



# Energy-Aware Satellite-Ground Co-Inference via Layer-Wise Processing Schedule Optimization

Yijie Chen

yijiechen@bupt.edu.cn

State Key Laboratory of Networking  
and Switching Technology, Beijing  
University of Posts and  
Telecommunications  
Beijing, China

Qiyang Zhang

qyzhang@bupt.edu.cn

State Key Laboratory of Networking  
and Switching Technology, Beijing  
University of Posts and  
Telecommunications  
Beijing, China

Ruolin Xing

xrl@bupt.edu.cn

State Key Laboratory of Networking  
and Switching Technology, Beijing  
University of Posts and  
Telecommunications  
Beijing, China

Yuanzhe Li

liyuanzhe@air.tsinghua.edu.cn

Institute for AI Industry Research  
(AIR), Tsinghua University  
Beijing, China

Xiao Ma

maxiao18@bupt.edu.cn

State Key Laboratory of Networking  
and Switching Technology, Beijing  
University of Posts and  
Telecommunications  
Beijing, China

Chaoxin Yu

yuchaoxin@gd.chinamobile.com

China Mobile GBA Innovation  
Institute  
Guangzhou, China

Yiran Zhang

yiranzhang@bupt.edu.cn

State Key Laboratory of Networking  
and Switching Technology, Beijing  
University of Posts and  
Telecommunications  
Beijing, China

Ao Zhou\*

aozhou@bupt.edu.cn

State Key Laboratory of Networking  
and Switching Technology, Beijing  
University of Posts and  
Telecommunications  
Beijing, China

Shangguang Wang

sgwang@bupt.edu.cn

State Key Laboratory of Networking  
and Switching Technology, Beijing  
University of Posts and  
Telecommunications  
Beijing, China

## ABSTRACT

Recent advancements in Low Earth Orbit (LEO) satellites are facilitating the provision of Deep Neural Networks (DNNs)-inherent services to achieve ubiquitous coverage via satellite computing. However, the computational demands and energy consumption of DNN models pose significant challenges for satellite computing with limited power and computation resources. Based on the hierarchical characteristics of DNN models, we propose a satellite-ground co-inference strategy that executing certain layers on satellites and the remaining layers on ground servers. However, identifying the optimal layers for in-orbit processing with latency constraints is challenging due to the uncertain energy consumption across diverse models. To explore the correlation between energy consumption and layer types, we conduct comprehensive measurements on a hardware device commonly found in commercial LEO satellites and develop a layer-based energy consumption prediction model. Then,

we formulate an optimization problem of minimizing the energy consumption on the satellite within the latency constraint as an integer nonlinear programming problem. Solving this problem is difficult due to combinatorial explosion in the discrete solution space. To address this, we propose an improved algorithm based on genetic algorithms. Using configurations from a real satellite, we conduct simulation experiments, concluding that our algorithm significantly improves energy savings by an average of 27%.

## KEYWORDS

LEO satellite, satellite computing, energy prediction, satellite-ground co-inference, Deep Neural Networks

## ACM Reference Format:

Yijie Chen, Qiyang Zhang, Ruolin Xing, Yuanzhe Li, Xiao Ma, Chaoxin Yu, Yiran Zhang, Ao Zhou, and Shangguang Wang. 2024. Energy-Aware Satellite-Ground Co-Inference via Layer-Wise Processing Schedule Optimization. In *15th Asia-Pacific Symposium on Internetware (Internetware 2024)*, July 24–26, 2024, Macau, China. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3671016.3674811>

## 1 INTRODUCTION

With the growing demand for space exploration and technological advancements, Low Earth orbit (LEO) satellites, as demonstrated by prominent constellations Telesat, OneWeb, and SpaceX, are rapidly evolving [6]. Due to their capacity for comprehensive coverage, LEO satellites present a new opportunity for numerous services

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

*Internetware 2024, July 24–26, 2024, Macau, China*

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0705-6/24/07

<https://doi.org/10.1145/3671016.3674811>

inherent to Deep Neural Networks (DNNs), such as Earth observation involving the capture of ground imagery and the utilization of DNN models to analyze occurrences (e.g., remote monitoring, weather forecasting and emergency response). However, the constrained energy and computational resources of satellites, coupled with limitations in downlink transmission, pose urgent challenges in delivering these services under latency constraints [20].

There are currently two primary approaches to address this issue. The predominant approach involves satellites capturing ground imagery and transmitting all images back to Earth for processing, employing a conventional technique known as the “bent-pipe” architecture [10]. Despite its potential, satellite-ground communication has limited capacity and often experiences unreliable, intermittent disruptions. These interruptions frequently result in a significant backlog of data awaiting transmission. Alternatively, the latest research is focused on exploring the computational potential of satellites, and the concept of satellite computing has been proposed and applied [9, 17]. A novel approach proposes executing DNN models directly on the satellite for satellite computing [7]. However, due to the current capability of LEO satellites and energy sources from solar power generation, executing large-scale models (i.e., those composed of multiple layers) in orbit has become exceedingly challenging for satellite computing [31]. While small-scale DNN models are being considered, they may not ensure the same level of satellite computing accuracy [15].

To address the challenge, given that energy consumption and latency are influenced by layers, we break down the structure of the DNN model into individual layers, treating each layer as a distinct and independent subtask. The input for each layer derives from the output of the preceding layer. A promising strategy entails executing select layers on satellites, transmitting intermediate results to ground stations, and subsequently executing the remaining layers on the ground. As the output matrices of layers in models typically decrease compared to the initial inputs, this approach efficiently utilizes the limited resources of LEO satellites, resulting in a significant reduction in transmitted data size.

However, since the energy consumption of different models is unknown and variable, accurately executing which layers on satellites poses a challenge. Exploring the correlation between energy consumption and layer types necessitates obtaining energy consumption data for various layers. Yet measuring the energy consumption for each layer across diverse models is costly due to model complexity or even infeasible due to high inference time overhead, particularly with time-consuming layers. To tackle this challenge, we devise and implement a comprehensive measurement framework on a hardware device to assess energy consumption across different layer configurations. We summarize our findings and identify the critical yet unexplored issue of the nonlinear correlation between energy and essential layers. Drawing from these insights, we propose a generalized layer-based energy consumption prediction model corresponding to specific layers and models. This predictor accurately estimates energy consumption for DNN models at the layer level.

The constrained energy and computation capabilities of LEO satellites, along with transmission capacity, collectively impact in-orbit execution latency and energy consumption. This presents a complex challenge for co-inference between satellites and ground

stations. To facilitate energy-efficient satellite computing, we formulate a problem aiming at minimizing energy consumption with latency constraints on the satellite. Additionally, we propose an enhanced algorithm to effectively address this problem. In order to make the simulation results more realistic, we use real-world configurations from the in-orbit satellite to conduct the simulations. Results demonstrate that the proposed algorithm can support energy-efficient satellite computing.

The main contributions of this paper are as follows.

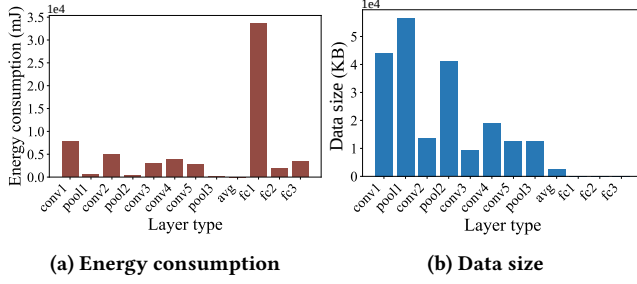
- We conduct a preliminary measurement for energy consumption on diverse layers and models. Then summarize the insightful observations and propose a general layer-based energy consumption prediction model corresponding to specific kernels and models.
- We formulate a DNN layer-wise processing schedule optimization problem for satellite-ground co-inference via the proposed prediction model. Then we propose an improved genetic algorithm-based method to solve the problem.
- We use configurations from a real satellite to simulate the experiments. The results indicate that our algorithm effectively saves energy 27× on average under various conditions.

## 2 RELATED WORK

**Satellite computing.** With the rapid advancement of satellite technology, current satellites can be equipped with computing payloads to facilitate computational processing capabilities. Recently, in-orbit computing has attracted significant attention within the satellite community. For example, OEC [9] and Kodan [7] aim to enhance downlink connections’ efficiency, particularly in scenarios where satellites are saturated, by performing partial processing and data filtering on satellites through early rejection of non-valuable imagery. Additionally, Zhang et al. [39] introduced a satellite-ground cooperative system, employing shallow models for processing satellite remote sensing data and deep DNN models on the ground for more precise computations. However, both these works overlook the restricted energy consumption onboard, leading to computational bottlenecks for DNN model-based applications.

**Shortage of energy on satellites.** Most LEO satellites encounter computational bottlenecks due to their small size and reliance on solar energy collection. Some work has explored satellite energy consumption. Xing et al. [31] observed the insufficient energy on the shaded side as a primary constraint for satellite computing advancement. Yang et al. [33] investigated the correlation between the number of charge-discharge cycles and the discharge depth of satellite solar panels with battery lifespan, suggesting that controlling these cycles could extend battery life. Thus, developing a strategic decision-making system to maximize energy efficiency in orbit while limiting charge-discharge depth holds significant importance. Therefore, our proposal focuses on energy-aware in-orbit inference while considering the general model structure of employed DNN models.

**Layered DNNs computation.** DNN models typically consist of various layers, each requiring specific computing resources and consuming energy accordingly [37, 38]. This insight allows large-scale DNN tasks to be fragmented into layer-specific subtasks [5], as demonstrated by numerous successful experiments.



**Figure 1: The variation in energy consumption and data size across each layer of AlexNet.**

**Table 1: Configurations for each layer type.**

| Layer type | Configuration                   |
|------------|---------------------------------|
| conv+relu  | ( $HW, K, S, C_{in}, C_{out}$ ) |
| pool       | ( $HW, K, S, C_{in}$ )          |
| fc         | ( $HW, K, S, C_{in}, C_{out}$ ) |

Hu et al. [12] introduced DNN surgery for concurrent processing of segmentation DNNs at edge and cloud nodes. Jeong et al. [13] proposed a partition-based loading technique for DNN computations. Currently, DNN partitioning for edge computing is not directly applicable to satellite systems due to the short communication windows between satellites and ground stations. In contrast, our study extensively examines real-world energy consumption of individual layers using a generic model, enabling precise energy profiling of layers and resource-efficient inference tasks in satellite orbits.

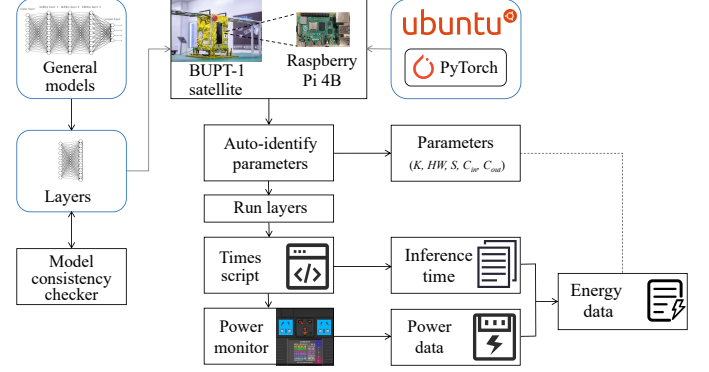
### 3 MEASUREMENT AND ANALYSIS

We present a reproducible methodology for energy measurement, facilitating in-depth analysis of layer energy consumption. Building on the insights gained from these measurements, our objective is to develop a layer-based energy consumption prediction suitable for general DNN models deployed on satellites.

In this paper, our focus is on layer configuration, as a complete DNN model comprises layer-level units with diverse layer types. The different types of layers-level parameters of each layer contribute significantly to layer variations in energy consumption during DNN model execution. To delve deeper into the impact of different configurations of layers, we break down the DNN models. Figure 1 illustrates the energy consumption and data size of each layer in AlexNet, revealing a notable discrepancy across the various layers. Therefore, analyzing the energy usage across various layer configurations is fundamental to establishing a thorough comprehension of energy consumption during DNN model executions.

#### 3.1 Measurement Settings

**Device and tools.** Commercial Off-The-Shelf (COTS) device is commonly found in commercial LEO satellites, improving computing ability and accelerating the development of satellite computing [2]. As illustrated in Figure 2, we built a hardware platform



**Figure 2: The workflow of measurement.**

on a COTS device with the same configuration as a real satellite “BUPT-1”. Given the impracticality of monitoring the energy consumption of individual layers in orbit, we conduct a comparison of the power consumption while executing DNN models on both ground hardware platform and “BUPT-1” satellite. The existing research confirms that the energy consumption is nearly identical between ground-based and satellite operations. There exists only a minimal difference in the performance of the same tasks whether during daylight or eclipse [31]. Therefore, employing ground-based devices to simulate computing devices on satellites for acquiring energy consumption data is reasonable. Additionally, all power consumption data is obtained using the *ATORCH* power monitor, and inference time is measured using a time script built by the PyTorch inference library [21].

**Dataset.** We utilize the xView<sup>1</sup> dataset in the context of satellite remote sensing for satellite computing. xView is one of the largest publicly available datasets of satellite imagery. This dataset comprises images of diverse global scenes, annotated with bounding boxes.

**Models.** Our measurement involved the utilization of layer-type parameter configurations drawn from numerous renowned models, including AlexNet [14], GoogLeNet [28], VGG [27], and MobileNet [11] and so on, enabling us to assess thousands of diverse parameter configurations across various layer types.

**Rules for measurement.** This paper focuses on three types of layers: convolution (conv+relu<sup>2</sup>), pooling (pool), and fully-connected (fc) layers, as they are the primary contributors to the energy consumption of DNN operations [26, 30]. Table 1 shows their respective configurations. For parameter selection, we referred to the parameter ranges commonly used in DNN models (such as the layer size in convolution layers typically being {1, 3, 5, 7, 9, 11}) and constructed single-layer models with conv+relu, pool, and fc layers.

**Workflow.** Figure 2 illustrates the overall workflow of our measurement. For each DNN model, we initially break down this model into several layers, and employ a model consistency checker to validate the logic of model partitions [36]. Next, we push the layer and the library to the satellite, and automatically extract and save the parameters of each layer. Following, we execute the layers, during which the satellite loads the library and layers into memory for

<sup>1</sup><http://xviewdataset.org/>

<sup>2</sup>In this paper, + represents kernel fusion.

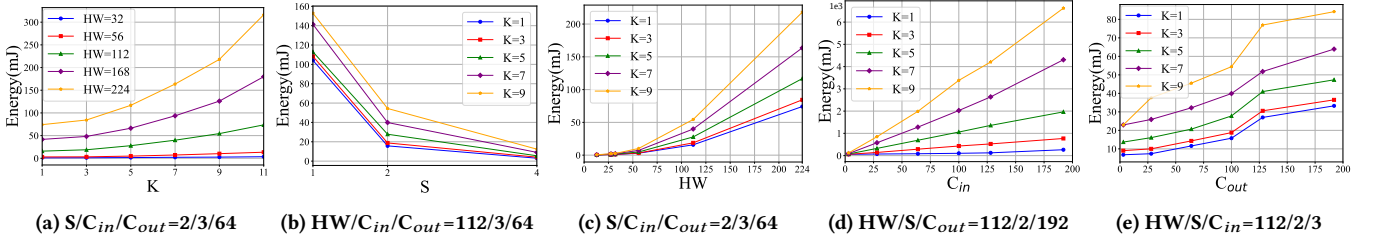


Figure 3: Energy consumption of conv+relu vs. different configurations.

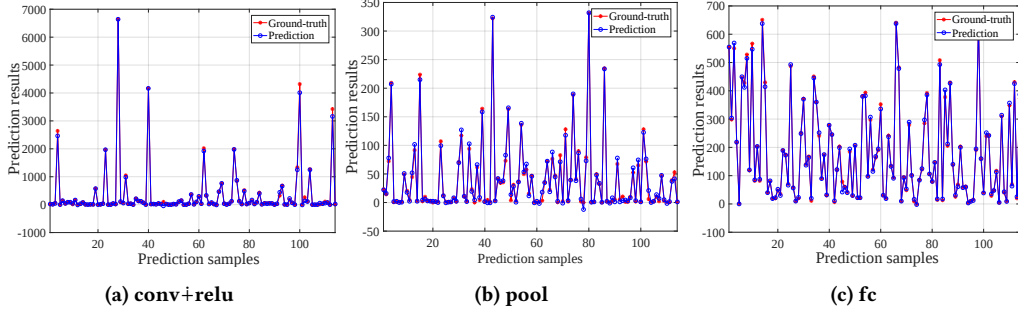


Figure 4: Comparison of prediction results and the ground-truth.

warm-up and performs inference  $N$  times (typically 200 by default) [34, 38]. Simultaneously, we employ a timing script to measure the average inference time of each layer and a power monitor to record the average power consumption of each layer. Multiplying the two values yields the energy consumption of executing each layer. Finally, we store the energy consumption along with the corresponding parameters to the dataset.

### 3.2 Results and Implications

To study the impact of 5 configurations ( $K, S, HW, C_{in}, C_{out}$ ) on the energy consumption of the conv+relu layer, we test different configurations with varying energy levels. In Figure 3, each evaluation changes one configuration while keeping the other four constant.

The trends show that all 5 parameters impact energy consumption either positively or negatively, but each has a non-linear correlation. We observe a disproportionate influence of these configurations on energy consumption. For instance,  $HW$  has the most significant effect on the energy consumption of the conv+relu layer. As shown in Figure 3, when the other parameters are held constant and  $HW$  (from 112 to 224),  $K$  (from 3 to 5),  $C_{in}$  (from 64 to 128),  $C_{out}$  (from 32 to 64), and  $S$  (from 4 to 2) each double, the corresponding changes in energy consumption are  $4.7\times$  (from 15.764mJ to 74.154mJ),  $1.3\times$  (from 16.037mJ to 21.544mJ),  $1.4\times$  (from 508.609mJ to 735.523mJ),  $1.5\times$  (from 350.038mJ to 539.060mJ), and  $2.17\times$  (from 55.239mJ to 25.394mJ). Furthermore, the variation in  $S$  is negatively correlated with energy consumption, while the other four parameters ( $HW, K, C_{in}, C_{out}$ ) are positively correlated, consistent with their correlation with computational load.

**Insights:** The aforementioned observation indicates that the configuration of 5 parameters ( $HW, K, S, C_{in}, C_{out}$ ) has a significant impact on the energy consumption of conv+relu layer. Hence, the

measurement motivates us to investigate the patterns between energy consumption and layer configuration.

### 3.3 Layer-based Energy Consumption Prediction

To explore the energy consumption patterns specific to each layer type, the energy consumption of different layers should be obtained. Yet measuring the energy consumption for each layer from different models is costly due to the complexity of models, or even infeasible due to the high inference time overhead [8]. Therefore, the establishment of a universal model for predicting DNN layer-based energy consumption is necessary. To deal with this challenge, we propose a regression prediction algorithm based on Particle Swarm Optimization for the BP neural network (PSO-BPR), thereby establishing the correlation between parameters and energy consumption. Moreover, our objective in designing the prediction model is to ensure both prediction accuracy and avoidance of overfitting. So we partition the entire dataset into training and testing sets in a 7:3 ratio and subsequently train the prediction model on the training set.

Algorithm 1 details the DNN layer-based energy prediction algorithm based PSO-BPR. Firstly, the network, along with the positions and velocities of each particle in the particle swarm, are initialized (lines 1-7). Subsequently, in a loop, the positions and velocities of the particle swarm are iteratively updated to progressively approach the optimal solution (lines 14-31). Within each iteration, the Fitness function is employed to determine the current optimal position and evaluating the results using the RMSE error predicted by the constructed BP network (lines 8-13).

We achieve approximate prediction results, as illustrated in Figure 4, where the blue signifies predictions and the red signifies the ground-truth. To quantify the prediction accuracy, we calculate

**Algorithm 1:** DNN Layer-Based Energy Prediction

---

**Input:** Layer-based energy consumption dataset:  $train\_p$ ,  $train\_t$ . The number of parameter types:  $inputnum$ . Particle swarm size:  $N$ .

**Output:** A precise layer-based energy prediction network.

- 1 **Initialize** a random population of individuals  $\{x_i\}$ ,  $i \in [1, N]$ ;
- 2 **Initialize** each individual's  $n$ -element velocity vector  $v_i$ ,  $i \in [1, N]$ ;
- 3 **Initialize** the best-so-far position of each individual:  
 $b_i \leftarrow \arg \min_x \{Fitness(x_i)\}, i \in [1, N]$ ;
- 4 **Initialize** the BP net;
- 5 **Define** the neighborhood size  $\sigma < N$ ;
- 6 **Define** the influence values  $\omega_1, \omega_2$ ;
- 7 **Define** the max and min velocity  $v_{max}, v_{min}$ ;
- 8 **Function**  $Fitness(\{x_i\}, net)$ :
- 9      $net.(w_1, w_2, B_1, B_2) \leftarrow (x_i[1 : n1], x_i[n1 + 1 : n2], x_i[n2 + 1 : n3], x_i[n3 + 1 : n4])$ ;
- 10     $net.train()$ ;
- 11     $error = \text{sum}(\text{RMSE}(net(train\_t)))$ ;
- 12    **return**  $error$ ;
- 13 **while** not (termination criterion) **do**
- 14    **foreach** individual  $x_i$  **do**
- 15      $H_i \leftarrow \sigma$  nearest neighbors of  $x_i$ ;
- 16      $h_i \leftarrow \arg \min_x \{Fitness(x) : x \in H_i\}$ ;
- 17      $v_i \leftarrow v_i + \omega_1 \cdot (b_i - x_i) + \omega_2 \cdot (h_i - x_i)$ ;
- 18     **if**  $|v_i| > v_{max}$  **then**
- 19        $v_i \leftarrow \frac{v_i \cdot v_{max}}{|v_i|}$ ;
- 20     **end**
- 21     **if**  $|v_i| < v_{min}$  **then**
- 22        $v_i \leftarrow \frac{v_i \cdot v_{min}}{|v_i|}$ ;
- 23     **end**
- 24      $x_i \leftarrow x_i + v_i$ ;
- 25      $b_i \leftarrow \arg \min \{Fitness(x_i), Fitness(b_i)\}$ ;
- 26    **end**
- 27     $net.(w_1, w_2, B_1, B_2) \leftarrow (x_i[1 : n1], x_i[n1 + 1 : n2], x_i[n2 + 1 : n3], x_i[n3 + 1 : n4])$ ;
- 28     $net.train()$ ;
- 29 **end**

---

the accuracy percentages. For the conv+relu layers, the accuracy within  $\pm 10\%$  is 54.1%, and within  $\pm 15\%$  is 62.3%. For the pooling layers, the accuracy within  $\pm 10\%$  is 48.8%, and within  $\pm 15\%$  is 52.2%. For the fc layers, the accuracy within  $\pm 10\%$  is 56.6%, and within  $\pm 15\%$  is 64.4%. These results demonstrate the highest level of accuracy in energy consumption prediction compared to existing prediction models [26, 30].

## 4 SYSTEM MODEL

In this section, we initially present the comprehensive satellite-ground co-inference model. Subsequently, we introduce the energy

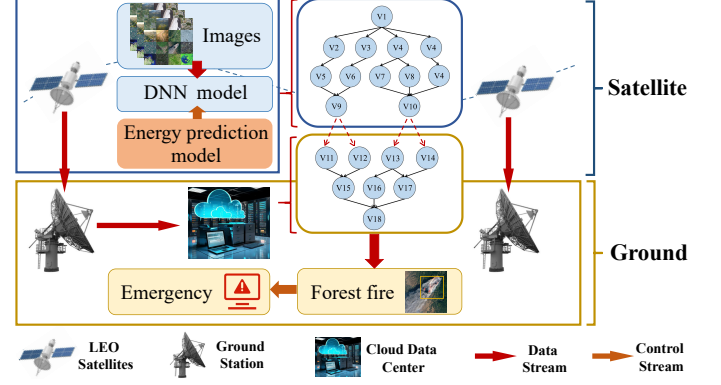


Figure 5: Satellite-ground co-inference system model.

system aboard the satellite. Finally, we formulate the layer-wise computing and transmission model.

### 4.1 Satellite-Ground Co-Inference Architecture

Our model necessitates partitioning a DNN model into two segments so that one is processed at the satellite and the other on the ground, as illustrated in Figure 5. The satellite segment comprises LEO satellites equipped with diverse payloads. The cameras capture observations of the Earth, then the computing module executes part of the DNN layers to process the captured images. Finally, the transmission module transmits the intermediate data results to the ground. The ground segment consists of ground stations and cloud data centers. The ground stations relay the intermediate data to cloud computing centers for final computations.

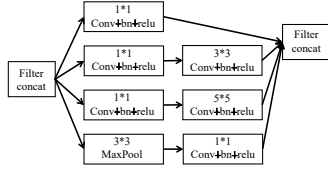
To facilitate expression, we model a DNN as a Directed Acyclic Graph (DAG), where each vertex corresponds to a layer, and the links between vertices denote layer dependencies [18]. Let  $\Gamma = (V, \mathcal{E})$  denote the DAG of a DNN model, where  $V = \{v_1, v_2, \dots, v_n\}$  is the set of vertices representing the layers of the DNN. The set  $\mathcal{E}$  represents the edges. An edge  $(v_i, v_j) \in \mathcal{E}$  signifies that  $v_i$  must be processed before  $v_j$ , and  $v_i$  provides output data to  $v_j$  as input data. Figure 7 illustrates the DAG of the inception v1 of GoogLeNet as depicted in Figure 6. In Figure 7,  $V = \{v_1, v_2, \dots, v_9\}$  represents the fc, conv+bn+relu, and pooling layers, while  $\mathcal{E} = \{(v_1, v_2), (v_1, v_3), \dots, (v_8, v_9)\}$  depicts dependencies.

### 4.2 LEO Satellite Energy Model

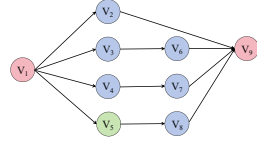
The satellite energy model consist of three components: energy harvesting, energy storage, and energy consumption [4, 19], illustrated in Figure 9.

**Energy harvesting.** Satellites derive their energy from solar panels capturing solar radiation. Statistics indicate that 85% of the energy consumption by LEO satellites is directly sourced from solar cell arrays [24]. Additionally, the power generated by solar cells correlates directly with the cosine of the solar incidence angle  $\theta$  (the angle between light and the normal to the panel) [29]. Therefore, the energy acquired by the solar cell array can be expressed as:

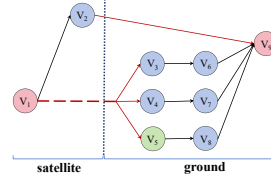
$$P_{solar} = P_{sun} \cdot \cos \theta \quad (1)$$



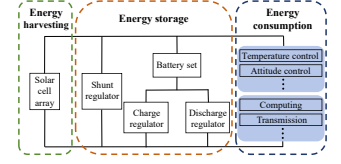
**Figure 6: Inception v1 model represented in layer form.**



**Figure 7: DAG of inception v1 model.**



**Figure 8: An illustration of DNN partition and data transmitting.**



**Figure 9: Satellite energy model.**

where  $P_{sun}$  is the direct solar power. And the energy obtained by the solar cell array is:

$$E_{solar} = P_{solar} \cdot t \quad (2)$$

**Energy storage.** The function of the energy storage system is to accumulate energy during daylight periods and discharge it during eclipse periods. This system comprises three regulators and battery sets. The shunt regulator serves to regulate the current, diverting a portion of the energy acquired from the solar panels directly to the payloads, denoted as  $E_{sup}^{solar}$ , while the remainder is directed to the battery set for storage. Consequently, the energy capacity that the battery set can store is determined by:

$$E_{bat} = E_{solar} - E_{sup}^{solar} \quad (3)$$

Attention must also be given to the lifespan of energy storage systems. Prolonging the lifespan of these systems necessitates restricting the depth of discharge [16]. For instance, nickel-cadmium batteries used in LEO satellites maintain a discharge depth within 10% to 20%, while nickel-hydrogen batteries typically range from 30% to 40% [1]. Consequently, the available energy of the battery system must satisfy the condition:  $E_{sup}^{bat} \leq \gamma \cdot E_{bat}^{max}$ , where  $\gamma$  represents the depth of discharge, and  $E_{bat}^{max}$  denotes the maximum battery energy capacity.

**Energy consumption.** Each component of the energy consumption system requires electrical energy, encompassing tasks such as attitude control, remote sensing, computing, and transmission [3, 23]. Certain modules are dedicated to maintaining the satellite's essential functions and are accorded higher priority for energy allocation. This segment of power is denoted as  $P_{sys}$ , with energy consumption defined by  $E_{sys} = P_{sys} \cdot t$ . Other modules serve specific functions, in our satellite-ground co-inference task, these include computing and transmission modules, with energy consumption denoted as  $E_{task}$ . When energy onboard the satellite is insufficient, priority is given to the higher-priority components, necessitating the shutdown of task payloads. Therefore, the available energy for task modules is determined by  $E_{task} = E_{sup} - E_{sys}$ , where  $E_{sup}$  represents the energy supplied by the satellite, sourced from both the solar cell array and the battery. Hence,  $E_{sup} = E_{sup}^{bat} + E_{sup}^{solar}$ , where  $E_{sup}^{solar}$  and  $E_{sup}^{bat}$  denote the energy provided by the solar cell array and the battery.

### 4.3 Layer-wise Computing and Transmission Model

We propose a layer-wise satellite-ground co-inference framework. Due to constraints on computing resources and energy on satellites,

executing a complete DNN model independently is an unachievable task for LEO satellites. Furthermore, based on research of existing DNN models, running certain layers in most DNN models often leads to changes in the data volume of intermediate results compared to the original input data [32]. Based on the layer-wised framework, we introduce the computing and transmission models following.

**Computing model.** We can predict the energy consumption  $E_{com}^{v_i}$  required for computing each layer, as presented in Section 3. We utilize a binary variable  $h_{v_i}$  to indicate where  $v_i$  is computed.  $h_{v_i} = 1$  represents that  $v_i$  is computed on the satellite, and  $h_{v_i} = 0$  represents that  $v_i$  is computed on the ground. Therefore, the total computing energy consumption on the satellite can be expressed as follows:

$$E_{com} = \sum_{i=1}^I h_{v_i} \cdot E_{com}^{v_i} \quad (4)$$

The satellite computing task can be completed when the energy  $E_{sup}$  provided by the satellite is at least equal to the sum of the computational energy  $E_{com}$  and  $E_{sys}$ . Therefore, the latency required for computation on the satellite can be expressed as follows:

$$t_{com}^{satellite} = \frac{E_{sup} - E_{com}}{P_{sys}} \quad (5)$$

After computation on the satellite, the intermediate results need to be transmitted to the ground. Let  $D_{v_i} = \alpha_{v_i} D_0$  represent the size of output data for each layer, where  $\alpha_{v_i}$  is the ratio of the output matrix for layer  $v_i$  to the initial matrix  $D_0$ . Therefore, the size of the intermediate results transmitted from the satellite to the ground, denoted as  $D$ , can be expressed as follows:

$$D = \sum_{i=1}^I \sum_{j=i+1}^I (C_{v_i v_j} \cdot (h_{v_i} - h_{v_j}) \cdot D_{v_i}) \quad (6)$$

Thus, the computing latency in the ground is  $t_{com}^{ground} = D \cdot \beta$ , where  $\beta$  represents the constant latency for processing 1MB of data in the ground cloud data center.

**Transmission model.** We establish a model to determine the maximum achievable bit rate based on the received signal power in the satellite downlink channel [9, 22]. The received signal power can be expressed as follows:

$$C = P_{trans} L_l G_t G_r \left( \frac{\lambda}{4\pi S} \right)^2 \quad (7)$$

where  $P_{trans}$  represents transmit power,  $L_l$  represents the line loss factor at the transmitter,  $G_t$  represents the transmitter gain parallel to the separation vector,  $G_r$  represents the receiver gain parallel to

the separation vector,  $\lambda$  is the center frequency of the channel, and  $S$  is the magnitude of the separation vector. The maximum bit rate can be expressed as follows:

$$R_{trans}^{max} = B \log_2 \left( 1 + \frac{C}{N} \right) \quad (8)$$

where  $B$  is the channel bandwidth,  $C$  is the received signal power as defined above, and  $N$  is the received noise power. The received noise power  $N = \kappa MB$ , where  $\kappa$  is the Boltzmann constant, and  $M$  is the system noise temperature. The system noise temperature of both satellite and ground is modeled as described in [35].

We can express the transmission energy consumption of DNN on the satellite. When layer  $v_j$  needs to offload to the ground for computing while its antecedent layer  $v_i$  is computed on the satellite. The output data of layer  $v_i$  should be transmitted to the ground station as intermediate results. As defined above, the transmission energy cost for offloading  $D_{v_i}$  from the satellite can be expressed as follows:

$$E_{trans}^{v_i} = P_{trans} \cdot t_{v_i} = P_{trans} \cdot \frac{D_{v_i}}{R_{trans}} \quad (9)$$

where  $R_{trans}$  is the real transmission data rate.

We consider all potential transmission scenarios across various DNN architectures. As the constructed DAG of the DNN mentioned above, the relationships between layers in the DNN can be depicted using a binary variable set  $C_{v_i v_j}$ . When  $C_{v_i v_j} = 1$ , it signifies an edge between  $v_i$  and  $v_j$ , and  $v_i$  is the antecedent layer of  $v_j$ . As depicted in Figure 8, when computing  $v_1, v_2$  on the satellite and the remaining layers on the ground,  $h_{v_1}$  and  $h_{v_2}$  both equal 1, while  $h_{v_i}$  for the other vertices equals 0. Therefore, if  $h_{v_i} - h_{v_j} = 1$ , it signifies that the output of layer  $v_i$  must be transmitted to the ground station as an intermediate result. In this scenario, there are four pairs of vertices with  $h_{v_i} - h_{v_j} = 1$ . However, since the preceding vertices of  $v_3, v_4, v_5$  are all  $v_1$ , only one transmission of the output data of  $v_1$  to the ground station is necessary. Therefore, when multiple successor vertices for the same vertices require offloading, it is necessary to divide by the number of edges to avoid redundant computations. Therefore, the transmission cost for completing a DNN task on the satellite can be expressed as:

$$E_{trans} = \sum_{i=1}^I \frac{\sum_{j=i+1}^I (C_{v_i v_j} \cdot (h_{v_i} - h_{v_j}) \cdot E_{trans}^{v_i})}{\sum_{j=i+1}^I (C_{v_i v_j} \cdot (h_{v_i} - h_{v_j}))} \quad (10)$$

To ensure the intermediate results are successfully transmitted to the ground for further computation, the transmitting latency must meet two conditions. Firstly, the energy collected on the satellite after completing the computation task must be sufficient to cover the energy consumption required for transmission. The latency satisfying this condition is denoted as  $t_{trans}^1$ . Secondly, the intermediate data must be successfully transmitted to the ground station. The latency satisfying this condition is denoted as  $t_{trans}^2$ . Therefore, the transmitting latency  $t_{trans}$  can be expressed as:  $t_{trans} = \max(t_{trans}^1, t_{trans}^2)$ .  $t_{trans}^1$  represents the latency that satisfies the equation  $E_{task} = E_{com} + E_{trans}$ , can be expressed as:

$$t_{trans}^1 = \frac{E_{sup} - E_{com} - E_{trans}}{P_{sys}} \quad (11)$$

To satisfy the second condition, we consider scenarios where the satellite must complete all necessary data transmissions over multiple orbital periods based on real conditions. Consequently, the latency of transmitting data from the satellite to the ground can be divided into two parts: the latency of satellite data transmission  $t_{tr}$  and the latency of waiting for data transmission when the ground station is out of contact with the satellite during the operational cycle  $t_{per}$ . Therefore, the latency for transmitting the intermediate data  $D$  from the satellite to the ground can be expressed as follows:

$$\begin{aligned} t_{trans}^2 &= t_{tr} + t_{per} \\ &= \frac{D}{R_{trans}} + t_{cyc} \cdot \left( \left\lceil \frac{D}{R_{trans} \cdot t_{con}} \right\rceil - 1 \right) \end{aligned} \quad (12)$$

where  $t_{cyc}$  represents the satellite orbital period, and  $t_{con}$  denotes the transmission window time between the satellite and the ground.

## 5 PROBLEM SOLUTION

We first proposed an exact solution for the problem by formulating an integer nonlinear program (INLP). Subsequently, we designed a global optimal algorithm to solve the problem and obtain the optimal strategy.

### 5.1 Problem Formulation

Our goal is to minimize energy consumption on satellites for each inference task, including both computational and communication energy. In the satellite-ground co-inference model we have established, it is necessary to determine the values of  $h_{v_i}$ . Therefore, the optimization problem can be formulated as an integer nonlinear program as follows.

$$\begin{aligned} \min E &= E_{com} + E_{trans} = \\ &\sum_{i=1}^I h_{v_i} \cdot E_{com} + \frac{\sum_{j=i+1}^I (C_{v_i v_j} \cdot (h_{v_i} - h_{v_j}) \cdot E_{trans}^{v_i})}{\sum_{j=i+1}^I (C_{v_i v_j} \cdot (h_{v_i} - h_{v_j}))} \end{aligned} \quad (13)$$

s.t.

$$E_{sup}^{bat} \leq \gamma \cdot E_{bat}^{max} \quad (14)$$

$$t_{com}^{satellite} + t_{trans} + t_{com}^{ground} \leq T \quad (15)$$

$$R_{trans} \leq R_{trans}^{max} \quad (16)$$

$$\theta \in [0, \frac{\pi}{2}] \quad (17)$$

$$h_v \in \{0, 1\} \quad (18)$$

where Eq. (14) constrains the depth of discharge of the battery set. Eq. (15) denotes that the overall duration of satellite-ground co-inference must be completed within a specified time delay, where  $T$  represents the maximum allowable task latency. Eq. (16) limits the satellite downlink transmission rate to be less than the theoretical maximum transmission rate. Eq. (17) specifies the range of variation for the solar incidence angle. Eq. (18) restricts the binary variables.

### 5.2 Satellite-Ground Co-Inference Algorithm

The conventional optimization algorithms cannot solve the minimize energy consumption on satellites with multiple constraints. Because commonly used methods for nonlinear programming problems such as interior point methods and gradient descent can only yield local optimal solutions. Therefore, to obtain the globally optimal solution for the offloading strategy, we need to choose a

**Algorithm 2:** Satellite-Ground Co-Inference

**Input:** The DAG of a DNN model. The initial data volume  $D_0$  and the collection of ratios of the output matrices of layers to the initial input matrix  $\{\alpha_v\}$ . The constraints  $C$ . Energy Consumption Prediction Net based on PSO-BPR: *net*. The constants:  $\{\theta\}$ ,  $R_{trans}$ ,  $P_{sun}$  and so on.

**Output:** An offloading decision  $\{h_v\}$  for the inference request.

- 1 **Generate**  $M$  random individuals as the initial population  $P(0)$ ;
- 2 **Define** Probability of Crossover  $p_c$ , and Mutation Occurrence  $p_m$ ;
- 3 **Define** population size  $M$ ;
- 4 **Define** the number of iterations  $t = 0$ ;
- 5  $E_{com}^0 \leftarrow \text{net}(\{\alpha_v\}, D_0, \text{DAG})$ ;
- 6 **while not** (termination criterion) **do**
- 7     **foreach** individual  $i$  in  $M$  **do**
- 8         Evaluate the fitness  $E(i)$  of  $P(t)$ ;
- 9     **end**
- 10      $P(t+1) = \emptyset$ ;
- 11     **while**  $|P(t+1)| < |P(t)|$  **do**
- 12         Select two individuals  $ind_a, ind_b$  in  $P(t)$  with the fitness;
- 13         **if**  $\text{random}(0, 1) < p_c$  **then**
- 14              $ind_c \leftarrow$  perform crossover operations on  $ind_a, ind_b$ ;
- 15         **end**
- 16         **if**  $\text{random}(0, 1) < p_m$  **then**
- 17              $ind_d \leftarrow$  apply mutation operations on  $ind_c$ ;
- 18         **end**
- 19         **if**  $ind_d$  satisfy the constraints  $C$  **then**
- 20              $P(t+1) \leftarrow ind_d$ ;
- 21         **end**
- 22     **end**
- 23      $t \leftarrow t+1$ ;
- 24 **end**

heuristic intelligent algorithm for solving it. We propose the Genetic Algorithm (GA) for the problem. GA is a heuristic optimization technique that solves optimization problems by simulating natural selection and genetic mechanisms in biological evolution. Moreover, GA is the most suitable heuristic algorithm for optimizing problems in discrete spaces and exhibits robustness in searching multi-modal spaces.

The detailed algorithm is provided in Algorithm 2. The satellite-ground co-inference algorithm is solved using Integer Nonlinear Programming based on the Genetic Algorithm (INLPGA). First, initialization is performed (lines 1-4). Next, the predictive network calculates the energy consumption for each DNN layer (line 5). The population then undergoes continuous iterations until the termination condition is met. In each iteration, the energy consumption  $E(i)$  of each individual in the parent population is calculated (lines

**Table 2:** Simulation parameters.

| Parameters                                   | Value                        |
|----------------------------------------------|------------------------------|
| Orbital height $H$                           | 450 km                       |
| Orbital period of the satellite $t_{cyc}$    | 5,601 seconds                |
| Window time $t_{con}$                        | 6 minutes                    |
| Direct solar power $P_{sun}$                 | 40 W                         |
| Satellite system power consumption $P_{sys}$ | 10 W                         |
| Max battery energy capacity $E_{bat}^{max}$  | $8.28 \times 10^5 J$         |
| Max discharge depth                          | 40%                          |
| Max transmission rate $R_{trans}^{max}$      | [20, 100] MB/s               |
| Latency of processing 1MB on ground $\beta$  | $[10^{-6}, 10^{-3}]$ seconds |
| Probability of Crossover $p_c$               | 0.8                          |
| Mutation Occurrence $p_m$                    | 0.01                         |

7-9). The offspring population  $P(t+1)$  is initialized, and individuals are selected using a proportional selection algorithm until  $P(t+1)$  matches  $P(t)$  (lines 10-12). Single-point crossover is performed on two individuals, producing individual  $ind_c$  (lines 13-15). Gaussian mutation is then applied to  $ind_c$ , resulting in individual  $ind_d$  (lines 16-18). If  $ind_d$  satisfies all constraints, it is added to  $P(t+1)$  (lines 19-21). Finally, the value of  $t$  is incremented for the next generation (line 23).

## 6 EVALUATION

In this section, the proposed satellite-ground co-inference algorithm (INLPGA) is analyzed via experiment. We evaluate the performance of our proposed algorithm by comparing it with the following three algorithms:

- **Greedy algorithm (GREEDY):** This approach employs the greedy principle to choose the best option based on current state, aiming to aggregate the final outcomes [25].
- **All tasks are completed on the satellite (ARS):** The satellite autonomously executes the entire DNN task on the on-board payload, then transmits the results to the ground stations [7].
- **All tasks are completed on the ground (ARG):** The satellite transmits all initial data to ground stations [10]. We designate the ARG time as the maximum task delay constraint  $T$  to ensure real-time performance.

### 6.1 Experiment Setup

In the experiment, we utilized the parameters of satellites in a real-world “BUPT-1” LEO satellite. Using STK simulation software<sup>3</sup>, we calculated the solar incidence angle  $\theta$  for each second. Consequently, we computed the power obtained by the solar panel  $P_{sun} \cdot \cos \theta$ , as mentioned in Section IV. The main simulation parameters were set according to [31, 40]. We list the main simulation parameters in Table 2.

### 6.2 The Impact of Different Models

To evaluate the impact of various models, we selected three widely used DNN models: AlexNet, VGG, and GoogLeNet. These models,

<sup>3</sup><https://www.ansys.com/products/missions/ansys-stk>

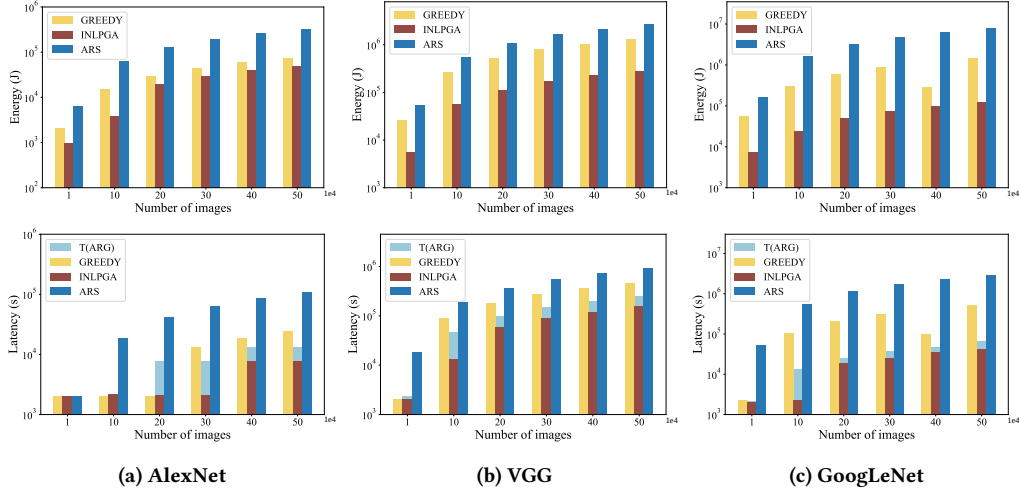


Figure 10: Comparison of energy consumption and latency.

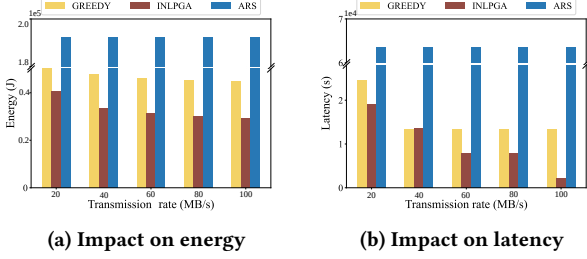


Figure 11: The impact of different downlink transmission rates.

differing in structure and complexity, help ensure convincing results. AlexNet, the simplest with the fewest layers and parameters, contrasts with the more complex structures of VGG and GoogLeNet. From Figure 10, it can be observed that the more complex the DNN structure, the more energy consumption and latency for task completion. We set the number of images varies from  $1 \times 10^4$  to  $5 \times 10^5$  with the transmission rate of 100MB/s. For AlexNet, using the algorithm proposed in this paper, the energy consumption ranges from 977 J to 48,880 J. For VGG, the energy consumption ranges from 5,616 J to 280,817 J. For GoogLeNet, the energy consumption ranges from 7,327 J to 122,157 Js. In terms of energy consumption, both VGG and GoogLeNet show an increase of one order of magnitude compared to AlexNet overall. In terms of latency, VGG, and GoogLeNet show an increase of two orders of magnitude and one order of magnitude respectively compared to AlexNet. Moreover, We observed that the energy consumption of our proposed algorithm INLPGA consistently remains the lowest among the three algorithms. It validates the remarkable effectiveness and robustness of our proposed algorithm.

### 6.3 The Impact of Data Volume

To investigate the impact of data volume, Figure 10 illustrates the performance of energy consumption and latency across the three

models with the number of images varying from  $1 \times 10^4$  to  $5 \times 10^5$ . Both metrics exhibit an increasing trend with the increase in data volume. When computing data for  $5 \times 10^5$  images, Figure 10b shows that with the VGG model, the ARS algorithm consumes 8.6× more energy and experiences a latency 4.9× longer compared to the INLPGA. Figure 10c shows that with the GoogLeNet model, the ARS algorithm consumes 65× more energy and experiences a latency 66× longer compared to the INLPGA. We conclude that the proposed algorithm is more suitable for scenarios involving large volumes of computational data.

### 6.4 The Impact of Transmission Rate

Figure 11 further illustrates the impact of transmission rates on energy consumption and latency. We conducted tests on the AlexNet model with data volumes of  $3 \times 10^5$  images while varying the satellite's downlink rates.

Figure 11a illustrates the impact of satellite downlink transmission rates on energy consumption. We observed that increasing the rate from 20MB/s to 100MB/s results in only a slight decrease in energy consumption, by 38%. Moreover, under different transmission rates, the energy consumption of our algorithm consistently remains lower than other algorithms. This finding indicates that, even under poor satellite-ground communication conditions, using the proposed algorithm can still save energy consumption on the satellite. Figure 11b illustrates the impact of satellite downlink transmission rates on latency. We concluded that increasing the rate from 20MB/s to 100MB/s results in a significant decrease in latency, by 8.9×. Additionally, from Figure 11b, we observed that under different transmission rates, the energy consumption of our algorithm consistently remains much lower than that of the ARG algorithm, demonstrating the superiority of decreasing latency.

## 7 CONCLUSION

This paper addresses the challenges of limited energy acquisition on LEO satellites by providing DNN-inherent service via satellite-ground co-inference. We first conduct preliminary measurements

to explore the correlation between energy consumption and layer types, establishing a layer-based energy consumption prediction model. Next, we formulate and solve a DNN layer-wise processing schedule optimization problem using a genetic algorithm. Using configurations from the real-world in-orbit satellite "BUPT-1," our simulations demonstrate the effectiveness of our algorithm, achieving an average energy savings of 27×. For future work, we plan to deploy our method on real satellites for further experimentation.

## ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under grants U21B2016, 62032003, 62372061, 62302055, 62302262, and the Fundamental Research Funds for the Central Universities.

## REFERENCES

- [1] Patrick Bernard and Michael Lippert. 2015. Nickel-cadmium and nickel-metal hydride battery energy storage. *Electrochemical energy storage for renewable sources and grid balancing* (2015), 223–251.
- [2] Debopam Bhattacharjee, Simon Kassing, Melissa Licciardello, and Ankit Singla. 2020. In-orbit computing: An outlandish thought experiment? *Proceedings of the 19th ACM Workshop on Hot Topics in Networks* (2020), 197–204.
- [3] Teresa M Braun. 2012. Satellite Communications payload and system. (2012).
- [4] Qi Chen, Zhigang Liu, Xiaofeng Zhang, and Liying Zhu. 2020. *Spacecraft Power System Technologies*.
- [5] Yijie Chen, Qiyang Zhang, Yiran Zhang, Xiao Ma, and Ao Zhou. 2023. Energy and Time-Aware Inference Offloading for DNN-based Applications in LEO Satellites. *2023 IEEE 31st International Conference on Network Protocols (ICNP)* (2023), 1–6.
- [6] Inigo Del Portillo, Bruce G Cameron, and Edward F Crawley. 2019. A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. *Acta astronautica* 159 (2019), 123–135.
- [7] Bradley Denby, Krishna Chintalapudi, Ranveer Chandra, Brandon Lucia, and Shadi Noghbi. 2023. Kodan: Addressing the computational bottleneck in space. *Proceedings of the 28th International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3* (2023), 392–403.
- [8] Bradley Denby and Brandon Lucia. 2019. Orbital edge computing: Machine inference in space. *IEEE Computer Architecture Letters* 18, 1 (2019), 59–62.
- [9] Bradley Denby and Brandon Lucia. 2020. Orbital edge computing: Nanosatellite constellations as a new class of computer system. *Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems* (2020), 939–954.
- [10] Daniel Fischer, David Basin, Knut Eckstein, and Thomas Engel. 2012. Predictable mobile routing for spacecraft networks. *IEEE Transactions on Mobile Computing* 12, 6 (2012), 1174–1187.
- [11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
- [12] Chuang Hu, Wei Bao, Dan Wang, and Fengming Liu. 2019. Dynamic adaptive DNN surgery for inference acceleration on the edge. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications* (2019), 1423–1431.
- [13] Hyuk-Jin Jeong, Hyeon-Jae Lee, Chang Hyun Shin, and Soo-Mook Moon. 2018. IONN: Incremental offloading of neural network computations from mobile devices to edge servers. *Proceedings of the ACM symposium on cloud computing* (2018), 401–411.
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).
- [15] Zeqi Lai, Hewu Li, Qian Wu, Qiang Ni, Mingyang Lv, Jihao Li, Jianping Wu, Jun Liu, and Yuanjie Li. 2022. Futuristic 6G pervasive on-demand services: Integrating space edge computing with terrestrial networks. *IEEE Vehicular Technology Magazine* 18, 1 (2022), 80–90.
- [16] Qing Li, Shangguang Wang, Xiao Ma, Ao Zhou, and Fangchun Yang. 2021. Towards Sustainable Satellite Edge Computing. *2021 IEEE International Conference on Edge Computing (EDGE)* (2021), 1–8.
- [17] Yuanjie Li, Hewu Li, Wei Liu, Lixin Liu, Wei Zhao, Yimei Chen, Jianping Wu, Qian Wu, Jun Liu, Zeqi Lai, et al. 2023. A networking perspective on starlink's self-driving leo mega-constellation. *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking* (2023), 1–16.
- [18] Huanghuang Liang, Qianlong Sang, Chuang Hu, Dazhao Cheng, Xiaobo Zhou, Dan Wang, Wei Bao, and Yu Wang. 2023. DNN surgery: Accelerating DNN inference on the edge through layer partitioning. *IEEE transactions on Cloud Computing* (2023).
- [19] Timothy M Lim, Aaron M Cramer, James E Lumpp, and Samir A Rawashdeh. 2018. A modular electrical power system architecture for small spacecraft. *IEEE Trans. Aerospace Electron. Systems* 54, 4 (2018), 1832–1849.
- [20] Wenyu Liu, Gaofeng Ren, Runsheng Yu, Shi Guo, Jianke Zhu, and Lei Zhang. 2022. Image-adaptive YOLO for object detection in adverse weather conditions. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 2 (2022), 1792–1800.
- [21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [22] Pedro Pereira, Bruno J Guerreiro, and Pedro Lourenço. 2022. Distributed model predictive control method for spacecraft formation flying in a leader-follower formation. *IEEE Trans. Aerospace Electron. Systems* (2022).
- [23] Jaan Praks, M Rizwan Mughal, R Vainio, P Janhunen, J Envall, P Oleynik, Antti Näsälä, H Leppinen, P Niemelä, A Slavinskis, et al. 2021. Aalto-1, multi-payload CubeSat: Design, integration and launch. *Acta Astronautica* 187 (2021), 370–383.
- [24] HS Rauschenbach. 2012. Solar cell array design handbook: The principles and technology of photovoltaic energy conversion. *NASA STI/Recon Technical Report A* 80 (2012), 34847.
- [25] Ergys Ristani and Carlo Tomasi. 2018. Features for multi-target multi-camera tracking and re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 6036–6046.
- [26] Crefeda Faviola Rodrigues, Graham Riley, and Mikel Luján. 2018. SyNERGY: An energy measurement and prediction framework for Convolutional Neural Networks on Jetson TX1. *Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA)* (2018), 375–382.
- [27] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.
- [29] Wei-Chi Tan, Jingang Hu, et al. 2009. Spacecraft Systems Engineering. (2009).
- [30] Xiaolong Tu, Anik Mallik, Dawei Chen, Kyungtae Han, Onur Altintas, Haoxin Wang, and Jiang Xie. 2023. Unveiling Energy Efficiency in Deep Learning: Measurement, Prediction, and Scoring across Edge Devices. *2023 IEEE/ACM Symposium on Edge Computing (SEC)*, 80–93.
- [31] Ruolin Xing, Mengwei Xu, Ao Zhou, Qing Li, Yiran Zhang, Feng Qian, and Shangguang Wang. 2024. Deciphering the enigma of satellite computing with cots devices: Measurement and analysis. *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (2024).
- [32] Zichuan Xu, Liqian Zhao, Weifa Liang, Omer F Rana, Pan Zhou, Qiufen Xia, Wenzheng Xu, and Guowei Wu. 2020. Energy-aware inference offloading for DNN-driven applications in mobile edge clouds. *IEEE Transactions on Parallel and Distributed Systems* 32, 4 (2020), 799–814.
- [33] Yuan Yang, Mingwei Xu, Dan Wang, and Yu Wang. 2016. Towards energy-efficient routing in satellite networks. *IEEE Journal on Selected Areas in Communications* 34, 12 (2016), 3869–3886.
- [34] Rongjie Yi, Ting Cao, Ao Zhou, Xiao Ma, Shangguang Wang, and Mengwei Xu. 2023. Boosting DNN Cold Inference on Devices. *The 21st International Conference on Mobile Systems, Applications, and Services* (2023).
- [35] Kegen Yu, Yunwei Li, and Xin Chang. 2018. Snow depth estimation based on combination of pseudorange and carrier phase of GNSS dual-frequency signals. *IEEE Transactions on Geoscience and Remote Sensing* 57, 3 (2018), 1817–1828.
- [36] Li Lyna Zhang, Shihao Han, Jianyu Wei, Ningxin Zheng, Ting Cao, Yuqing Yang, and Yunxin Liu. 2021. Nn-meter: Towards accurate latency prediction of deep-learning model inference on diverse edge devices. *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (2021), 81–93.
- [37] Qiyang Zhang, Xiangying Che, Yijie Chen, Xiao Ma, Mengwei Xu, Schahram Dustdar, Xuanzhe Liu, and Shangguang Wang. 2023. A Comprehensive Deep Learning Library Benchmark and Optimal Library Selection. *IEEE Transactions on Mobile Computing* (2023).
- [38] Qiyang Zhang, Xiang Li, Xiangying Che, Xiao Ma, Ao Zhou, Mengwei Xu, Shangguang Wang, Yun Ma, and Xuanzhe Liu. 2022. A comprehensive benchmark of deep learning libraries on mobile devices. *Proceedings of the ACM Web Conference 2022* (2022), 3298–3307.
- [39] Qiyang Zhang, Xin Yuan, Ruolin Xing, Yiran Zhang, Zimu Zheng, Xiao Ma, Mengwei Xu, Schahram Dustdar, and Shangguang Wang. 2024. Resource-efficient In-orbit Detection of Earth Objects. *IEEE INFOCOM 2024-IEEE conference on computer communications* (2024).
- [40] Tongxin Zhu, Jianzhong Li, Zhipeng Cai, Yingshu Li, and Hong Gao. 2020. Computation scheduling for wireless powered mobile edge computing networks. *IEEE INFOCOM 2020-IEEE Conference on Computer Communications* (2020), 596–605.