most existing codecs for multi-dimensional arrays are hand-engineered (in the sense of Section 1.2) and employ the transform-based compression approach described in Section 2. A key to achieving good compression ratios is to exploit correlation in the data; many codecs divide the input data into sub-blocks[2] and try to identify correlations within a sub-blocks. SZ3 [160] uses a spline-based interpolator as a transform to identify correlations in a given sub-block. ZFP [53] instead decorrelates individual sub-blocks using an orthogonal transform. TTHRESH [93] uses a generalization of the singular value (SVD) decomposition to tensors with more than three dimensions to transform the data.

Several studies have evaluated the impact of lossy compression on the output of climate simulators [70, 90, 167]. Specifically, Underwood et al. [167] develop a set of "assessments" that a compressor needs to pass (e.g. sufficiently high SSIM score) in order for it to be safely used. They identify SZ3 as the best lossy compressor but at the same time emphasize that more work is needed to design quantitative metrics to compare the performance of different compressors.

Some compression methods directly exploit the fact that the data was generated from a chaotic system. When simulating a chaotic dynamical system in 32-or 64-bit precision then a few of the least significant bits in the IEEE floating point representation will generally constitute random

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

noise. These bits are then not useful in predicting the future state of the system and can therefore be discarded. This gives rise to the notion of how much "real" information there is in the bit-representation for a given variable [74, 134]. This has been the motivation for a couple of compression schemes [67, 134] which (sometimes adaptively) 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 paragraphs because it can simply be run as a pre-processing step before passing the dataset to a compressor.[3]

Compared to other data modalities such as images, text, or video there has been relatively little work on developing neural compression methodologies for climate data. Existing work mainly focuses on using implicit neural representations (INRs) to overfit a single neural network to a single dataset [158, 180]. Specifically, [180] adapts INRs to climate and weather data by transforming the input data using Fourier features [31, 120]. However, methods based on INRs tend to smooth out extreme values which is generally undesirable. Overall, the development of neural compression methods for climate and weather data is still an area of open research.

[3]The same argument has also been used to justify running climate and weather models with lower bit representation [112, 152, 138].

5 Neural Compression: Implementation & Application

The process of using neurally compressed embeddings of Earth observation data varies largely from using the images they are based upon. Therefore, in this chapter, we want to speculate about the impact the use of compressed embeddings could have on not only storing and transferring the data, but also on the analysis and application that are made possible.

5.1 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 expenditures. Despite the broad significance of EO data across various fields [65], a compression method for EO data that would effectively balance storage saving and processing speed is still not available. This renders at impossible processing of the data on a large scale in environments other than the one it is already stored in. High costs of transferring the data to another computing environment plays a major role in federated geospatial analytics. However, as demonstrated in the previous sections, using neural compression algorithms can drastically reduce the size of EO imagery. Therefore, a compression scheme based on deep neural networks can contribute to solving the aforementioned challenges created by the large size of EO data.

Spaceborne sensors usually have a limited lifetime of less than a decade and may be decommissioned much earlier if technical failures occur, e.g., as with the interruption of service of the Sentinel-1B satellite [163]. Although follow-up missions often fill the data gap, these come with updated hardware and, more importantly, slightly different acquisition modes, spectra, orbiting properties, and inclination angles. That all leads to the acquired data formats and statistics to differ from the previous products. Foundational models for the decommissioned missions may therefore under-perform or fail entirely for the newly acquired data. At the same time, it is desirable to keep performance as high as possible for the older, archival data to allow for applications like long-term timeseries analysis or change detection across different sensor generations. This calls for flexible, interpretable (and possibly modular) formulations of foundational models that can be updated with new data of (slightly) different characteristics at low computational cost.

Scaling traditional EO data flows often leads to transmitting unnecessary data because each step in the pipeline operates in isolation to facilitate parallel processing over large areas. This issue becomes more pronounced when integrating data from different modalities, particularly when data resides on disparate (cloud) computing platforms. High compression of data transferred between steps can significantly reduce data size, though decompression during processing may be time-consuming. Further, the significantly reduced amount of data that must be transferred when working with such neurally compressed data can lead to a proliferation of large-scale models in the context of environmental monitoring: the reduced size 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 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. 15.

General-purpose embeddings $\mathbf{z}$ generated by foundation models created in the transformation step f (cf. Section 2.1.2) have shown to be effective in training task-specific heads for various downstream tasks [192]. Combining these with neural compression as discussed in Section 2.3.4 optimizes data storage and transfer, while enabling diverse image analysis tasks on compressed

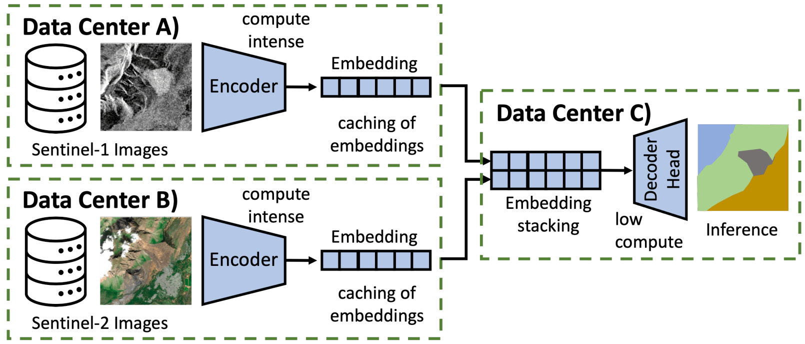

Figure 15: Concept of data federation through compressed embedding sharing between data centers.

image vectors. Compressed embeddings strike a balance between storage and processing: the computationally intensive generation of embeddings can be performed on high-performance computing (HPC) systems, while lightweight decoding is reserved for final applications. Intermediate steps can thus operate on the reduced data size provided by the embeddings. The shift from raw data to neurally compressed embeddings shows potential for quicker data analysis, improved modeling accuracy, and advanced simulations, all while reducing data transfer (e.g., egress) and storage costs. Moreover, working with image vectors, particularly when stemming from foundation models, demonstrated high capabilities in few-shot learning [173], thereby substantially mitigating resources needed for data labeling while maintaining high task accuracy.

While the use of deep neural compression of EO data can contribute to solving the challenges of storage, transfer, and processing outlined above, the process of neural compression itself yields new challenges that must be overcome to work with neurally compressed EO data.

Firstly, when working with neurally compressed embeddings, certain use cases could benefit from processing them in cloud environments rather than locally: Even though applying neural compression achieves significant compression of EO data, the amount of data needed to process high spatial and temporal extent may still be very large. Further, training task-specific machine learning models for processing the downloaded general-purpose embeddings may also - depending on the task and the model's architecture - require strong computational power. Lastly, not everyone may have the appropriate technical infrastructure at hand to perform neural decompression which requires passing the data through another pre-trained neural decoder network to reverse the transform coding scheme (see Section 2.1.2). For these reasons, to fully reap the benefits of neurally compressed EO embeddings, cloud platforms focused on processing EO data need to have the capability of loading and then processing the neurally compressed embeddings. Currently, neither proprietary EO cloud platforms such as Google Earth Engine [72] nor the OpenEO standard [146] for interacting with EO cloud computing backends offers support for such neurally compressed embeddings.

Secondly, an open standard is needed for storing and transferring the compressed embeddings, for example between storage and processing environments. To foster interoperability between systems, such a file format must contain all information necessary for it to be decompressed to retrieve the neurally encoded information. Yet, to our knowledge such a standard does currently not exist. At the time of writing, the cloud native geospatial foundation [217] is conducting a survey about how the geospatial community is storing embeddings in GeoParquet, with the aim of formulating guidelines or introducing a standard on storing and exchanging embeddings in the future. While this survey may develop into a solution for storing raw embeddings $\mathbf{z}$, it still does not solve the challenge of creating a data format for storing neurally compressed embeddings in a standardized way. Along with the compressed data, metadata must be provided to give insight into e.g. what foundation model was used generate the embeddings $\mathbf{z}$, and the geographic area to which these embeddings apply. To make the neurally compressed EO data findable in terms of FAIR data principles [66], they should be cataloged in a standardized way, for example, by registering them to a Spatio-Temporal Asset catalog (STAC) with sufficient metadata.

Conclusively, we argue that existing data catalogs and EO processing platforms must be extended to support the provision and processing of (neurally compressed) embeddings. Only then it is possible to fully materialize the advantages of working with embedding representations instead of the base imagery.

5.2 Cost- and Energy-Efficiency & Latency

Energy efficiency is crucial in remote sensing; an example is the need to maximize the operational lifespan of nanosatellites, ensure effective data transfer within limited downlink windows, and reduce operational costs [207]. Efficient onboard data processing reduces the volume of data needing transfer, mitigating bandwidth constraints and allowing more critical data to be prioritized. Advances in Orbital Edge Computing and neural feature compression 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 Earth observation data.

However, the approaches reviewed have significant impact also for the ground operation of the data infrastructure. In this section, we aim at giving back-of-the-envelope estimations that semi-quantitatively seek to asses the role that neural compression can have in geospatial analytics. A concise system analysis of the information and communication technology infrastructure is not in the scope of this review. Also, a detailed scenario analysis is not manageable. However, we aim at providing a first orientation with a simple model calculation. The demonstrative case we study is the Copernicus program, with its main data sources being the Sentinel satellites. We investigate the hypothetical case, where Copernicus data products are compressed and potentially decompressed on the consumer side, and compare to the state without the compression. The focus lies on energy efficiency, but aspects of cost-efficiency and latency will be included.

According to the Copernicus Data Dashboard[4], by the time of writing this article, the total volume of data products grow by 759 TB per month, and 6.2 PB of data products are downloaded in the respective period, which we coarse round to 10 and 100 PB/year. Price indications for data transport out of cloud storage (egress costs) are given on the website of Amazon Web Services (AWS)[5]. While the detailed pricing depends on the site of host and consumer, 20 USD/TB is a realistic lower bound for 2024, however this cost includes a high quality-of-service. Hence, a gross data transfer cost of 2.000.000 USD/year for data product download is a first estimate.

The energy footprint of data transfer is very difficult to assess. For example, a meta-study by Aslan et al. summarizes 14 studies that even after adjusting the system boundaries deviate by a factor of ten and more[81]. The web page *wholegraindigital* elucidates the challenge of defining suitable system boundaries[6]. The authors challenge the idea of a single metric measuring the energy

[4]https://dashboard.dataspace.copernicus.eu, accessed on 2024/06/27
[5]https://aws.amazon.com/s3/pricing/?trk=ap_card
[6]https://www.wholegraindigital.com/blog/website-energy-consumption/

of data transfer and point out that Aslan et al. only evaluate the usage of a subsystem. However, as a starting point, we pick the number of 0.01 kWh/GB, carefully following the extrapolation in Fig. 3 of Aslan et al. but correcting upwards. With this assumption, the annual energy cost of data transfer sums up to about 1 GWh/year.

With these very coarse assumptions, we can proceed towards identifying the saving potential. We assume a neural compression algorithm is used for 50% of the data products downloads and achieves a compression factor (e.g. compression ratio of 100x) so that the data transfer of the neural compression can be neglected in comparison to the standard transfer. Then, an energy saving potential of 500 MWh/year seems possible.

However, the compression is associated with energy consumption. We propose a simple order-of-magnitude estimate to assess the consumption that transfers the insights from computer vision to remote sensing using BigEarthNet as intermediate, where multispectral data is brought to a similar form as natural images [103]. Typical convolutional encoder networks from computer vision require between several and 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 the context of the ImageNet datasets [139, 33].

For transformer architectures that have gained a lot of traction also in the field of computer vision, 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 $6\dot{1}0^{16}$ FLOPs in total, or about 10^{15} FLOPs/GB.

The GPU that currently dominates AI compute centers, Nvidia's A100 GPUs can, according to our experience in Ref. [133], with moderate optimization sustain 50% of their nominial performance using fp16 accuracy throughout the ML workloads. Hence, they can achieve approximately 150 TeraFLOPs per second at a power consumption of 400 W[7]. 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 11.000 kWh for compressing the entire data. 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.

Comparing the potential energy savings and the compression, despite all uncertainties, it is apparent that a one-off compression of the data is almost negligible and appears as it is two orders of magnitude apart from the transfer consumption. As a last 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 decompresses 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) it is required to consider the energy consumption of the decompression. In many encoder-decoder architectures, computational efforts of both are balanced between both. In the considered case, this would in turn produce a computation effort of approximately 10^{15} FLOPs/GB, but it is important to point out that the overall data consumption at present is approximately 10x higher than the amount of incoming data. Furthermore it is important to consider with which computational devices (CPUs,

[7] https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf

deprecated GPUs, etc.) and with which expertise the decompression is performed. For example, the energy efficiency can be 1-2 orders of magnitude worse, compensating the saving of the efficient transfer entirely. However, projects like `llama.cpp`[8] indicate that with advanced techniques such as model quantization, even computationally demanding ML models can be executed on a wide variety of hardware systems.

Finally, we would like to share two estimates demonstrating the possible latency improvements when employing neural compression: A researcher may want to perform a spatio-temporal analysis over 10 years worth of multispectral imagery across all of Germany. For May 2024, the sum of all Sentinel-2 images available for the entire country equates to 470GB (L1C data product) in JPEG2000 format. This extrapolates to an estimated volume of 56TB per decade. With an internet download speed of 100MBit/s, the data is available with a delay of about 52 days, while a hypothetical neural compression ten times more efficient compared to JPEG2000 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 is covers an area of approximately $148\,940\,000\,\mathrm{km}^2$, and given a single Sentinel-2 tile covers about $10\,000\,\mathrm{km}^2$, that amounts to roughly $15,000$ tiles. With one tile consuming 0.8GB in size, that amounts to 12TB of data downloaded volume to generate a landcover product again in JPEG200. With the same internet download speed from the previous example, we obtain a latency of 11 days compared to about 1 day when neural compression gets applied.

In 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, and a reduction of the latency that can make data-intensive application also possible for consumers with comparably lower bandwidth. However, the analysis is coarse, almost simplistic and suffers from methodological problems regarding data transfer energy consumption. Especially the consumer side requires a careful considerations, potentially education and training material to ensure that the energy savings are not overcompensated on the consumer side.

Future projections are extremely challenging. Nvidias Road Map presentation at Computex [9] indicates optimism about future performance and energy efficiency gains that make compute-heavy approaches more attractive. Comparably, the cost of data transfer is decrease as well. However the complex interplay of efficiency and demand with focus on data transfer energy demand is explained by Koomey and Masanet [137]. The increasing energy efficiency can lead to reduced energy consumption, but a more accessible resource can generate increasing demand that overcompensate the energy savings.

5.3 Democratization for Applications

As discussed in the previous sections of this work, the optimized compression of the massive raster data generated by Earth observation systems and climate simulators shows potential for reducing the energy cost and transmission latency both for data-distributing platforms and their end users. As a result, stakeholders with limited compute or bandwidth may access scientific data previously out of reach for their resources. Moreover, when applied to embeddings generated by large pre-trained models, neural compression would also permit downstream users with modest compute and deep learning expertise to benefit from expressive feature representations without the need for

[8] https://github.com/ggerganov/llama.cpp

[9] https://blogs.nvidia.com/blog/computex-2024-jensen-huang/

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.

5.3.1 Global Vegetation Structure Analysis

Worldwide mapping of vegetation properties is of prime importance for understanding the global carbon cycle [95], the impact of human activities on carbon emissions [132], and the study of ecosystem services [87]. 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 [96, 140].

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 remote sensing data with terrestrial laser scanning (TLS) and aerial laser scanning (ALS) [153]. 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 [142]. Recently, combining spaceborne LiDAR measurements from the NASA GEDI mission [108] with Sentinel imagery has shown great potential for regressing forest biophysical variables like AGB or CH at a global scale and 10 m resolution [183].

Still, Lang et al. [183] 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 self-supervised learning leading to the current emergence of remote sensing foundation models [165, 191, 181, 210] renders possible the distribution of expressive feature representations directly usable for downstream vegetation-related tasks [211] 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 remote sensing 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 in the tropics such as palm oil or cocoa [190, 182].

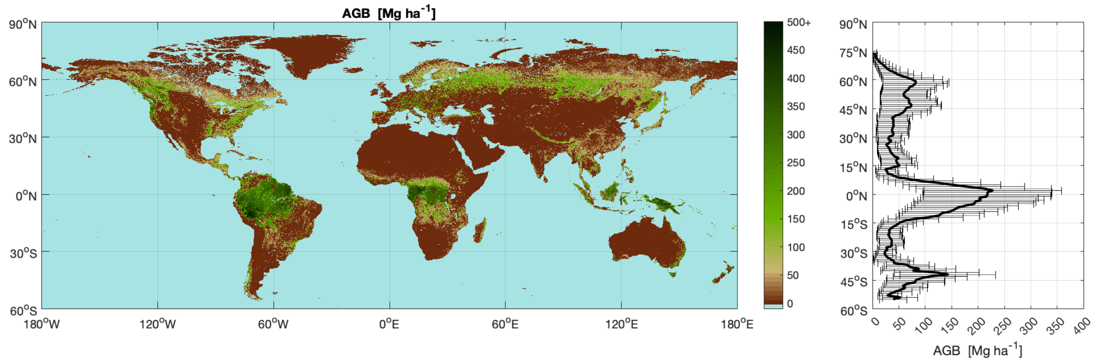

Figure 16: Aboveground biomass map from the Climate Change Initiative (CCI) Biomass project [145]. Despite its global coverage, this product has a 100 m ground sampling resolution and is only available for 2017, 2018, and 2020. This is limiting for applications needing to monitor the evolution of vegetation 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.

5.3.2 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 [35], optical modalities [41] and Automatic identification system (AIS) [56]. 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 [57]. Vessel identification could greatly benefit from neurally-compressed remote sensing maritime 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.

5.3.3 Climate and Air Pollution Prediction

As described in Section 4 high-resolution climate models are able to resolve key small scale phenomena such as clouds and ocean eddies. 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 remote sensing devices such as geostationary satellites. 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 produced by a foundation model 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.
- Using the embeddings to compare 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 [189].

Hence, the development of geospatial foundation models have the potential to significantly advance our understanding of climate dynamics and improve the accuracy of climate predictions.

5.3.4 Early Crop Stress and Yield Prediction

European agriculture is continuously affected by an increasing frequency of weather extremes [107, 111], 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 crop production [25]. Sentinel satellite missions [44] have significantly contributed to agricultural monitoring with their high temporal frequency and spatial resolution. Despite the development of crop maps and crop yield forecasting activities at the European scale [105], integrating Earth observation and weather/climate is needed to capture the effect of increasing weather extremes on agricultural phenology.

In particular, the early prediction of crop stress or crop yield at a country or continental scale could benefit from Sentinel-1/2 time series to monitor crop phenology. Indeed, satellite time series have proven to improve crop type classification[91], as they capture the dynamic changes in crops spectral and temporal signatures throughout the growing season. 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 multimodal, 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.

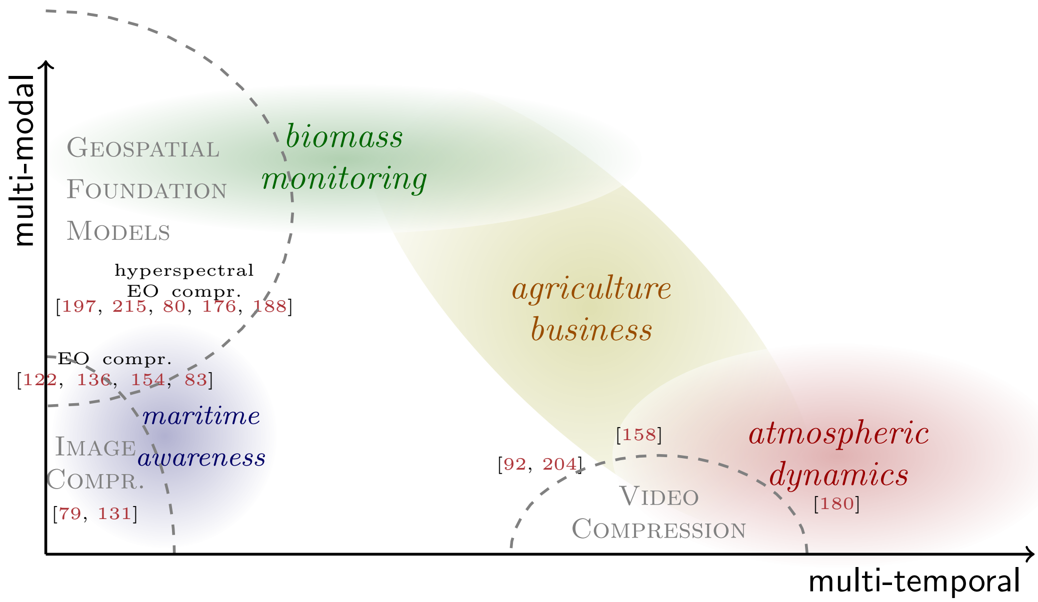

Figure 17: We represent downstream applications from the perspective of conceptual dimensions relevant for neural compression methodologies: multi-temporality and multi-modality. References to existing neural data compression literature are positioned with respect to these concepts. A gap in the literature can be observed for the compression of multimodal and multi-temporal data, showing potential for several downstream applications.

6 Perspectives & Recommendations

We conclude our literature survey on neural compression for geospatial analytics by a summary presented through Fig. 17. Novel methodologies to compress Earth Observation (EO) and Earth System Model (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 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 foundation models 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 towards 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. A dataset meant as a benchmark for learned compression for the wide range of existing remote sensing 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 earth observation modalities is critical in enabling improvements in methodology in the field.

Further development of the transform f for neural compression to better suit different modalities within remote sensing 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 remote sensing compression. On the other hand, other differentiating data characteristics in this domain seem relatively underexplored as of yet regarding their integration in neural compressors. For instance, remote sensing 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 Earth Observation, 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 Earth Observation, neural compression for climate model data has little to no track record in the academic literature. Existing approaches for compressing the outputs of climate simulators mainly rely on hand-engineered transform coding schemes. Data-driven neural compression offers an attractive alternative. A major challenge poses modeling of correlations over long, climate-relevant timescales. In principle, neural compression schemes should be able to efficiently exploit these correlations but their application to this problem domain has been relatively under-explored.

However, the biggest 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. 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 neural compression 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 [121, 28], mechanistic modeling [49], and the existing rich statistics toolbox including significance tests [9]. Furthermore, well-calibrated uncertainties can work as a natural link to physics-based forward simulations in computational science [101], e.g. to tightly integrate radiative transfer models with learning-based approaches in remote sensing. 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, foundation models capable of uncertainty prediction could, for instance, identify strategical training samples within an active learning setting.

References

[1] Kari Karhunen. "Zur spektraltheorie stochastischer prozesse". In: *Annales Academiae Scientiarum Fennicae. Series A. I. Mathematica-Physica*. 1946.

[2] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* (1948).

[3] David A. Huffman. "A Method for the Construction of Minimum-Redundancy Codes". In: *Proceedings of the IRE* (1952).

[4] L.R. Tucker. "Some Mathematical Notes on Three-Mode Factor Analysis". In: *Psychometrika* (1966).

[5] Nasir Ahmed, T. Natarajan, and K.R. Rao. "Discrete Cosine Transform". In: *IEEE Transactions on Computers* (1974).

[6] Michel Loève. *Probability Theory*. 4th. 1977.

[7] J. Rissanen and G. G. Langdon. "Arithmetic Coding". In: *IBM Journal of Research and Development* (1979).

[8] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*. 1986.

[9] Erich Leo Lehmann, Joseph P Romano, and George Casella. *Testing statistical hypotheses*. Vol. 3. Springer, 1986.

[10] Sonehara et al. "Image data compression using a neural network model". In: *International 1989 Joint Conference on Neural Networks*. 1989.

[11] Gregory K. Wallace. "The JPEG Still Picture Compression Standard". In: *Commun. ACM* (1991).

[12] Ingrid Daubechies. *Ten lectures on wavelets*. USA, 1992.

[13] Karlheinz Brandenburg and Gerhard Stoll. "ISO-MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio". In: *Journal of the Audio Engineering Society* (1994).

[14] Geoffrey E. Hinton and Richard S. Zemel. "Autoencoders, Minimum Description Length and Helmholtz Free Energy". In: *Advances in Neural Information Processing Systems*. 1994.

[15] Christopher M. Bishop. "Latent Variable Models". In: Dordrecht, 1998.

[16] Emmanuel J. Candès and David L. Donoho. "Ridgelets: A Key to Higher-Dimensional Intermittency?" In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* (1999).

[17] P. Hou et al. "Improving JPEG performance in conjunction with cloud editing for remote sensing applications". In: *IEEE Transactions on Geoscience and Remote Sensing* (2000).

[18] P. Luigi Dragotti, G. Poggi, and A.R.P. Ragozini. "Compression of multispectral images by three-dimensional SPIHT algorithm". In: *IEEE Transactions on Geoscience and Remote Sensing* (2000).

[19] V.K. Goyal. "Theoretical foundations of transform coding". In: *IEEE Signal Processing Magazine* (2001).

[20] Sunghyun Lim, Kwanghoon Sohn, and Chulhee Lee. "Compression for hyperspectral images using three dimensional wavelet transform". In: *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217)*. 2001.

[21] D. Markman and D. Malah. "Hyperspectral image coding using 3D transforms". In: *International Conference on Image Processing*. 2001.

[22] H.S. Lee, N.H. Younan, and R.L. King. "Hyperspectral image cube compression combining JPEG-2000 and spectral decorrelation". In: *IEEE International Geoscience and Remote Sensing Symposium*. 2002.

[23] I. Hacihaliloglu and M. Karta. "DCT and DWT based image compression in remote sensing images". In: *IEEE Antennas and Propagation Society Symposium*. 2004.

[24] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* (2004).

[25] Michael A White et al. "A global framework for monitoring phenological responses to climate change". In: *Geophysical Research Letters* (2005).

[26] Shuyuan Yang, Min Wang, and Licheng Jiao. "Compression of Remote Sensing Images Based on Ridgelet and Neural Network". In: *Advances in Neural Networks – ISNN 2005*. 2005.

[27] Pen-Shu Yeh et al. “The new CCSDS image compression recommendation”. In: *IEEE Aerospace Conference*. 2005.

[28] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[29] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. 2006.

[30] Qian Du and James E. Fowler. “Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis”. In: *IEEE Geoscience and Remote Sensing Letters* (2007).

[31] Ali Rahimi and Benjamin Recht. “Random features for large-scale kernel machines”. In: *Advances in neural information processing systems* (2007).

[32] Qian Du and James E. Fowler. “Low-Complexity Principal Component Analysis for Hyperspectral Image Compression”. In: *The International Journal of High Performance Computing Applications* (2008).

[33] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. 2009.

[34] Fernando Garcia-Vilchez and Joan Serra-Sagrista. “Extending the CCSDS Recommendation for Image Data Compression for Remote Sensing Scenarios”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2009).

[35] Gerard Margarit, José A Barba Milanés, and Antonio Tabasco. “Operational ship monitoring system based on synthetic aperture radar processing”. In: *Remote Sensing* (2009).

[36] Guoxia Yu, Tanya Vladimirova, and Martin N. Sweeting. “Image compression systems on board satellites”. In: *Acta Astronautica* (2009).

[37] A.Karami et al. *Hyperspectral Image Compression Based on Tucker Decomposition and Discrete Cosine Transform*. 2010.

[38] Jorge Gonzalez-Conejero, Joan Bartrina-Rapesta, and Joan Serra-Sagrista. “JPEG2000 Encoding of Remote Sensing Multispectral Images With No-Data Regions”. In: *IEEE Geoscience and Remote Sensing Letters* (2010).

[39] Nan Li and Baoxin Li. “Tensor completion for on-board compression of hyperspectral images”. In: *IEEE International Conference on Image Processing*. 2010.

[40] Iain E. Richardson. *The H.264 Advanced Video Compression Standard*. 2nd. 2010.

[41] Changren Zhu et al. “A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features”. In: *IEEE Transactions on geoscience and remote sensing* (2010).

[42] A. Karami, M. Yazdi, and G. Mercier. “Hyperspectral image compression based on tucker decomposition and wavelet transform”. In: *2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. 2011.

[43] Jonathan Masci et al. “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction”. In: *Proceedings of the 21st International Conference on Artificial Neural Networks*. ICANN’11. Berlin, Heidelberg, 2011.

[44] Michael Berger et al. “ESA’s sentinel missions in support of Earth system science”. In: *Remote sensing of environment* (2012).

[45] Azam Karami, Mehran Yazdi, and Grégoire Mercier. “Compression of Hyperspectral Images Using Discerete Wavelet Transform and Tucker Decomposition”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2012).

[46] Nicholas D. Sidiropoulos and Anastasios Kyrillidis. “Multi-Way Compressed Sensing for Sparse Low-Rank Tensors”. In: *IEEE Signal Processing Letters* (2012).

[47] Gary J. Sullivan et al. “Overview of the High Efficiency Video Coding (HEVC) Standard”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2012).

[48] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. “Estimating or propagating gradients through stochastic neurons for conditional computation”. In: *arXiv preprint arXiv:1308.3432* (2013).

[49] Andrzej Kraslawski and Ilkka Turunen. *23rd European Symposium on Computer Aided Process Engineering*. Elsevier, 2013.

[50] Bjorn Stevens and Sandrine Bony. “What are climate models missing?” In: *science* (2013).

[51] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[52] Jin Li, Fei Xing, and Zheng You. “Compression of Multispectral Images with Comparatively Few Bands Using Posttransform Tucker Decomposition”. In: *Mathematical Problems in Engineering* (2014).

[53] Peter Lindstrom. “Fixed-rate compressed floating-point arrays”. In: *IEEE transactions on visualization and computer graphics* (2014).

[54] Mehdi Mirza and Simon Osindero. “Conditional Generative Adversarial Nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[55] Tim Palmer. “Climate forecasting: Build high-resolution global climate models”. In: *Nature* (2014).

[56] Ramona Pelich et al. “AIS-based evaluation of target detectors and SAR sensors characteristics for maritime surveillance”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2014).

[57] Jiexiong Tang et al. “Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine”. In: *IEEE transactions on geoscience and remote sensing* (2014).

[58] Jennifer 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”. In: *Bulletin of the American Meteorological Society* (2015).

[59] George Toderici et al. “Variable rate image compression with recurrent neural networks”. In: *arXiv preprint arXiv:1511.06085* (2015).

[60] Qian Wu, Rong Zhang, and Fan Wang. *Hyperspectral data compression using double sparsity model*. 2015.

[61] Lefei Zhang et al. “Compression of hyperspectral remote sensing images by tensor approach”. In: *Neurocomputing*. Advances in Self-Organizing Maps Subtitle of the special issue: Selected Papers from the Workshop on Self-Organizing Maps 2012 (WSOM 2012) (2015).

[62] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. “End-to-end optimization of nonlinear transform codes for perceptual quality”. In: *2016 Picture Coding Symposium (PCS)*. 2016.

[63] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. “End-to-End Optimized Image Compression”. In: *International Conference on Learning Representations* (2016).

[64] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[65] Pratistha Kansakar and Faisal Hossain. “A review of applications of satellite earth observation data for global societal benefit and stewardship of planet earth”. In: *Space Policy* (2016).

[66] Mark D Wilkinson et al. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific data* (2016).

[67] Charles S Zender. “Bit Grooming: statistically accurate precision-preserving quantization with compression, evaluated in the netCDF Operators (NCO, v4. 4.8+)”. In: *Geoscientific Model Development* (2016).

[68] Mengfei Zhang et al. “A Low-Rank Tensor Decomposition Based Hyperspectral Image Compression Algorithm”. In: 2016.

[69] Eirikur Agustsson et al. “Soft-to-hard vector quantization for end-to-end learning compressible representations”. In: *Advances in neural information processing systems* (2017).

[70] Allison H Baker et al. “Toward a multi-method approach: Lossy data compression for climate simulation data”. In: *High Performance Computing*. 2017.

[71] Bo Du et al. “PLTD: Patch-Based Low-Rank Tensor Decomposition for Hyperspectral Images”. In: *IEEE Transactions on Multimedia* (2017).

[72] Noel Gorelick et al. “Google Earth Engine: Planetary-scale geospatial analysis for everyone”. In: *Remote Sensing of Environment* (2017).

[73] Huadong Guo et al. “Big Earth Data: a new challenge and opportunity for Digital Earth’s development”. In: *International Journal of Digital Earth* (2017).

[74] Stephen Jeffress, Peter Düben, and Tim Palmer. "Bitwise efficiency in chaotic models". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* (2017).

[75] Jin Li and Zilong Liu. "Compression of hyper-spectral images using an accelerated nonnegative tensor decomposition". In: *Open Physics* (2017).

[76] Xudong Mao et al. "Least Squares Generative Adversarial Networks". In: *IEEE International Conference on Computer Vision*. 2017.

[77] Tapio Schneider et al. "Climate goals and computing the future of clouds". In: *Nature Climate Change* (2017).

[78] George Toderici et al. "Full Resolution Image Compression with Recurrent Neural Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[79] Aaron Van Den Oord, Oriol Vinyals, et al. "Neural discrete representation learning". In: *Advances in neural information processing systems* (2017).

[80] Hairong Wang and Turgay Celik. *Sparse Representation-Based Hyperspectral Data Processing: Lossy Compression*. 2017.

[81] Joshua Aslan et al. "Electricity intensity of internet data transmission: Untangling the estimates". In: *Journal of industrial ecology* (2018).

[82] Johannes Ballé et al. "Variational Image Compression with a Scale Hyperprior". In: *International Conference on Learning Representations*. 2018.

[83] Jen-Tzung Chien and Yi-Ting Bao. "Tensor-factorized neural networks". In: *IEEE Transactions on Neural Networks and Learning Systems* (2018).

[84] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* (2018).

[85] Benoit Jacob et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[86] Mu Li et al. "Learning convolutional networks for content-weighted image compression". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[87] Peter Manning et al. "Redefining ecosystem multifunctionality". In: *Nature ecology & evolution* (2018).

[88] Fabian Mentzer et al. "Conditional probability models for deep image compression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[89] David Minnen, Johannes Ballé, and George D Toderici. "Joint Autoregressive and Hierarchical Priors for Learned Image Compression". In: *Advances in Neural Information Processing Systems*. 2018.

[90] Andrew Poppick et al. "A statistical analysis of compressed climate model data". In: *Proc. DRBSD* (2018).

[91] Francesco Vuolo et al. *How much does multi-temporal Sentinel-2 data improve crop type classification?* 2018.

[92] Xu Wang et al. "Virtual Background Reference Frame Based Satellite Video Coding". In: *IEEE Signal Processing Letters* (2018).

[93] Rafael Ballester-Ripoll, Peter Lindstrom, and Renato Pajarola. "TTHRESH: Tensor compression for multidimensional visual data". In: *IEEE transactions on visualization and computer graphics* (2019).

[94] Zhiqin Chen and Hao Zhang. "Learning implicit fields for generative shape modeling". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

[95] Pieter De Frenne et al. "Global buffering of temperatures under forest canopies". In: *Nature Ecology & Evolution* (2019).

[96] Walter Jetz et al. "Essential biodiversity variables for mapping and monitoring species populations". In: *Nature ecology & evolution* (2019).

[97] Jin Li and Zilong Liu. "Multispectral Transforms Using Convolution Neural Networks for Remote Sensing Multispectral Image Compression". In: *Remote Sensing* (2019).

[98] Fabian Mentzer et al. "Practical full resolution learned lossless image compression". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

[99] Lars Mescheder et al. “Occupancy networks: Learning 3d reconstruction in function space”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

[100] Jeong Joon Park et al. “Deepsdf: Learning continuous signed distance functions for shape representation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

[101] Markus Reichstein et al. “Deep learning and process understanding for data-driven Earth system science”. In: *Nature* 566.7743 (2019), pp. 195–204.

[102] Bjorn Stevens et al. “DYAMOND: the DYnamics of the Atmospheric general circulation Modeled On Non-hydrostatic Domains”. In: *Progress in Earth and Planetary Science* (2019).

[103] Gencer Sumbul et al. “Bigearthnet: A large-scale benchmark archive for remote sensing image understanding”. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. 2019.

[104] James Townsend, Tom Bird, and David Barber. “Practical lossless compression with latent variables using bits back coding”. In: *arXiv preprint arXiv:1901.04866* (2019).

[105] 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”. In: *Agricultural Systems* (2019).

[106] Johannes Ballé et al. “Nonlinear transform coding”. In: *IEEE Journal of Selected Topics in Signal Processing* (2020).

[107] Damien Beillouin et al. “Impact of extreme weather conditions on European crop production in 2018”. In: *Philosophical Transactions of the Royal Society B* (2020).

[108] Ralph Dubayah et al. “The Global Ecosystem Dynamics Investigation: High-resolution laser ranging of the Earth’s forests and topography”. In: *Science of remote sensing* (2020).

[109] Adem Ertem et al. *Superpixel based compression of hyperspectral image with modified dictionary and sparse representation*. 2020.

[110] Hans Hersbach et al. “The ERA5 global reanalysis”. In: *Quarterly Journal of the Royal Meteorological Society* (2020).

[111] Jordan Hristov et al. “Analysis of climate change impacts on EU agriculture by 2050”. In: *Publications Office of the European Union, Luxembourg, Luxembourg* (2020).

[112] Milan Klöwer, Peter D Düben, and TN Palmer. “Number formats, error mitigation, and scope for 16-bit arithmetics in weather and climate modeling analyzed with a shallow water model”. In: *Journal of Advances in Modeling Earth Systems* (2020).

[113] J. Kuester, W. Gross, and W. Middelmann. “An approach to near-lossless hyperspectral data compression using deep autoencoder”. In: *Proceedings of SPIE - The International Society for Optical Engineering*. 2020.

[114] Yingchang Li et al. “Forest aboveground biomass estimation using Landsat 8 and Sentinel-1A data with machine learning algorithms”. In: *Scientific Reports* (2020).

[115] Elias Machairas and Nektarios 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”. In: *Electronics* (2020).

[116] Fabian Mentzer, Luc Van Gool, and Michael Tschannen. “Learning better lossless compression using lossy compression”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[117] David Minnen and Saurabh Singh. “Channel-wise autoregressive entropy models for learned image compression”. In: *IEEE International Conference on Image Processing*. 2020.

[118] Saurabh Singh et al. “End-to-end learning of compressible features”. In: *2020 IEEE International Conference on Image Processing (ICIP)*. 2020.

[119] Vincent Sitzmann et al. “Implicit neural representations with periodic activation functions”. In: *Advances in neural information processing systems* (2020).

[120] Matthew Tancik et al. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: *NeurIPS* (2020).

[121] Andrew Gordon Wilson. “The case for Bayesian deep learning”. In: *arXiv preprint arXiv:2001.10995* (2020).

[122] Vinicius Alves de Oliveira et al. "Reduced-Complexity End-to-End Variational Autoencoder for on Board Satellite Image Compression". In: *Remote Sensing* (2021).

[123] Peter Bauer, Bjorn Stevens, and Wilco Hazeleger. "A digital twin of Earth for the green transition". In: *Nature Climate Change* (2021).

[124] Thomas Bird et al. "3d scene compression through entropy penalized neural representation functions". In: *2021 Picture Coding Symposium (PCS)*. 2021.

[125] Rishi Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *arXiv preprint arXiv:2108.07258* (2021).

[126] Mathilde Caron et al. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.

[127] Hao Chen et al. "Nerv: Neural representations for videos". In: *Advances in Neural Information Processing Systems* (2021).

[128] Yilun Du et al. "Neural radiance flow for 4d view synthesis and video processing". In: *IEEE/CVF International Conference on Computer Vision*. 2021.

[129] Yann Dubois et al. "Lossy compression for lossless prediction". In: *Advances in Neural Information Processing Systems* (2021).

[130] Emilien Dupont et al. "Coin: Compression with implicit neural representations". In: *arXiv preprint arXiv:2103.03123* (2021).

[131] Patrick Esser, Robin Rombach, and Björn Ommer. "Taming Transformers for High-Resolution Image Synthesis". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[132] Nguyen Tien Hoang and Keiichiro Kanemoto. "Mapping the deforestation footprint of nations reveals growing threat to tropical forests". In: *Nature Ecology & Evolution* (2021).

[133] Stefan Kesselheim et al. "Juwels booster–a supercomputer for large-scale ai research". In: *High Performance Computing: ISC High Performance Digital 2021 International Workshops, Frankfurt am Main, Germany, June 24–July 2, 2021, Revised Selected Papers 36*. 2021.

[134] Milan Klöwer et al. "Compressing atmospheric data into its real information content". In: *Nature Computational Science* (2021).

[135] F. Kong et al. "End-to-end multispectral image compression framework based on adaptive multiscale feature extraction". In: *Journal of Electronic Imaging* (2021).

[136] F. Kong et al. "Spectral–spatial feature partitioned extraction based on CNN for multispectral image compression". In: *Remote Sensing* (2021).

[137] Jonathan Koomey and Eric Masanet. "Does not compute: Avoiding pitfalls assessing the Internet's energy and carbon impacts". In: *Joule* (2021).

[138] Simon TK Lang et al. "More accuracy with less precision". In: *Quarterly Journal of the Royal Meteorological Society* (2021).

[139] Ze Liu et al. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.

[140] Mirco Migliavacca et al. "The three major axes of terrestrial ecosystem function". In: *Nature* (2021).

[141] Ben Mildenhall et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* (2021).

[142] Peter Potapov et al. "Mapping global forest canopy height through integration of GEDI and Landsat data". In: *Remote Sensing of Environment* (2021).

[143] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. 2021.

[144] Andrés C. Rodríguez et al. "Mapping oil palm density at country scale: An active learning approach". In: *Remote Sensing of Environment* (2021).

[145] Maurizio Santoro and Oliver Cartus. "ESA Biomass Climate Change Initiative (Biomass CCI): Global datasets of forest above-ground biomass for the years 2010, 2017 and 2018, v2". In: *Centre for Environmental Data Analysis* (2021).

[146] Matthias Schramm et al. "The openEO API–Harmonising the Use of Earth Observation Cloud Services Using Virtual Data Cube Functionalities". In: *Remote Sensing* (2021).

[147] Koki Tsubota and Kiyoharu Aizawa. "Comprehensive Comparisons Of Uniform Quantizers For Deep Image Compression". In: *2021 IEEE International Conference on Image Processing (ICIP)*. 2021.

[148] Deepak Vasisht, Jayanth Shenoy, and Ranveer Chandra. "L2D2: low latency distributed downlink for LEO satellites". In: *ACM SIGCOMM Conference*. 2021.

[149] Neil Zeghidour et al. "Soundstream: An end-to-end neural audio codec". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2021).

[150] George Zhang et al. "Universal rate-distortion-perception representations for lossy compression". In: *Advances in Neural Information Processing Systems* (2021).

[151] Shihui Zhao et al. "Sparse Flow Adversarial Model For Robust Image Compression". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2021.

[152] Jan Ackmann et al. "Mixed-Precision for Linear Solvers in Global Geophysical Flows". In: *Journal of Advances in Modeling Earth Systems* (2022).

[153] Benjamin Brede et al. "Non-destructive estimation of individual tree biomass: Allometric models, terrestrial and UAV laser scanning". In: *Remote Sensing of Environment* (2022).

[154] T. Cao et al. "Spectral–Spatial Feature Completely Separated Extraction with Tensor CNN for Multispectral Image Compression". In: *Lecture Notes in Electrical Engineering*. 2022.

[155] Yezhen Cong et al. "SatMAE: Pre-training Transformers for Temporal and Multi-Spectral Satellite Imagery". In: *Advances in Neural Information Processing Systems*. 2022.

[156] Copernicus. *Observer: Copernicus Climate and Atmosphere Services provide 2021 climate insights at global and European level*. 2022. URL: %5Curl%7Bhttps://www.copernicus.eu/en/news/news/observer-copernicus-climate-and-atmosphere-services-provide-2021-climate-insights-global%7D.

[157] Z. Di et al. "Learned Compression Framework with Pyramidal Features and Quality Enhancement for SAR Images". In: *IEEE Geoscience and Remote Sensing Letters* (2022).

[158] Emilien Dupont et al. "COIN++: Neural Compression Across Modalities". In: *Transactions on Machine Learning Research* (2022).

[159] Kaiming He et al. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

[160] Xin Liang et al. "Sz3: A modular framework for composing prediction-based error-bounded lossy compressors". In: *IEEE Transactions on Big Data* (2022).

[161] Yoshitomo Matsubara et al. "Supervised compression for resource-constrained edge computing systems". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.

[162] Fabian Mentzer et al. "VCT: A Video Compression Transformer". In: *Advances in Neural Information Processing Systems* (2022).

[163] Pierre Potin et al. "Status and Evolution of the Sentinel-1 mission". In: *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2022, pp. 4707–4710.

[164] Yannick Strümpler et al. "Implicit neural representations for image compression". In: *European Conference on Computer Vision*. 2022.

[165] Xian Sun et al. "RingMo: A remote sensing foundation model with masked image modeling". In: *IEEE Transactions on Geoscience and Remote Sensing* (2022).

[166] Lucas Theis et al. "Lossy image compression with compressive autoencoders". In: *International conference on learning representations*. 2022.

[167] Robert Underwood et al. "Understanding the effects of modern compressors on the community earth science model". In: *2022 IEEE/ACM 8th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD)*. 2022.

[168] Di Wang et al. "Advancing plain vision transformer toward remote sensing foundation model". In: *IEEE Transactions on Geoscience and Remote Sensing* (2022).

[169] Qihan Xu et al. “Synthetic Aperture Radar Image Compression Based on a Variational Autoencoder”. In: *IEEE Geoscience and Remote Sensing Letters* (2022).

[170] M. Zhu, G. Li, and W. Zhang. “Research on UAV remote sensing multispectral image compression based on CNN”. In: *International Conference on Geology, Mapping and Remote Sensing*. 2022.

[171] Yinhao Zhu, Yang Yang, and Taco Cohen. “Transformer-based Transform Coding”. In: *International Conference on Learning Representations*. 2022.

[172] Eirikur Agustsson et al. “Multi-realism image compression with a conditional generator”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[173] Matt Allen et al. “Fewshot learning on global multimodal embeddings for earth observation tasks”. In: *arXiv preprint arXiv:2310.00119* (2023).

[174] Grégoire Delétang et al. “Language modeling is compression”. In: *arXiv preprint arXiv:2309.10668* (2023).

[175] C. Fu and B. Du. “Remote Sensing Image Compression Based on the Multiple Prior Information”. In: *Remote Sensing* (2023).

[176] Martin Hermann Paul Fuchs and Begüm Demir. “Hyspecnet-11k: A large-scale hyperspectral dataset for benchmarking learning-based hyperspectral image compression methods”. In: *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. 2023.

[177] J. Gao et al. “Mixed Entropy Model Enhanced Residual Attention Network for Remote Sensing Image Compression”. In: *Neural Processing Letters* (2023).

[178] Carlos Gomes, Roberto Azevedo, and Christopher Schroers. “Video Compression With Entropy-Constrained Neural Representations”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[179] Danfeng Hong et al. “SpectralGPT: Spectral foundation model”. In: *arXiv preprint arXiv:2311.07113* (2023).

[180] Langwen Huang and Torsten Hoefler. “Compressing multidimensional weather and climate data into neural networks”. In: *The Eleventh International Conference on Learning Representations*. 2023.

[181] Johannes Jakubik et al. “Foundation models for generalist geospatial artificial intelligence”. In: *arXiv preprint arXiv:2310.18660* (2023).

[182] Nikolai Kalischek et al. “Cocoa plantations are associated with deforestation in Côte d’Ivoire and Ghana”. In: *Nature Food* (2023).

[183] Nico Lang et al. “A high-resolution canopy height model of the Earth”. In: *Nature Ecology & Evolution* (2023).

[184] Christian Lessig et al. “AtmoRep: A stochastic model of atmosphere dynamics using large scale representation learning”. In: *arXiv preprint arXiv:2308.13280* (2023).

[185] J. Li, Y. Ye, and B. Liu. “An Intelligent Image Compression Method Based on Generative Adversarial Networks for Satellites”. In: *International Symposium on Computer Engineering and Intelligent Communications*. 2023.

[186] Wenwen Li et al. “Assessment of IBM and NASA’s geospatial foundation model in flood inundation mapping”. In: *arXiv preprint arXiv:2309.14500* (2023).

[187] Xin Li et al. “Remote Sensing Image Compression Method Based on Implicit Neural Representation”. In: *2023 12th International Conference on Computing and Pattern Recognition* (2023).

[188] Sebastià Mijares i Verdú et al. “A Scalable Reduced-Complexity Compression of Hyperspectral Remote Sensing Images Using Deep Learning”. In: *Remote Sensing* (18 2023).

[189] Griffin Mooers et al. “Comparing storm resolving models and climates via unsupervised machine learning”. In: *Scientific Reports* (2023).

[190] Cécile Renier et al. “Transparency, traceability and deforestation in the Ivorian cocoa supply chain”. In: *Environmental Research Letters* (2023).

[191] Michael J Smith, Luke Fleming, and James Geach. “EarthPT: a foundation model for Earth Observation”. In: *NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning*. 2023.

[192] Yi Wang et al. “SSL4EO-S12: A large-scale multimodal, multitemporal dataset for self-supervised learning in Earth observation [Software and Data Sets]”. In: *IEEE Geoscience and Remote Sensing Magazine* (2023).

[193] S. Xiang and Q. Liang. "Remote sensing image compression with long-range convolution and improved non-local attention model". In: *Signal Processing* (2023).

[194] S. Xiang, Q. Liang, and L. Fang. "Discrete Wavelet Transform-Based Gaussian Mixture Model for Remote Sensing Image Compression". In: *IEEE Transactions on Geoscience and Remote Sensing* (2023).

[195] Yibo Yang, Stephan Mandt, Lucas Theis, et al. "An introduction to neural data compression". In: *Foundations and Trends® in Computer Graphics and Vision* (2023).

[196] Lijun Yu et al. "Language Model Beats Diffusion–Tokenizer is Key to Visual Generation". In: *arXiv preprint arXiv:2310.05737* (2023).

[197] D. Anuradha et al. "Efficient Compression for Remote Sensing: Multispectral Transform and Deep Recurrent Neural Networks for Lossless Hyper-Spectral Imagine". In: *International Journal of Advanced Computer Science and Applications* (2024).

[198] Cristian Bodnar et al. *Aurora: A Foundation Model of the Atmosphere*. Tech. rep. 2024.

[199] *Certain data included herein are derived from Clarivate Web of Science.* 2024. URL: https://webofscience.com.

[200] *Citation Report graphic is derived from Clarivate Web of Science.* 2024. URL: https://webofscience.com.

[201] J. Clauson et al. *Earth Observing Sensing Satellites Online Compendium: U.S. Geological Survey digital data*. 2024.

[202] Copernicus. *Copernicus European Flood Awareness System*. 2024.

[203] J. Deng and L. Huang. "Synthetic Aperture Radar Image Compression Based on Low-Frequency Rejection and Quality Map Guidance". In: *Remote Sensing* (2024).

[204] Kuntai Du et al. "Earth+: on-board satellite imagery compression leveraging historical earth observations". In: *arXiv preprint arXiv:2403.11434* (2024).

[205] Alireza Furtuanpey, Philipp Raith, and Schahram Dustdar. "FrankenSplit: Efficient Neural Feature Compression with Shallow Variational Bottleneck Injection for Mobile Edge Computing". In: *IEEE Transactions on Mobile Computing* (2024).

[206] Alireza Furutanpey et al. "FOOL: Addressing the Downlink Bottleneck in Satellite Computing with Neural Feature Compression". In: *arXiv preprint arXiv:2403.16677* (2024).

[207] Alireza Furutanpey et al. "FOOL: Addressing the Downlink Bottleneck in Satellite Computing with Neural Feature Compression". In: *arXiv preprint arXiv:2403.16677* (2024).

[208] Carlos Gomes and Thomas Brunschwiler. "Compressed Multi-task embeddings for Data-Efficient Downstream training and inference in Earth Observation". In: *arXiv preprint arXiv:2403.17886* (2024).

[209] Xin Guo et al. "Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.

[210] Fan Liu et al. "RemoteCLIP: A vision language foundation model for remote sensing". In: *IEEE Transactions on Geoscience and Remote Sensing* (2024).

[211] Jamie Tolan et al. "Very high resolution canopy height maps from RGB imagery using self-supervised vision transformer and convolutional decoder trained on aerial lidar". In: *Remote Sensing of Environment* (2024).

[212] R. Wilkinson et al. "Environmental impacts of earth observation data in the constellation and cloud computing era". In: *Science of The Total Environment* (2024).

[213] S. Xiang and Q. Liang. "Remote Sensing Image Compression Based on High-Frequency and Low-Frequency Components". In: *IEEE Transactions on Geoscience and Remote Sensing* (2024).

[214] W. Ye et al. "GFSCompNet: remote sensing image compression network based on global feature-assisted segmentation". In: *Multimedia Tools and Applications* (2024).

[215] L. Zhang et al. "Compressing Hyperspectral Images Into Multilayer Perceptrons Using Fast-Time Hyperspectral Neural Radiance Fields". In: *IEEE Geoscience and Remote Sensing Letters* (2024).

[216] Guy E Blelloch et al. "Introduction to data compression". In: ().

[217] Cloud-Native Geospatial Foundation. *Survey for current practices storing embeddings in GeoParquet*.