Toward Efficient Satellite Computing Through Adaptive Compression

Chen Yang, Qibo Sun, Qiyang Zhang, Hao Lu, Claudio A. Ardagna, *Senior Member, IEEE*, Shangguang Wang, *Senior Member, IEEE*, and Mengwei Xu

***Abstract*—The rapid development of Low Earth Orbit (LEO) satellite constellations offers significant potential for in-orbit services, particularly in mitigating the impact of sudden natural disasters. However, the massive data collected by these satellites are often large and severely constrained by limited transmission capabilities when sending data to the ground. Satellite computing, which utilizes onboard computational capacity to process data before transmission, presents a promising solution to alleviate the downlink burden. Nonetheless, this paradigm introduces another bottleneck: limited onboard computing capacity, resulting in slow in-orbit processing and poor results. Current satellite computing systems struggle to efficiently address both data transmission and computing bottlenecks, particularly for urgent disaster services that demand accurate and timely results. Thus, we introduce an efficient satellite computing system designed to jointly mitigate these bottlenecks, thereby providing better service. The core idea is to utilize onboard computing capacity for swift in-orbit annotation of image regions, enabling adaptive compression and download based on annotation confidence and perceived downlink availability. Once the data is downloaded, image restoration and re-inference are performed on the ground to enhance accuracy. Compared to satellite-only inference, our system demonstrates an average improvement in inference accuracy of 3.8%. Furthermore, compared to ground-only inference, with only a 2.8% accuracy loss, our system achieves a 38.4% reduction in response time and saves 71.6% of downlink volume on average.**

***Index Terms*—Satellite computing, in-orbit service, inference, adaptive compression.**

I. Introduction

In RECENT years, satellites have transitioned from large and expensive to small and economical. Coupled with the development of Commercial Off-The-Shelf (COTS) hardware, major cloud service providers (such as Google and AWS) and satellite companies (such as OrbitsEdge) have proposed the concept of "Space Infrastructure as a Service" [1]. This initiative aims to equip large-scale small satellite constellations with intelligent computing capabilities to efficiently process space data. At the core of this initiative is the satellite computing paradigm, which can extract vast amounts of information collected from space, providing valuable analysis for applications such as global climate monitoring and disaster management, thereby enhancing the quality and speed of satellite services. For instance, during disasters like Australia's Black Summer bushfires and flooding in Bangladesh, intelligent processing and analysis of satellite imagery have demonstrated the capability to accurately locate disasters. This potential can mitigate impacts over extensive areas, covering millions of square kilometers, and significantly reduce economic losses, potentially saving billions of dollars.

Despite the tremendous potential and value of satellite intelligent computing, obtaining accurate and timely analysis results remains a major challenge. This issue significantly hinders the provision of high-quality satellite services and restricts the practical deployment of satellite computing systems. The response time from dispatching commands to satellites to obtaining detailed image analysis on the ground typically exceeds 8 hours (even a day) [5]. Additionally, performing processes directly in orbit is frequently deemed impractical due to the insufficient accuracy of the obtained onboard results. Both onboard inaccurate results and extended response time are deemed unacceptable, particularly in rapidly evolving disaster scenarios. Therefore, there is a necessity to design an efficient satellite computing system capable of enhancing accuracy within acceptable latency.

Enhancing the availability of satellite computing systems primarily faces two inherent bottlenecks: limited onboard computing capability and insufficient satellite-ground transmission volume. To alleviate these bottlenecks, some works [6], [7], [8] explored the deployment of lightweight models on satellites by using COTS hardware for inference, aiming to reduce in-orbit processing latency. However, for computing-intensive computing tasks, the limited onboard computing capabilities of satellites hinder their ability to manage all observation tasks and ensure the accuracy of in-orbit processing. On the other hand, some efforts [9], [10], [11], [12] investigated the satellite-ground collaborative computing paradigm, by capitalizing on the strengths of both in-orbit computing and ground-based computing to decrease latency and enhance accuracy. Nevertheless, constrained by limited transmission capacity and brief contact satellite-ground downlink, not all observations can be transmitted to the ground. Achieving better inference accuracy

Received 2 July 2024; revised 5 September 2024; accepted 21 September 2024. Date of publication 30 September 2024; date of current version 30 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62425203, Grant 62372061, and Grant 62032003, and in part by Natural Science Foundation of Chongqing, China under Grant CSTB2023NSCQ-LMX0020. *(Corresponding authors: Qibo Sun; Qiyang Zhang.)*

Chen Yang, Qibo Sun, Hao Lu, Shangguang Wang, and Mengwei Xu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yangc@bupt.edu.cn; qbsun@bupt.edu.cn; triplelontheroad@bupt.edu.cn; sgwang@bupt.edu.cn; mwx@bupt.edu.cn).

Qiyang Zhang is with the Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, School of Computer Science, Peking University, Beijing 100871, China (e-mail: qiyangzhang@pku.edu.cn).

Claudio A. Ardagna is with the Dipartimento di Informatica, Università degli Studi di Milano, 20122 Milano, Italy (e-mail: claudio.ardagna@unimi.it).

Digital Object Identifier 10.1109/TSC.2024.3470341

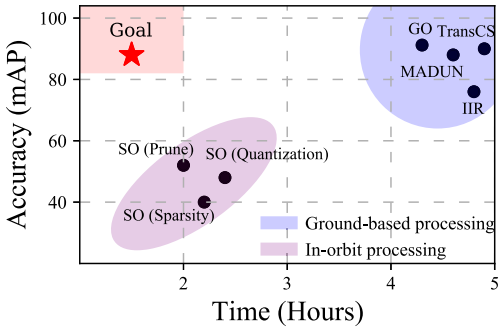


Fig. 1. The accuracy-latency tradeoff between two deployment approaches of satellite computing systems. Accuracy is determined by the average mAP validated on NWPU (§4), while efficiency is measured by the time consumption. SO (Prune, Sparsity, Quantization): Deploy a (pruning, sparse, quantized) model for in-orbit inference on satellite. GO: Download raw data to the ground for inference. MADUN [2], IIR [3], and TransCS [4]: Download compressed data, then perform restoration and inference on the ground.

performance becomes challenging. Therefore, our work jointly considers the above two bottlenecks, and our goal strives to maximize the system accuracy while minimizing the latency, as shown in Fig. 1.

To this end, this paper aims to design an efficient satellite computing system. The core idea of our system revolves around transmitting all images to the ground and leveraging the abundant computing resources available on the ground to enhance system accuracy. To minimize the downlink volume and transmission time between satellite and ground, we initially process all captured images using compression techniques. Subsequently, based on the downloaded compressed images, we perform image restoration on the ground to improve system accuracy. However, the design of the satellite computing system encounters two key challenges that have not been thoroughly addressed in existing literature. First, due to the saturated downlink, efficient transmission requires sending more useful data that enhances accuracy with reduced volume. However, traditional existing strategies to reduce data volume cannot be directly applied to this issue. For instance, applying uniform compression to all images significantly degrades system accuracy. Thus, how to balance compression and transmission volume by determining the optimal image compression ratio is key to optimizing the capability of transmission. Second, to improve the accuracy of the satellite computing system, downloaded images require restoring more information with less time overhead. However, restoring the resolution of all compressed images introduces substantial time overhead. Specifically, highly compressed images, having lost significant information, may not yield beneficial results when inputted into powerful models. Thus, balancing the time overhead with the number of restored images to determine the optimal restoration strategy remains challenging.

To tackle the above challenges, we conducted preliminary experiments and found some crucial observations that motivated the design of this satellite computing system. First, onboard computing resources are better suited for identifying the importance of regions within images rather than analyzing the entirety of the images. We hence design an adaptive compression strategy based on the annotated importance of each image region in orbit. Considering the constraint of downlink volume, we prioritize allocating lower compression ratios to more important regions to preserve crucial information in images, thereby enhancing the efficiency of downlink utilization. Second, restoring the annotated important regions of compressed images, as opposed to all regions, results in greater accuracy improvements with less time overhead. Thus, we design a patch-padding restoration method based on in-orbit annotations of image regions. By exclusively restoring the image regions annotated in orbit, this approach minimizes time overhead while efficiently improving system accuracy.

In this paper, we propose an efficient satellite computing system `AdaEO`. `AdaEO` not only enhances accuracy but also contributes to reducing task response time. This system conducts in-orbit annotation and identification of crucial image regions, facilitating the coarse-grained localization of critical regions and enabling rapid low-precision responses. Subsequently, `AdaEO` employs adaptive compression for these regions based on their measured importance, effectively reducing the overall transmission data volume. Additionally, `AdaEO` leverages a patch-padding method based on downloaded images, resulting in an overall improvement in system accuracy. We implemented and evaluated `AdaEO` on three widely used satellite image processing task datasets. The results show that, compared to the satellite-only (SO) inference method, `AdaEO` tremendously reduces the response time 38.4% on average and improves the model accuracy 3.8% on average. Moreover, compared to the ground-only (GO) inference method, `AdaEO` achieves substantial 71.6% on average downlink usage savings while maintaining the model performance.

The key contributions of this paper are:

- Aiming to maximize accuracy within acceptable latency, we propose an efficient satellite computing system called `AdaEO`, with a joint consideration between computing and transmission bottlenecks.
- We first conduct extensive preliminary measurements, summarizing the key observations and derived that different regions of each image contribute variably to the system's accuracy. This insight motivated the design of `AdaEO`.
- We further design a confidence-adaptive image compression method to optimize the transmission volume and time. Additionally, we design a patch-padding image restoration method to enhance the accuracy of `AdaEO`.
- We implement and evaluate the performance of `AdaEO` and demonstrate that `AdaEO` has advantages in accuracy and response time over baselines.

II. Background and Motivation

A. Satellite Computing

Over the past decade, advancements in satellite technology and COTS hardware have driven a rapid increase in the deployment of small LEO CubeSats. By equipping these satellites with computational resources, they can progressively offer strong service guarantees for various terrestrial scenarios, providing in-orbit services to the public. Specifically, image processing, as one of the fundamental computational tasks for satellites, requires downloading raw observational data to ground-based data centers for processing under the traditional bent-pipe

architecture. However, due to inherent bottlenecks in satellite-ground connection windows and downlink bandwidth, this traditional paradigm struggles to support low-latency applications such as disaster monitoring and oil spill detection. Additionally, although these satellites can collect images covering over 350 million square kilometers daily, much of this data is difficult to download to the ground in a short time, making it challenging to utilize effectively. Satellite computing proposes the use of COTS hardware deployed in orbit to perform preliminary processing on raw data before transmission to the ground, thereby alleviating data downlink pressure. As satellite technology continues to evolve and COTS hardware capabilities improve, this new paradigm of satellite computing holds the potential to provide more effective services for real-time and high-accuracy applications, particularly in disaster monitoring.

B. Limitations of Deploying Efficient Satellite Computing Systems

As illustrated above, while satellite computing holds immense potential to enhance in-orbit service efficiency, the surge in data volume combined with the limited performance of COTS hardware still constrains its practical utility. Therefore, deploying an efficient satellite computing system is crucial. Specifically, there are two unique challenges:

Onboard computational bottleneck: Due to the limitations of current satellite design and COTS hardware, in-orbit computing systems typically have restricted computational capabilities. The large size and huge volume of data collected by satellites pose a significant challenge for efficient processing by these systems. Currently, in-orbit computing systems struggle to perform inference and analysis on all data promptly and accurately, making it difficult to meet service demands. Specifically, insufficient accuracy and long processing times render in-orbit computing results significantly less useful.

Satellite-to-ground downlink transmission bottleneck: Despite current downlink bandwidths reaching 10 s of Mbps, the brief connection windows, usually lasting only a few minutes, limit the amount of data that can be transmitted to ground-based data centers. This prevents the utilization of powerful ground-based computing resources to enhance inference performance. Even saturated downlink capacities cannot support the download of all collected data (10 s to 100 s of TB/day), and the extent to which inference performance can be improved depends significantly on the total amount of downloadable data.

C. Existing Solutions and Motivation

The performance of current satellite computing is constrained by the aforementioned unique challenges. To unlock the potential value of the vast amount of data collected by these satellites, we have focused on improving the intelligent analysis and processing capabilities of in-orbit satellites. Specifically, we aim to enhance the efficiency of deployed satellite computing systems, thereby offering better services to the public. Based on efforts from various perspectives, we group the related work into three main categories:

Satellite computing with AI capability: With the improvement of computational capabilities in satellite COTS hardware, efforts have been made to integrate AI capabilities into satellites to provide better services. Initiatives such as Tiansuan Constellation [13], [14] and OEC [15] have proposed satellite computing to support in-orbit processing of collected data. To enhance data analysis and processing efficiency, several works [13], [15], [16], [17] have attempted to deploy inference models on satellites. While these efforts leverage limited in-orbit computational capabilities for intelligent computing, the results often remain impractical. For instance, some work [18] has analyzed the energy consumption and latency of various in-orbit inference tasks. Despite increasing the utilization of in-orbit computational resources, the inference accuracy remains insufficient. Other studies have sought to optimize computational system performance through satellite-ground collaborative offloading [18] and multi-satellite collaborative model early-exit mechanisms [19]. However, limited data transmission capacity constrains the improvement of inference performance. Additionally, some efforts [12], [20] have attempted to reduce data transmission volume by filtering images based on target coordinates. Yet, accurately locating the target position is not always feasible. Even when ground coordinates are identified, converting them to usable in-orbit coordinates involves complex transformations. Therefore, deploying an efficient satellite intelligent computing system presents significant challenges in practical applications.

Efficient inference systems for resource-constrained devices: With the rapid development of mobile/edge devices, the deployment of inference systems on these devices is evolving towards greater resource efficiency [21]. Performance optimization on devices has garnered extensive attention from both academia and industry [22], [23], [24]. Various research efforts have focused on reducing the overhead of DL systems on mobile devices, employing techniques such as offloading, model quantization, model sparsity, and model pruning [25], [26], [27], [28], [29], [30]. These approaches strive to balance inference latency and model accuracy. Our work draws inspiration from these efforts but emphasizes the rapid provision of in-orbit inference results immediately after a disaster. Additionally, we aim to maximize inference accuracy over time, ensuring both prompt and accurate responses in critical situations.

Image compression and resolution restoration: Image compression effectively transmits images in scenarios with limited storage space and bandwidth, while resolution restoration compensates for lost image details, enhancing overall quality. Some efforts have explored compressed sensing [31], [32] by designing sparse sampling matrices and reconstruction methods, significantly reducing required bandwidth compared to traditional methods. Other works focus on these processes separately. For example, some research discards redundant information to ensure higher compression efficiency [33], while others predict and estimate high-resolution details from low-resolution images [34]. Additionally, generative techniques such as VAE, GAN, and Stable Diffusion [35], [36] have been introduced to generate images through models [35], [36]. Unlike existing accuracy-prioritized compression methods such as PNG and JPEG [37], [38], as well as traditional compressed sensing algorithms. We strive to design an end-to-end satellite computing system that provides adaptive compression and corresponding restoration of downloaded images under constrained downlinks, achieving efficient transmission and accurate compensation.

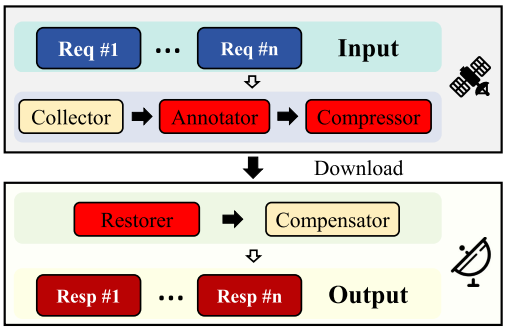


Fig. 2. The overall workflow of AdaEO.

In this paper, we explore a solution to jointly address the aforementioned two bottlenecks, aiming to equip satellites with efficient intelligent processing capabilities. Despite the limited on-board computational power, it is adequate to support in-orbit data analysis and preliminary processing. Specifically, performing initial annotations on important regions of images is more suitable than directly utilizing on-board inference results. Additionally, adaptive compression based on the importance of image regions lays the groundwork for more efficient utilization of downlink volume, thereby contributing to the enhancement of inference accuracy. Therefore, designing and deploying an efficient satellite computing system that enables satellites to perform practical intelligent processing is both critical and meaningful.

III. AdaEO System Design

This work proposes an efficient satellite computing system AdaEO, aimed to improve inference performance including accuracy and response time. The core idea involves first identifying critical regions based on the constrained computational capabilities of satellites. Following this initial step, adaptive image compression is implemented to alleviate the transmission burden of data downlink. Finally, ground-based resources are utilized to further enhance the inference accuracy of the system.

A. System Overview

Fig. 2 shows the detailed illustration of our AdaEO pipeline. Overall, AdaEO adopts a satellite-ground collaborative framework. Initially, AdaEO conducts the preliminary processing of the collected data onboard. Subsequently, AdaEO utilizes the satellite-ground downlink for transmitting the processed data to the ground station, thereby enhancing the entire system performance.

In AdaEO, in-orbit system includes three primary components: the collector, annotator, and compressor. The collector receives image capture task commands and orchestrates the image collection process. The annotator carries out preliminary annotations of image regions in orbit by utilizing deployed lightweight models. The compressor adaptively compresses data during download, considering the current link status. Note that, our design primarily emphasizes the latter two components. Specifically, we focus on annotating critical image regions in orbit and adaptively compressing the downloaded data. This compression is guided by both the link status and the annotations made on the image regions. Our approach is designed to alleviate the two issues of inefficient inference: the limited computational capacity of satellites that makes it challenging to perform high-precision inference; and the constraints in the transmission capacity of satellite-ground links that make it challenging to download all data.

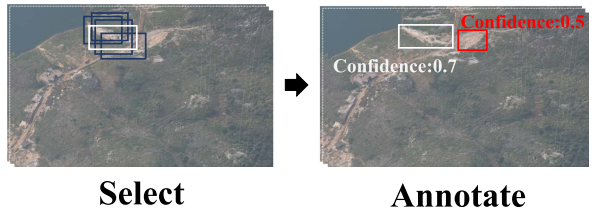


Fig. 3. The diagram of swift in-orbit annotation.

System in the ground station includes two crucial components: the restorer and the compensator. The restorer is responsible for the resolution recovery of the compressed images received from satellites, and obtaining an image dataset of restored images. The compensator utilizes this restored image dataset for re-tuning, enhancing the system's inference accuracy. Our main focus lies in the design of the restorer component. Specifically, we concentrate on how to efficiently restore the compressed images once they have been downloaded, and further leverage the powerful models on the ground to assist in-orbit inference performance.

In the process of in-orbit region annotation, confidence thresholds based on the IoG detected by the onboard model are established, and these thresholds are utilized to evaluate in-orbit computing. This confidence threshold indicates the probability of accuracy for the annotated regions, which lies in the range [0, 1] [39]. A higher threshold correlates with more reliable detection results, suggesting a greater probability of accuracy in the annotated regions.

B. Swift In-Orbit Annotation

How to efficiently process large captured images using ML techniques is a crucial issue currently. A large amount of images typically contain thousands of megapixels, which makes the satellite-ground downlink unaffordable. Moreover, in-orbit inference faces computational and energy constraints. To alleviate the above constraints, we argue that AdaEO should prioritize the swift annotation of onboard images, utilizing the constrained resources available on the satellite. This process facilitates the identification of vital regions within the images, and serves as a basis to alleviate the burden on the satellite-ground downlink.

To fully capitalize on the maximum benefits of the entire system, obtaining swift inference results directly in orbit is both necessary and advantageous. For instance, in disaster scenarios, rapid escalation in the number of casualties and the expansion of affected regions before effective containment [40], swift and coarse-grained annotation of disaster zones can be more beneficial than continually seeking to enhance the precision of disaster location inference.

Fig. 3 depicts the diagram of our swift in-orbit annotation method, with the goal of rapidly detecting and delineating the rough disaster-affected regions. The design is outlined as follows: i) We deploy a lightweight neural network (pruned DETR)

on the satellite, which performs object detection directly on the images captured by the satellite without training process; ii) We identify bounding boxes corresponding to targets within each image, along with their associated confidence thresholds.

We cannot employ existing classical methods such as traditional IoU metric [41] directly as our criteria, as IoU-based selection of bounding boxes inevitably result in the omission of part of important regions. IoU metric actually calculates the intersection over the union between candidate bounding boxes and the ground truth. Although a higher IoU value suggests a smaller superfluous area relative to the ground truth, it does not guarantee comprehensive coverage of the ground truth's maximum extent. In the context of low-precision inference, candidates chosen based on high IoU values may tend to minimize overlap with redundant areas, inadvertently increasing the likelihood of omitting crucial regions. Such a scenario is counterproductive to our objective of annotating disaster areas in a coarse-grained manner.

Therefore, to ensure more inclusive coverage of the ground truth, we define IoG as the standard for selecting candidate bounding boxes in the following.

$$IoG = \frac{A_O}{A_G} \tag{1}$$

where A_O represents the area of intersection between the candidate bounding box and the ground truth, and A_G denotes the area of the ground truth. We choose the bounding box with the highest IoG for rapid in-orbit annotation. Note that, the accuracy of these annotated boxes is not of primary concern. Their purpose is to differentiate image regions of varying importance and provide a reference for disaster relief efforts.

C. Confidence-Adaptive Image Compression

Upon conducting swift annotations of image regions, in-orbit inference results are inadequate as the final output of the system. For instance, in the context of post-disaster relief operations, low-precision inference results might lead to substantial resource wastage or even misguide rescue efforts. Hence, there is an urgent need to enhance in-orbit inference accuracy. On the other hand, directly downloading all images to the ground for analysis to enhance inference accuracy is infeasible due to current limitations in downlink bandwidth.

To address these issues our approach utilizes compression techniques to prioritize and transmit vital image regions based on perceived transmission capacity and improves the inference accuracy using ground-based models. Specifically, we denote the set of image indexes as $\mathcal{I} = \{1, \ldots, i, \ldots, I\}$, and $\mathcal{D} = \{D_1, \ldots, D_i, \ldots, D_I\}, i \in \mathcal{I}$; where $\mathcal{D}$ represents all the images that need to be downloaded, and D_i represents each specific image. Similarly, we denote the set of compression ratio indexes as $\mathcal{K} = 1, \ldots, k, \ldots, K$, and we define $\mathcal{C} = \{C_1, \ldots, C_k, \ldots, C_K\}, k \in \mathcal{K}$, where $\mathcal{C}$ represents the compression factors, and C_k represents the compression value applied to a specific image.

These images will be input into the model for tuning, to improve system performance by minimizing the loss function $F(\mathcal{D})$, and f_i represents the loss of information for each image after compression. The formulate problem is denoted by:

$$\min F(\mathcal{D}) = \sum_{i=1}^{I} f_i(D_i) \tag{2}$$

Considering the compression ratio of each image $\mathcal{C}[\mathcal{D}]$ deeply affects the system performance, we further refine the problem as follows:

$$\mathbf{P1}: \min\ F(\mathcal{C}[\mathcal{D}]) = f_1(\mathcal{C}[D_1]), \ldots, f_i(\mathcal{C}[D_i]), \ldots, f_I(\mathcal{C}[D_I]), \tag{3a}$$

$$\text{s.t.}\quad \xi(\mathcal{C}[D_1] + \cdots \mathcal{C}[D_i] + \cdots + \mathcal{C}[D_I]) \leq tB, \forall i \in \mathcal{I}. \tag{3b}$$

where t is the duration of connectivity between the satellite and the ground station, B denotes the downlink bandwidth, and ξ represents the total storage size occupied by all processed images.

However, addressing **P1** presents significant challenges for two primary reasons. First, designing a compression method that minimally impacts accuracy is challenging. This is because compressing the original image $\mathcal{C}[\mathcal{D}]$ reduces data transmission. However, it inevitably leads to increased information loss at higher compression ratios C_k, thereby degrading system performance. Second, establishing an exact analytical relationship between the loss function f and each compressed image $\mathcal{C}[D_i]$ is generally unattainable. The reason is that when the same compression ratio is applied, each image has varying effects on the system's accuracy.

To tackle **P1**, we investigate the interaction between different ratios of image compression and their impact on the system. Indeed, the importance of each image in improving system accuracy is not uniform [12]. This insight led us to a deeper analysis of different regions within each image. Based on the previous annotation of different regions within each image with different confidence thresholds, we further investigate whether these distinct regions contribute to varying enhancements in system accuracy. Therefore, we conducted preliminary experiments by applying varying compression ratios to these regions and found an interesting observation for algorithm design.

• *Observation 1: The importance of annotated regions in images for system accuracy gain is higher compared to other regions:* Fig. 4 shows the accuracy and compressed image size when employing varying compression ratios within a designated image region. Specifically, when opting for a lower compression ratio (2× compression) in the annotated image region and a higher compression ratio (8× compression) in the unannotated image region, we observe an improved accuracy performance by 7.3%, while saving 39.5% image size. It is concluded that different regions within the image contribute to varying gains in system accuracy. This enables us to adopt a lower compression ratio in regions of higher importance, preserving more of the original information, while employing a higher compression ratio in other areas to economize on the required data transmission size.

Based on the above observation, we defined a set of region indexes in each image D_i as $\mathcal{J} = \{1, \ldots, j, \ldots, J\}$, and then we divided the images into regions and defined $\mathcal{C} = \{C_1, \ldots, C_k, \ldots, C_K\}, k \in \mathcal{K}$; where R_{ij} represents the j-th region of

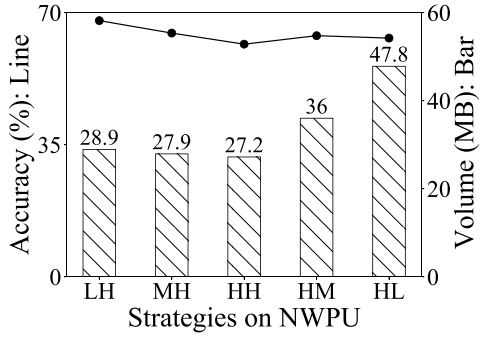


Fig. 4. The system's performance with different compression ratios on various annotated regions using DETR model. LH: Allocate compression ratio of 2 to regions with high confidence, and allocate 8 to regions with low confidence. MH: Allocate 4 to high, and 8 to low. HH: Allocate 8 to all regions. HM: Allocate 4 to high, and 2 to low. HL: Allocate 8 to high, and 2 to low.

the i-th image. Then, the set of original images $\mathcal{D}$ can be redefined as $\mathcal{R} = \{R_{11}, \cdots, R_{1j}, \cdots, R_{i1}, \cdots, R_{ij}\}, i \in \mathcal{I}, j \in \mathcal{J}$. Moreover, based on the confidence values from swift annotated regions, we sort the elements within each $\mathcal{R}$ in descending order, denoted as $\mathcal{R}_{\mathrm{S}} = \{R_{\mathrm{S}11}, \cdots, R_{\mathrm{S}1j}, \cdots, R_{\mathrm{S}i1}, \cdots, R_{\mathrm{S}ij}\}, i \in \mathcal{I}, j \in \mathcal{J}$. We also sorted the elements within $\mathcal{C}$ in descending order, defined as $\mathcal{C}_{\mathrm{S}} = \{C_{\mathrm{S}1}, C_{\mathrm{S}2}, \cdots, C_{\mathrm{S}k}\}$. Moreover, we defined $T = \mathcal{C}_{\mathrm{S}}\mathcal{R}_{\mathrm{S}}x$, where T_{ij} represents that we use $C_{\mathrm{S}k} \in \mathcal{C}_{\mathrm{S}}$ to compress the j-th region of the i-th image, i.e., $T_{ij} = C_{\mathrm{S}1}R_{\mathrm{S}ij}x_{i1} + C_{\mathrm{S}2}R_{\mathrm{S}ij}x_{i2} + \cdots + C_{\mathrm{S}k}R_{\mathrm{S}ij}x_{ij}$. Then, we endeavored to refine **P1** into **P2**:

$$\mathbf{P2}: \min_{x_{ij}} \{F(T_{ij})\}, \tag{4a}$$

$$s.t. \quad \xi\left(\sum_{i=1}^{I}\sum_{j=1}^{J} T_{ij}\right) \leq tB, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \tag{4b}$$

$$x_{ij} = \{0, 1\}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \tag{4c}$$

$$\sum_{j=1}^{J} x_{ij} = 1, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}. \tag{4d}$$

In **P2**, to minimize the decline in system accuracy by adjusting distinct compression ratios for individual regions of the image: Constraint (4b) represents the total bandwidth limitation; Constraint (4c) mandates that each region will be compressed by a specific compression ratio, Constraint (4d) mandates that each region can only have one matching compression ratio. The key consideration is how to ensure that the transmitted compressed data $\xi(\sum_{i=1}^{n}\sum_{j=1}^{m} T_{ij})$ does not exceed the data capacity available for transmission. Note that, the $T = \mathcal{C}_{\mathrm{S}}\mathcal{R}_{\mathrm{S}}x$ is controlled by the binary vector x_{ij}, while $\mathcal{C}_{\mathrm{S}}$ is a set and each element is taken from a continuous interval. Therefore, **P2** is a classical NP-hard problem [42]. Specifically, considering the satellite hardware's efficient support for compression, in most cases, it is more friendly to set the compression ratio as powers of 2. Thus, we re-define the previous $\mathcal{K} = \{2^K, \ldots, 2^k, \ldots, 2^1\}$ and $\mathcal{C}_{\mathrm{S}} = \{C_{\mathrm{S}1}, C_{\mathrm{S}2}, \ldots, C_{\mathrm{S}k}\}, k \in \mathcal{K}$.

Furthermore, there are some interesting observations from the previous measurements when exploring the impact of varying compression ratios on system accuracy: 1) When the compression ratio of images exceeds 16, the preservation of vital information deteriorates, leading to a notable decline in system accuracy. 2) As confidence progressively decreases, there are certain thresholds where system accuracy decreases significantly. To enhance the system's accuracy by ensuring that adaptive image compression retains important information, we employ a strategy where we partition the confidence into intervals based on thresholds that have a significant impact on system accuracy. Within these intervals, we then assign appropriate compression ratios. Moreover, we define $\mathcal{K} = \{2^4, 2^3, 2^2, 2^1\}$, and $\mathcal{C}_{\mathrm{S}} = \{C_{\mathrm{S}1}, C_{\mathrm{S}2}, \ldots, C_{\mathrm{S}k}\}, k \in \mathcal{K}$, then we create a list L_T based on $\mathcal{V}_T$ to record different threshold values for dividing $\mathcal{R}_{\mathrm{S}}$.

Algorithm 1 shows an adaptive approach to adjust the compression ratio of image regions with varying confidence, aiming to enhance the data transmission efficiency between satellites

Algorithm 1: Confidence-Adaptive Image Compression Algorithm.

Input : Initial list of threshold values to divide confidence: L_T, Initial list of compression ratio values: $\mathcal{C}_{\mathrm{S}}$, Initial allocation of compression ratio on threshold interval: A_I, Total transmission constraint: tB_T

Output: Final allocation of compression ratio on threshold interval: A_F.

Initialization:
  $\mathrm{Dict_T}$; // Create a threshold_states dict
  for $k \leftarrow 0$ **to** *Len*($\mathcal{C}_{\mathrm{S}}$) **do**
    for $i \leftarrow 0$ **to** *Len*(L_T) **do**
      $\mathrm{Dict_T[k][i]} = \{$"threshold" : L_T[i], "ratio" : $\mathcal{C}_{\mathrm{S}}$[i]$\}$;
  $tB_R = tB_T - \mathrm{Cost}(A_I)$; // Calculate the rest of transmission constraint
Function Allocation(Dict[0], flag)*:*
  Dict_{new}[0][0] = Narrow(Dict[0][0]); // Reduce the size of the threshold interval
  Thre_{change} = Minus(Dict[0][0], Dict_{new}[0][0]); // Calculate the change scope of the threshold interval
  Dict[0][0] = Dict_{new}[0][0];
  if flag **then**
    R_{new} = Dict_{new}[0][0]["ratio"]/2;
  else
    R_{new} = Dict_{new}[0][0]["ratio"] $*$ 2;
  Update A_I; // Allocate R_{new} on the narrow scope of Thre_{change} while remain Dict_{new}[0][0]["ratio"] on Dict_{new}[0][0]
  Update $tB_R = tB_T - \mathrm{Cost}(A_I)$;
while $tB_R > 0$:
  $\mathrm{Dict_T}$[1] = $\mathrm{Sort}_{Descending}$($\mathrm{Dict_T}$[1]); // Based on the product of confidence and regions on each threshold interval.
  Update Allocation($\mathrm{Dict_T}$[1], 1);
while $tB_R < 0$:
  $\mathrm{Dict_T}$[0] = Sort($\mathrm{Dict_T}$[0]); // Based on the confidence of regions on each threshold interval.
  Update Allocation($\mathrm{Dict_T}$[0], 0);
$A_F = A_I$;
Return A_F;

and ground stations. Initially, compression ratios are allocated for each confidence interval using a default setting, and the remaining transmission capacity constraint is calculated (line 5). We first apply a high compression ratio to regions below the minimum threshold and the background. When surplus transmission capacity is available, we activate the first two confidence intervals. Specifically, we prioritize allocating the lowest compression ratio to the first interval and initially allocate the highest compression ratio to the second interval. If the transmission capacity is insufficient, we sort the images in the first interval in descending order of confidence and sequentially assign double the current compression ratio until there is enough capacity to download all images (lines 17–19). If there is additional transmission capacity remaining, we sort the images in the second interval based on the product of confidence and image regions and sequentially allocate half of the current compression ratio until the available transmission capacity is fully utilized (lines 20–22). Despite the high complexity of this method, it executes quickly due to the limited compression ratios supported by the satellite's hardware and the significant impact of specific confidence thresholds on system accuracy. Finally, all images annotated by swift in-orbit inference can be adaptively compressed and downloaded to ground stations.

D. Patch-Padding Image Restoration

After adaptive compressing and processing the image regions in orbit, we download these images to enhance system accuracy through analysis by powerful models on the ground. Due to the different compression ratios on these downloaded images, we need to perform a restoration process before analyzing them. An intuitive approach would be to restore these images to their original resolutions, utilizing restored images to compensate for the system's accuracy. However, this method may not be entirely feasible due to the associated high time costs.

To tackle this issue, we further explore the time expenditure at each step in the entire subsequent process. This analysis identified the resolution restoration of all compressed images as the critical bottleneck in the entire process. In our measurements, we observed that images subjected to high compression ratios accounted for 72.3% of the total restoration time. Additionally, within these highly compressed images, 63.6% of the regions are predominantly less significant, repetitive background regions. This observation led to a pivotal insight: selectively restoring only essential regions in the images, which have been subjected to lower ratios of compression. These restored segments could then be integrated into the highly compressed background regions, thereby synthesizing a new composite image, as shown in Fig. 5. Hence, we performed some preliminary experiments and found crucial observations from the results.

• *Observation 2: Patch-padding restoration appears to be more efficient for resolution-compensated scenarios:* Fig. 6 depicts the system accuracy and time consumption in various image recovery scenarios, particularly when dealing with heavily compressed images. Initially, images undergo heavy compression (8×). We then evaluate the efficacy of fully restoring the compressed images against restoring only predetermined annotated regions during the recovery process. Our findings reveal a significant reduction in recovery time by 69.7% when

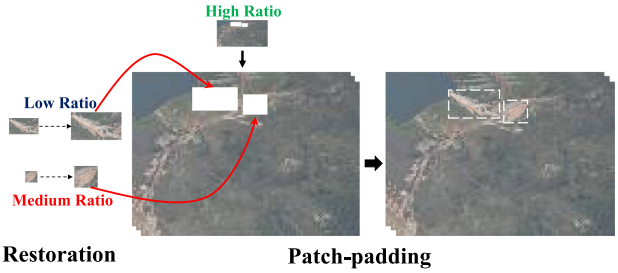


Fig. 5. The diagram of patch-padding restoration.

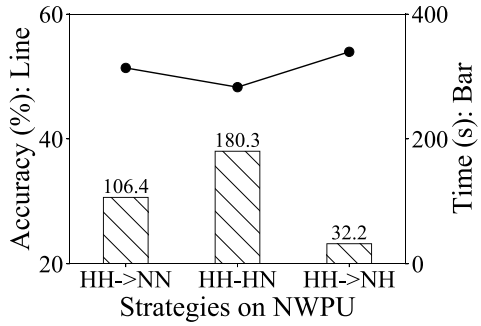


Fig. 6. The cost of various restoration strategies using DETR model. HH->NN: The compression ratio on all regions is 8, and all regions are restored. HH->HN: Restore the regions with low confidence. HH->NH: Restore the regions with high confidence.

restoring only the annotated regions, alongside a 5.0% increase in system accuracy. This improvement stems from the fact that the annotated regions in the images are crucial to the system's accuracy gains. Focusing restoration efforts on these specific regions not only minimizes recovery overhead, but also strategically concentrates coarse-grained "attention" on these regions, thereby amplifying their influence on system accuracy.

Hence, to enhance system performance with minimal overhead from the downloaded (compressed) images, we introduce a patch-padding resolution approach. This method entails a three-step design: initially restoring each annotated region in the compressed images; subsequently replacing these annotated regions in the compressed images with their restored counterparts; and finally, re-entering the patch-padding images into the system for tuning and re-inference. This re-inference process is designed to offset the diminished accuracy experienced during the rapid response phase. Note that, the essence of the patch-padding strategy lies in its ability to balance time efficiency against accuracy enhancement.

IV. Evaluation

We experimentally evaluated our approach in terms of end-to-end performance, response time, and compression.

A. Experiment Settings

To evaluate the performance of `AdaEO` on each dataset, we perform our experiments on a Ubuntu 18.04 Linux server with 8 NVIDIA A40 GPUs. Moreover, to simulate the limited computing capacity of the satellite and gain insights into runtime performance (response time), we utilize a real-world device:

TABLE I
DATASETS FOR EXPERIMENTS

Dataset	Size	GSD (m)	Volume (GB)
NWPU [44]	1000	0.5-2	0.1
RSOD [45]	1000	0.5-2	0.3
TGRS [46]	1500	0.15-1.2	3.4

Size: large-scale geospatial image resolution. GSD: geographic distance between adjacent pixels.

NVIDIA Jetson ORIN NX, which is a high-end edge computing platform suitable for deployment in vehicles and satellites, equipped with 16GB of RAM, 64GB of storage, and a 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores.

Datasets and models: We evaluate three classic geospatial datasets from diverse scenes, including vehicles, planes, overpasses, and oil tanks, as shown in Table I. Moreover, we employ a classical object detection model, called DEtection TRansformer [46] in our experiments, which consists of a set-based global loss and a Transformer encoder-decoder architecture.

Metrics: To quantify the performance of our system, we report the following metrics.

- *Accuracy:* The metric refers to mAP accuracy, a classic metric in object detection that considers both precision and recall across multiple categories.
- *Downlink volume:* The metric refers to the total amount of data transmitted from satellite to ground throughout the transmission process. It encompasses the cumulative size of all the images downloaded to the ground.
- *Final response time:* The metric signifies the time required to achieve convergence accuracy using a specific method. While the first response time denotes the time required to attain initial accuracy in orbit after tuning the deployed model on the satellite.

Baselines: We utilize three baselines in experiments to demonstrate AdaEO's performance:

- *SO:* All raw observational images are subjected to inference through the onboard models, and the resulting inferences are subsequently transmitted to the ground.
- *GO:* Satellites operate as bent-pipe systems [15], collecting raw observational images and transmitting them to the ground, then performing inference by the models deployed on the ground.
- *MADUN [2]:* The satellite-ground cooperation framework utilizes a state-of-the-art compressed sensing algorithm named MADUN, aligning with the processing workflow of AdaEO. The difference is that this framework executes uniform image compression in orbit, and then performs image restoration and re-inference on the ground.

B. End-to-End Performance

We show the overall performance of AdaEO compared to baselines. Fig. 7 shows the convergence accuracy performance of AdaEO compared to three baselines.

AdaEO compensates for the system accuracy after in-orbit compression and achieves accuracy close to GO results subsequently. Moreover, AdaEO demonstrates the capability to achieve 71.2% system accuracy (on average) within a relatively

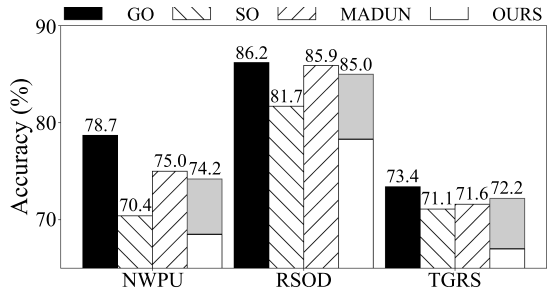


Fig. 7. The accuracy of AdaEO compared with three baselines. The gray area in the "OURS" represents the performance improvement that AdaEO can deliver compared to the performance at the first response time.

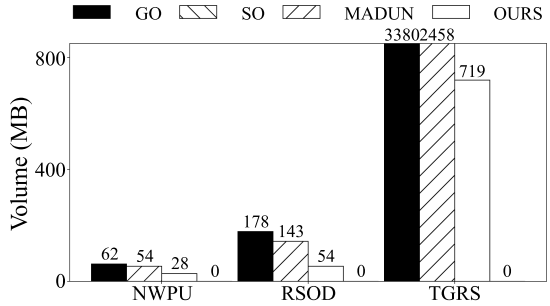


Fig. 8. The downlink volume of AdaEO compared with three baselines. SO does not need to use the download link.

short first response time. After subsequent accuracy compensation, it further offers 77.1% accuracy (on average). Compared to the final accuracy provided by SO, AdaEO achieves 95.7% performance in the first response, and the final compensation accuracy surpasses 3.8%. Compared to the final accuracy provided by GO, AdaEO achieves 89.6% performance in the first response and 97% accuracy in the final compensation. Compared to the final accuracy provided by MADUN, AdaEO achieves 91.9% performance in the first response and 99.2% accuracy in the final compensation. AdaEO's accuracy approaches that of MADUN and GO because AdaEO aims to preserve critical information in annotated regions during compression. The patch-padding restoration method further applies "attention" to annotated regions at the image granularity during subsequent compensation and re-inference.

Fig. 8 shows the performance comparison of downlink volume. The results show that AdaEO significantly reduces downlink volume compared to the baseline methods. Compared to GO, AdaEO significantly reduces downlink volume required by 71.6%. SO requires no downlink volume because its entire pipeline is executed in orbit. Compared to MADUN, AdaEO also reduces downlink volume by 62.8%. This is because AdaEO selectively and adaptively compresses downloaded images, i.e., applying low compression ratios to information in annotated regions with high confidence, while using high compression ratios on a large portion of unannotated regions (annotated regions with low confidence) with lower importance. Moreover, the unannotated regions with high compression ratios also contribute to the improvement of inference accuracy.

We further explore the response time required by the above methods, as shown in Fig. 9. The results show that across all datasets, AdaEO's first response time is as low as 0.5 seconds,

Methods	Time cost (seconds)		
	NWPU	RSOD	TGRS
GO	8,732.7	11,124.8	51,947.5
SO	9,812.6	7,849.7	29,231.1
MADUN	13,122.4	15,361.0	47,640.1
OURS (First)	0.5	0.7	4.1
OURS (Final)	6,244.0	5,476.5	33,203.9

Fig. 9. The response time of AdaEO compared to baselines.

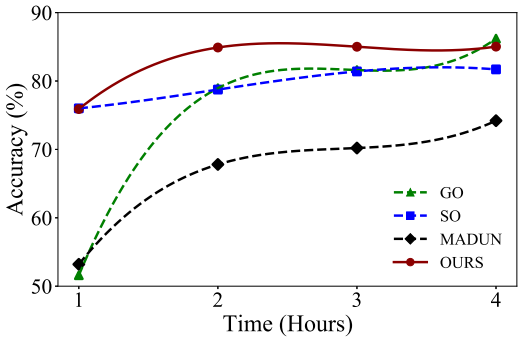


Fig. 10. The accuracy-time tradeoff of AdaEO compared to three baselines.

constituting less than 1‰ of the total pipeline response time. Compared to the response time required by the three baselines, the overall response time of AdaEO is only 61.5%, 77.7%, and 50.9%, respectively. Compared to MADUN, AdaEO only needs to recover annotated regions and find the corresponding regions in the full compressed images for replacement. Furthermore, SO and GO both require iterative training on all original images. GO needs to download all the original data to the ground before training, while AdaEO only needs to fine-tune the restored images. We conclude that, as satellite image resolution and quantity increase, the overhead associated with baselines will far exceed those of AdaEO.

C. Analysis of Response Time Constraints

We further investigate the performance of AdaEO compared to baselines under different time constraints. We maintain the hyper-parameter settings the same as the settings of the above end-to-end experiments. Fig. 10 illustrates the trade-off between response time and accuracy. We set the response times to 1, 2, 3, and 4 hours, and evaluate on the RSOD dataset in this experiment.

As observed, AdaEO consistently shows remarkable accuracy across different response times and nearly achieves the same accuracy as GO under extended response times. Specifically, when the response time is limited to less than one hour, AdaEO provides accuracy almost equivalent to SO (with only a 0.4% gap) and outperforms GO by 24.3%. Moreover, with the response time restricted to two and three hours, AdaEO surpasses both SO and GO by 3.3%−7.9% and 3.6%−6.7%, respectively. When the response time is limited to 4 hours, AdaEO enhances accuracy by 3.3% compared to SO, and achieves 98.6% of GO's performance. The underlying reason is that AdaEO swiftly and accurately annotates key regions of images, which is crucial for reducing response time and enhancing accuracy. Although SO can obtain results rapidly, it struggles to improve system accuracy. Conversely, GO can enhance system accuracy but requires downloading and processing all raw data, resulting in slower performance improvements. Notably, the current experiment is conducted on the RSOD dataset with a size of 0.3G, whereas real satellite computing scenarios involve larger quantities and sizes of images. In such cases, AdaEO can achieve higher accuracy in relatively shorter response times, demonstrating superior operational efficiency.

D. Analysis of Compression Condition Constraints

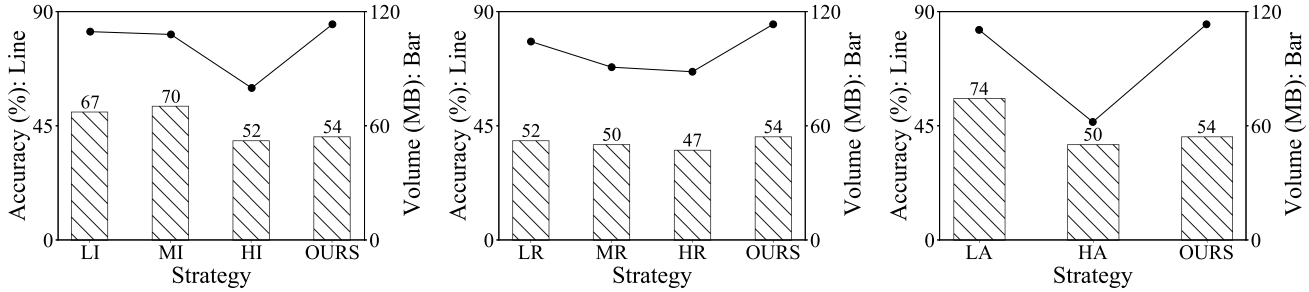
We investigate the performance of AdaEO across varying compression conditions. Specifically, we consider the following settings: i) Allocating different confidence threshold intervals; ii) Allocating different compression ratios; iii) Allocating different compression ratios on different confidence threshold intervals. We also keep the remaining hyper-parameters consistent with the experimental configuration.

Impact of threshold intervals: Fig. 11(a) reports the impact of threshold intervals on accuracy and downlink volume of AdaEO. We keep the compression ratios at 2, 4, and 8, and allocated three different confidence threshold intervals for these compression ratios as follows: i) LI: (1, 0.7), (0.7, 0.4), (0.4, 0.3); ii) MI: (1, 0.6), (0.6, 0.4), (0.4, 0.3); iii) HI: (1, 0.8), (0.8, 0.5), (0.5, 0.3). We explore the performance of AdaEO when applying the same compression ratio to regions of varying importance.

As observed, when lower compression ratios are allocated to regions of reduced importance, downlink volume increased by 28.8%−29.6%, while the accuracy performance of AdaEO decreased by 3.4%−4.7%. The underlying reason is that allocating lower compression ratios to these regions preserves some redundant information, increasing the pressure on downlink transmission, while it's difficult to gain accuracy improvement. Notably, the third setting resulted in a 29.5% decrease in system accuracy compared to OURS. The difference lies in the allocation of confidence threshold intervals for 4× compression ratio, indicating that even in annotated regions of low confidence, retaining some information to maintain system performance is necessary.

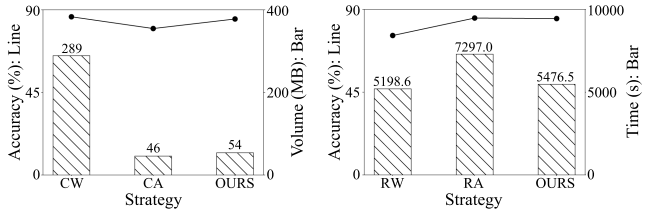
Impact of compression ratios: Fig. 11(b) illustrates the performance of AdaEO when different compression ratios are allocated to confidence threshold intervals. We keep the confidence threshold intervals at (1, 0.8), (0.8, 0.4), (0.4, 0.3), and allocated varying compression ratios in each interval as follows: i) LR: 2, 4, 16; ii) MR: 2, 8, 16; iii) HR: 4, 8, 16.

The results in Fig. 11(b) demonstrate that allocating higher compression ratios to regions of equal importance leads to a significant reduction in downlink volume, ranging from 3.7%−7.4%. However, this approach comes at a cost, as the accuracy performance of AdaEO decreases by 8%−19.8%. This decrease in accuracy occurs because regions of higher importance lose more information that is crucial for maintaining system accuracy as the compression ratio increases. Notably, in comparison to the second setting, the third setting results in only a 2.6% reduction in system accuracy. This is because, when the same compression ratio is uniformly applied to all regions, those of higher importance can more efficiently convey image information, thereby contributing to increased system accuracy while maintaining the same downlink volume conditions.



(a) Different confidence threshold intervals. (b) Different compression ratios. (c) Different partitions on threshold intervals.

Fig. 11. Effects of different AdaEO settings on system performance.



(a) Adaptive image compression. (b) Patch-padding restoration.

Fig. 12. Ablation study on key designs. CW: All images are not compressed. CA: The compression ratio on all images is 8. RW: All compressed images are not restored. RA: All compressed images are restored.

Impact of threshold intervals partitioning: Fig. 11(c) illustrates the performance of our AdaEO when different compression ratios are allocated to distinct confidence threshold intervals. We allocate different compression ratios to each interval sequentially as follows: i) LA: A compression ratio of 2 is allocated to (1, 0.8), and 4 is allocated to (0.8, 0.3). ii) HA: A compression ratio of 2 is allocated to (1, 0.8), 4 to (0.8, 0.6), 8 to (0.6, 0.4), and 16 to (0.4, 0.3).

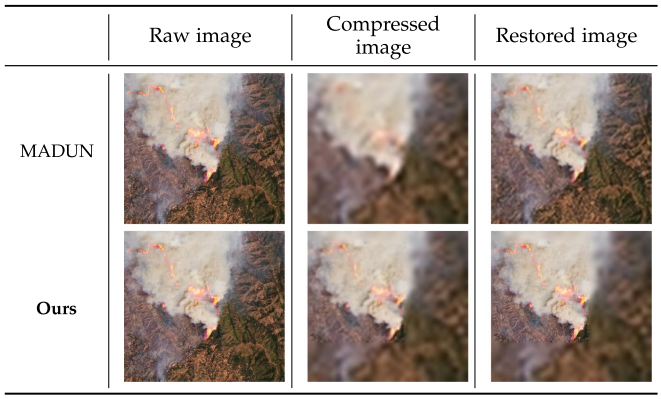
The results in Fig. 11(c) reveal that allocating different compression ratios on more finely defined confidence threshold intervals leads to the savings of downlink volume by 37% and 7.4%. However, the accuracy performance of AdaEO decreases by 2.5% and 45.2%, respectively. This phenomenon is attributed to the fact that when image regions are not of high importance, setting a lower compression ratio can result in an excess of redundant information. Consequently, increasing the compression ratio helps filter out much of this redundant information, which contributes minimally to system accuracy. Notably, compared to OURS, the second setting achieved a 7.4% reduction in downlink volume but led to a decrease of up to 45.2% in system accuracy. The underlying reason is that when compression ratios are already set to high values, further increasing them yields smaller savings in image space, implying reduced potential for conserving downlink volume. Simultaneously, as the amount of discardable redundant information diminishes, the impact on system accuracy becomes more pronounced.

E. Validation of Key Designs

Confidence-adaptive compression significantly reduces downlink volume: In Fig. 12(a), performing confidence-adaptive compression leads to a significant reduction in downlink volume while ensuring system accuracy. Compared to uncompressed transmission, adaptive compression reduces downlink volume by 81.3% with only a 1.4% loss in accuracy. In contrast to full compression, adaptive compression increases downlink volume by only 14.8% while improving accuracy by 6.6%. This is primarily due to the fact that adaptive compression retains as much of the information crucial for system gain as possible.

Patch-padding restoration effectively enhances systematic accuracy: Fig. 12(b) illustrates that the patch-padding restoration method can substantially enhance the system's accuracy. Compared to the method that omits image restoration after compression, the overall pipeline's time cost with the patch-padding restoration method increases by only 5.4%, while significantly improving accuracy by 12.2%. Furthermore, when contrasted with full restoration, the patch-padding restoration method cuts down the time required for the entire pipeline by 24.9%, while incurring a minimal accuracy loss of just 0.4%. This is because the patch-padding method focuses on directing coarse-grained 'attention' to the critical regions of the image during the resolution restoration process.

TABLE II
A VISUALIZATION OF AdaEO: THE CHANGES IN AN IMAGE ACROSS DIFFERENT STAGES



F. A Visualization of AdaEO

AdaEO can enhance the efficiency of rescue operations: Table II shows the changes of the image across different stages. The process begins with the rapid identification and response to critical regions while in orbit. Subsequently, AdaEO efficiently reduces downlink load by applying adaptive compression, tailored to the confidence levels of distinct regions. In the final

stage, we utilize the patch-padding technique to restore the resolution of the regions annotated as high confidence in the compressed image, thereby enhancing the overall system accuracy. Therefore, AdaEO not only assists in the rapid initiation of rescue operations, but also holds the potential to significantly contribute to the continuous enhancement of these operations through improved system accuracy.

V. Conclusion

In this paper, we introduce an effective satellite computing system framework through satellite-ground collaboration. Its primary emphasis is on providing post-disaster relief assistance, addressing the specific challenges arising from computational and downlink constraints. To this end, we formulate a transmission-constrained optimization problem to maximize model inference accuracy while guaranteeing response time requirements. We design an innovative pipeline that prioritizes in-orbit swift annotation to distinguish confidence regions within images and perform adaptive compression. Subsequently, the pipeline executes patch-padding restoration and accuracy compensation based on the downloaded compressed data. Experiments demonstrate that compared to traditional GO inference, AdaEO can achieve 89.6% accuracy in only less than 1‰ of the response time. Furthermore, AdaEO consistently improves inference accuracy up to 98.6% compared to GO inference while conserving 71.6% of downlink volume and 38.4% of the response time.

References

[1] Space infrastructure as a service, 1993. [Online]. Available: https://datacenterfrontier.com/data-centers-above-the-clouds-colocation-goes-to-space/

[2] J. Song, B. Chen, and J. Zhang, "Memory-augmented deep unfolding network for compressive sensing," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 4249–4258.

[3] M. Xiao et al., "Invertible image rescaling," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 126–144.

[4] M. Shen, H. Gan, C. Ning, Y. Hua, and T. Zhang, "TransCS: A transformer-based hybrid architecture for image compressed sensing," *IEEE Trans. Image Process.*, vol. 31, pp. 6991–7005, Nov. 2022, doi: 10.1109/TIP.2022.3217365.

[5] G. Denis, H. de Boissezon, S. Hosford, X. Pasco, B. Montfort, and F. Ranera, "The evolution of earth observation satellites in Europe and its impact on the performance of emergency response services," *Acta Astronautica*, vol. 127, pp. 619–633, 2016.

[6] J. Chen, L. Wei, and G. Zhao, "An improved lightweight model based on mask R-CNN for satellite component recognition," in *Proc. Int. Conf. Ind. Artif. Intell.*, 2020, pp. 1–6.

[7] R. Chen, X. Li, and S. Li, "A lightweight CNN model for refining moving vehicle detection from satellite videos," *IEEE Access*, vol. 8, pp. 221 897–221 917, 2020.

[8] Z. Zhang, A. Iwasaki, G. Xu, and J. Song, "Cloud detection on small satellites based on lightweight U-net and image compression," *J. Appl. Remote Sens.*, vol. 13, no. 2, 2019, Art. no. 026502.

[9] K. Yuan, X. Zhuang, G. Schaefer, J. Feng, L. Guan, and H. Fang, "Deep-learning-based multispectral satellite image segmentation for water body detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7422–7434, Jul. 2021, doi: 10.1109/JSTARS.2021.3098678.

[10] S. Ghassemi, A. Fiandrotti, G. Francini, and E. Magli, "Learning and adapting robust features for satellite image segmentation on heterogeneous data sets," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6517–6529, Sep. 2019.

[11] M. Wang, J. Wang, Y. Cui, J. Liu, and L. Chen, "Agricultural field boundary delineation with satellite image segmentation for high-resolution crop mapping: A case study of rice paddy," *Agronomy*, vol. 12, no. 10, 2022, Art. no. 2342.

[12] B. Denby, B. Lucia, S. Noghabi, K. Chintalapudi, and R. Chandra, "Kodan: Addressing the computational bottleneck in space," in *Proc. ACM Int. Conf. Architectural Support Program. Lang. Operating Syst.*, 2023, pp. 392–403.

[13] S. Wang, Q. Li, M. Xu, X. Ma, A. Zhou, and Q. Sun, "Tiansuan constellation: An open research platform," in *Proc. IEEE Int. Conf. Edge Comput.*, 2021, pp. 94–101.

[14] W. Shangguang, Z. Qiyang, X. Ruolin, Q. Fei, and X. Mengwei, "The first verification test of space-ground collaborative intelligence via cloud-native satellites," *China Commun.*, vol. 21, no. 4, pp. 208–217, 2024.

[15] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proc. Int. Conf. Architectural Support Program. Lang. Operating Syst.*, 2020, pp. 939–954.

[16] A. Chen, Y. Xie, Y. Wang, and L. Li, "Knowledge graph-based image recognition transfer learning method for on-orbit service manipulation," *Space Sci. Technol.*, vol. 2021, 2021, Art. no. 9807452.

[17] H. Li, C. Chen, C. Li, L. Liu, and G. Gui, "Aerial computing offloading by distributed deep learning in collaborative satellite-terrestrial networks," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, 2021, pp. 1–6.

[18] D. Liu, Z. Ma, A. Zhang, and K. Zheng, "MagicBatch: An energy-aware scheduling framework for DNN inference on heterogeneous edge servers in space-air-ground computation," in *Proc. Int. Conf. Big Data Intell. Comput.*, Springer, 2022, pp. 421–433.

[19] J. Guan, Q. Zhang, I. Murturi, P. K. Donta, S. Dustdar, and S. Wang, "Collaborative inference in DNN-based satellite systems with dynamic task streams," in *Proc. IEEE Int. Conf. Commun.*, 2024, pp. 3803–3808.

[20] B. Tao, O. Chabra, I. Janveja, I. Gupta, and D. Vasisht, "Known knowns and unknowns: Near-realtime earth observation via query bifurcation in serval," in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2024, pp. 809–824.

[21] M. Xu, F. Qian, Q. Mei, K. Huang, and X. Liu, "DeepType: On-device deep learning for input personalization service with minimal privacy concern," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 4, 2018, Art. no. 197.

[22] Y. Kim, J. Kim, D. Chae, D. Kim, and J. Kim, "μlayer: Low latency on-device inference using cooperative single-layer acceleration and processor-friendly quantization," in *Proc. Eur. Conf. Comput. Syst.*, 2019, pp. 1–15.

[23] S. Laskaridis, A. Kouris, and N. D. Lane, "Adaptive inference through early-exit networks: Design, challenges and directions," in *Proc. Int. Workshop Embedded Mobile Deep Learn.*, 2021, pp. 1–6.

[24] Y. Chen, Q. Zhang, Y. Zhang, X. Ma, and A. Zhou, "Energy and time-aware inference offloading for DNN-based applications in Leo satellites," in *Proc. IEEE Int. Conf. Netw. Protoc.*, 2023, pp. 1–6.

[25] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2704–2713.

[26] Y. Kang et al., "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM Special Int. Group Architecture Comput. Syst.*, vol. 45, no. 1, pp. 615–629, 2017.

[27] S. Liu, Y. Lin, Z. Zhou, K. Nan, H. Liu, and J. Du, "On-demand deep model compression for mobile devices: A usage-driven model selection framework," in *Proc. Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 389–400.

[28] Q. Zhang et al., "A comprehensive benchmark of deep learning libraries on mobile devices," in *Proc. ACM Web Conf.*, 2022, pp. 3298–3307.

[29] Q. Zhang et al., "A comprehensive deep learning library benchmark and optimal library selection," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5069–5082, May 2024.

[30] R. Yi, T. Cao, A. Zhou, X. Ma, S. Wang, and M. Xu, "Boosting DNN cold inference on edge devices," in *Proc. Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2023, pp. 516–529.

[31] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, Mar. 2020.

[32] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image compressed sensing using convolutional neural network," *IEEE Trans. Image Process.*, vol. 29, pp. 375–388, Jul. 2020, doi: 10.1109/TIP.2019.2928136.

[33] X. Yuan and R. Haimi-Cohen, "Image compression based on compressive sensing: End-to-end comparison with JPEG," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2889–2904, Nov. 2020.

[34] R. Heckel and M. Soltanolkotabi, "Compressive sensing with un-trained neural networks: Gradient descent finds a smooth approximation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4149–4158.

[35] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust compressed sensing MRI with deep generative priors," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 14 938–14 954.
[36] M. Kabkab, P. Samangouei, and R. Chellappa, "Task-aware compressed sensing with generative adversarial networks," in *Proc. Assoc. Advance. Artif. Intell. Conf.*, 2018, pp. 2297–2304.
[37] PNG, 1995. [Online]. Available: https://medium.com/@duhroach/how-png-works-f1174e3cc7b7
[38] ITU-T81, 1993. [Online]. Available: https://github.com/SixLabors/ImageSharp/blob/main/src/ImageSharp/Formats/Jpeg/itu-t81.pdf
[39] S. Wenkel, K. Alhazmi, T. Liiv, S. Alrshoud, and M. Simon, "Confidence score: The forgotten dimension of object detection performance evaluation," *Sensors*, vol. 21, no. 13, 2021, Art. no. 4350.
[40] M. Tanji and M. Thayer, "Maui on fire," 2023. [Online]. Available: https://www.mauinews.com/news/local-news/2023/08/maui-on-fire/
[41] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 784–799.
[42] D. Li et al., *Nonlinear Integer Programming*. Berlin, Germany: Springer, 2006.
[43] H. Su, S. Wei, M. Yan, C. Wang, J. Shi, and X. Zhang, "Object detection and instance segmentation in remote sensing imagery based on precise mask R-CNN," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 1454–1457.
[44] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2486–2498, May 2017.
[45] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, "Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5535–5548, Aug. 2019.
[46] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

Chen Yang is currently working toward the PhD degree in computer science with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He has published papers in MobiCom, *IEEE Transactions on Mobile Computing*, etc. His research interests include satellite edge computing and resource-efficient AI systems.

Qibo Sun received the PhD degree in communication and electronic system from the Beijing University of Posts and Telecommunication, in 2002. He is an associate professor with the School of Computer Science and Engineering, Beijing University of Posts and Telecommunications, China. His research interests include services computing, satellite computing, and space-ground integration network. He has published more than 100 papers. He is a member of the China Computer Federation and Chinese Association for Artificial Intelligence.

Qiyang Zhang received the PhD degree in computer science from the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Now, he is postdoc researcher with Peking University. He is also a visiting student with Distributed Systems Group, TU Wien from December 2022 to December 2023. He has published papers in WWW, INFOCOM, *IEEE Transactions on Mobile Computing*, etc. His research interests include satellite computing, edge intelligence.

Hao Lu is currently working toward the master degree with the School of Computer Science, Beijing University of Posts and Telecommunication, Beijing, China. His research interests include resource-efficient AI systems and federated learning.

Claudio A. Ardagna (Senior Member, IEEE) is full professor with Dipartimento di Informatica, Università degli Studi di Milano, the director of the CINI National Lab on Data Science, and co-founder of Moon Cloud srl. His research interests include cloud-edge-satellite security and assurance, and distributed system and AI certification. He has been visiting professor with the Université Jean Moulin Lyon 3 and visiting researcher with the Beijing University of Posts and Telecommunications, Khalifa University, George Mason University. He is member of the Steering Committee of *IEEE Transactions on Cloud Computing*, member of the editorial board of *IEEE Transactions on Cloud Computing* and *IEEE Transactions on Services Computing*.

Shangguang Wang (Senior Member, IEEE) is a professor with the School of Computer Science, Beijing University of Posts and Telecommunications, China. He is the founder & chief scientist with the Tiansuan Constellation. His research interests include service computing, mobile edge computing, and satellite computing. He is currently serving as chair of IEEE Technical Committee on Services Computing, and vice chair of IEEE Technical Committee on Cloud Computing. He also served as general chairs or program chairs of more than 10 IEEE conferences. He is a fellow of the IET. For further information on him, please visit: http://www.sguangwang.com.

Mengwei Xu is an associate professor with the Computer Science Department, Beijing University of Posts and Telecommunications. His research interests cover the broad areas of mobile computing, edge computing, artificial intelligence, and system software.