# Adaptive Offloading of Transformer Inference for Weak Edge Devices with Masked Autoencoders

TAO LIU and PENG LI, The University of Aizu, Japan
YU GU, Hefei University of Technology, China
PENG LIU, Hangzhou Dianzi University, China
HAO WANG, School of Cyber Engineering, Xidian University, China

Transformer is a popular machine learning model used by many intelligent applications in smart cities. However, it has high computational complexity and it would be hard to deploy it in weak-edge devices. This paper presents a novel two-round offloading scheme, called A-MOT, for efficient transformer inference. A-MOT only samples a small part of image data and sends it to edge servers, with negligible computational overhead at edge devices. The image is recovered by the server with the masked autoencoder (MAE) before the inference. In addition, an SLO-adaptive module is intended to achieve personalized transmission and effective bandwidth utilization. To avoid the large overhead on the repeat inference in the second round, A-MOT further contains a lightweight inference module to save inference time in the second round. Extensive experiments have been conducted to verify the effectiveness of the A-MOT.

CCS Concepts: • **Computing methodologies** → **Cooperation and coordination**; • **Networks** → **Wireless local area networks**; • **Human-centered computing** → *Empirical studies in ubiquitous and mobile computing*; • **Computer systems organization** → *Sensor networks.*

Additional Key Words and Phrases: sensor, edge, inference, MAE

## 1 INTRODUCTION

There is a strong demand to deploy intelligent applications, e.g., object detection [14], data augmentation [45], and image recognition [9, 10], on mobile/IoT devices with various sensors. These applications are based on the inference operations of complex deep neural networks (DNN), which can hardly run on devices with limited hardware resources. This dilemma motivates broad-spectrum research on offloading DNN inference operations to edge servers or clouds. An ideal offloading policy should satisfy three requirements: (1) high inference accuracy; (2) low communication overhead, as mobile/IoT devices using wireless networks usually have limited network bandwidth; and (3) low computation overhead on the device side [5, 17].

Unfortunately, none of the existing work can achieve all three requirements at the same time. A straightforward offloading strategy sends raw data to the cloud [21], which eliminates computation at edge devices and achieves high inference accuracy by using powerful hardware in the cloud. However, since raw data have a large size, this simple strategy would incur a high communication cost. Some recent works have proposed data preprocessing techniques at edge devices to reduce communication costs. Such preprocessing techniques include DNN model splitting [22], input data compression [25], and input data filtering [6]. DNN model splitting is based on the

Authors' addresses: Tao Liu, d8212107@u-aizu.ac.jp; Peng Li, pengli@u-aizu.ac.jp, The University of Aizu, AizuWakamatsu, Fukushima, Japan, 965-8580; Yu Gu, yugu.bruce@ieee.org, Hefei University of Technology, Hefei, China, yugu.bruce@ieee.org; Peng Liu, perrypliu@gmail.com, Hangzhou Dianzi University, Hangzhou, China; Hao Wang, haow@ieee.org, School of Cyber Engineering, Xidian University, Xi'an, China.


ACM 1550-4859/2024/1-ART
https://doi.org/10.1145/3639824

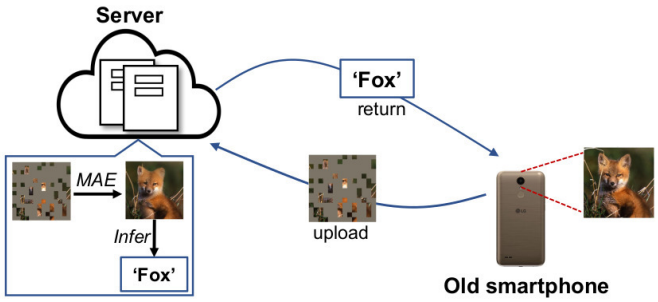



Fig. 1. The basic MAE-based offloading scheme.

observation that the output of some intermediate layers is smaller than the original input. Thus, we can trade running a few DNN layers at edge devices for a significant reduction in communication cost. However, running just a few DNN layers could be also a heavy burden for weak-edge devices. Furthermore, not all DNN models exhibit the feature of smaller intermediate data. Some DNN-based data compression methods also have a heavy computation overhead for edge devices [25]. A higher compression ratio can reduce the amount of data transmitted over networks, however, the inference accuracy could also be decreased if the data is over-compressed. Data filtering techniques select and send image regions, including target objects, instead of whole images. The commonly used MobileNet-SSD [13] filter demands about 1200 Million Floating Point Operations (MFLOPs) for $300 \times 300$ images [38], while some edge devices, like raspberry pi-zero-w and raspberry pi-aplus, only support about 200 Million Floating-point Operations per Second (MFLOPS) [36]. Du et al. [5] have proposed a server-driven offloading method for video analysis. Edge devices transmit low-quality frames with reduced size to the server first, and then the server identifies target regions and requests a re-transmission of high-quality content within these regions.

There are additional works that formulate and resolve various offloading optimization problems. The issue of privacy is considered during model splitting in [32] and the energy consumption constraint is added in [42]. The deep Q-network-based offloading strategies are also proposed with consideration of channel conditions [27, 29]. and reinforcement learning is used to let each device make its own offloading decision [47]. However, these methods are tailored to specific tasks and lack a holistic perspective that simultaneously considers computation, communication, and accuracy.

In this paper, we propose a new approach to breaking the myth of the impossible trinity of DNN offloading. The basic idea is shown in Fig. 1. Edge device (e.g., an old smartphone) collects image data and randomly samples a small portion of image patches, and sends them to the server, which then uses a masked autoencoder (MAE) [12] to recover the image and conduct inference. Sampling is a simple operation with negligible computation overhead for edge devices. Since the sampling ratio could be very low, usually less than 30%, only a small amount of data need to be transmitted over networks, leading to low communication cost. MAE was originally designed for pre-training, and we exploit its powerful capability in image recovery for DNN offloading. Therefore, it is promising to achieve high inference accuracy with limited sampled data. Note that our method is orthogonal to conventional compression methods that encode image data using various transform strategies. The sampled data can also be further compressed by these methods.

Although the MAE-based scheme shown in Fig. 1 is promising, we are facing several critical challenges to make it work efficiently in practice. The first is to determine how many patches should be sampled to guarantee a good recovery with high inference accuracy. More patches could be helpful for better image recovery while leading to higher communication costs. Especially, weak edge devices have insufficient hardware resources to

run complex algorithms for content recognition to make decisions. We address this challenge by designing a two-round offloading scheme for inference, named A-MOT (Adaptive MAE-based Offloading for Transformer inference). A-MOT contains an image selection process. Specifically, in the first round, edge devices randomly sample a small number of patches and send them to the server. If these patches are sufficient for recovery and obtaining inference results with high confidence, the server returns results and completes the inference service. Otherwise, several “important" patches are selected and requested by the server in the second round of offloading.

Second, we find that different images require different numbers of patches for correct inference. Some images with simple contents can be well recovered by MAE even with a few patches, but more patches are needed for complex images. Offloading efficiency could be further improved if this feature is well exploited. Since both edge devices and the server are unaware of image contents before the first round of offloading, we let all devices offload the same amount of patches. In the second round of offloading, the server requests different amounts of patches for images, by using the information obtained by MAE and inference operation. However, this method is agnostic to SLO (service level objective), i.e., it determines the number of patches without considering network bandwidth. Thus, A-MOT has an SLO-adaptive design that can decide how many patches are transmitted in the second round of offloading for different images, given a traffic budget.

The final challenge is the high computational burden on the server. Although the two-round offloading scheme is promising in terms of reducing communication costs and increasing inference accuracy, the server has high computational overhead because it needs to run two inference operations for some images that need the second round of offloading. The commonly used inference models are Vision Transformer (ViT)-based, which divides an image into multiple small patches to form a patch sequence. The most significant component of the models is the attention layer, where they calculate the attention value among each patch pair to generate new embeddings. The overhead of inference is thus positively related to the length of the patch sequence. The optimization method for language inference with the Transformer model in [7] does not work here since it takes advantage of the fact that language sentence lengths are naturally different, while images from the same device have the same size. To reduce the overhead, A-MOT has a lightweight inference operation for the second round of offloading. Instead of running a full inference with the complete patch sequence, the server lets newly received patches go through an encoder, which has the same attention layer as the inference model. This operation has low overhead because of the short input. Then, the embeddings generated by this encoder are combined with the ones from the first round. The combined results are sent to a decoder to generate the final output.

The main contributions of this paper are summarized as follows:

- We propose a two-round inference offloading scheme based on MAE so that weak edge devices can also achieve high inference accuracy with low communication costs.
- We design an SLO-adaptive strategy to maximize the inference accuracy with the constraint of limited network bandwidth.
- We reduce the computational cost at the server by proposing a lightweight inference operation for the second-round offloading.

The rest of this paper is organized as follows. We introduce the background and motivation in section 2 and section 3. The system design is described in section 4. We evaluate the system in section 5 with various baselines. Section 6 discusses some related works, and Section 7 is the conclusion.

## 2 BACKGROUND

### 2.1 Transformer and ViT

The Transformer has been proposed as a self-attention-based model that can effectively handle various learning tasks related to natural language processing (NLP) [20, 30, 35]. Given a sentence whose words can be expressed

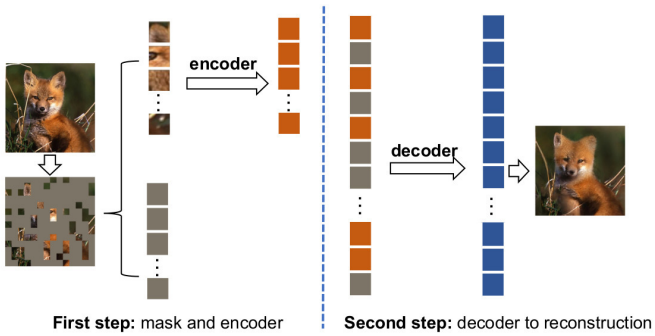



Fig. 2. The process of MAE.

as feature embeddings $X = [x_0, x_1, ..., x_n] \in \mathbb{R}^d$, the core self-attention operation can be described as follows:

$$Q_i = x_i W^Q,\ K_i = x_i W^K,\ V_i = x_i W^V. \qquad \forall x_i \in X \tag{1}$$

$$A_{i,j} = \frac{Q^i K^j}{\sqrt{d}},\ x_i^{'} = (\sum_{j=0}^{n} softmax(A_{i,j}) V_j). \qquad \forall x_j \in X \tag{2}$$

The $W^Q, W^K, W^V \in \mathbb{R}^d$ in Equation 1 are trainable weights that transform $X$ into query, key, and value matrices, respectively. $A_{i,j}$ is the normalized weight of value for $x_j$. The newly generated hidden embedding $x_i^{'}$ is a weighted sum based on all values. A complete attention layer contains multiple self-attention operations (multi-heads), and a Transformer model contains multiple attention layers.

Inspired by Transformer's successes in NLP, Vision Transformer (ViT) has been proposed to deal with image data [4]. Concretely, an image is divided into different patches, where each patch's size is fixed as $16 \times 16$ since it brings good performance. All patches go through a convolutional layer to generate their initial feature embeddings. After that, they are formatted as a sequence with position embedding. An additional classification patch is added to the beginning of a recognition task's sequence. The self-attention mechanism described above is applied to the sequence of patches to generate a high-level representation for each patch. The classification patch can therefore aggregate embedding data from all other patches based on the attention and output of the final recognition result.

## 2.2 Masked Autoencoders (MAE)

Training ViT requires a vast quantity of labeled data and computational resources. However, many image data have no labels. Masked Autoencoders (MAE) has been proposed for pre-training ViT model with unlabeled images [12]. Similar to the NLP pre-training that masks words in a sentence and utilizes the Transformer to infer[20], MAE also constructs a self-supervised task that masks image patches and then infers them using the ViT-based autoencoders.

As shown in Fig. 2, MAE is comprised of two steps. First, it masks some patches of an image and sends the rest to an encoder to generate high-level embeddings. Second, we combine these embeddings and the masked patches, which are then sent to a decoder to recover the original images. The masked patches are learnable vectors here since we do not know their content. The combined patch sequence goes through a decoder to infer the original pixel values of the masked patches. The mean squared errors of the inferred values and the real values are used as the loss for the backward propagation. The decoder is a tiny ViT with a limited number of attention layers. In

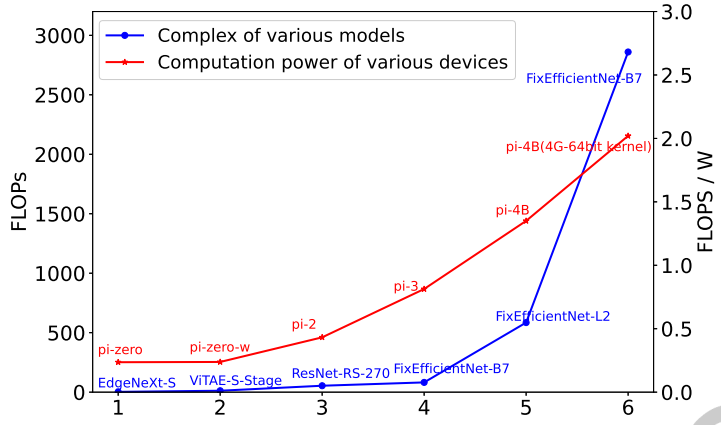



Fig. 3. The development of models' complexity and devices computation power per energy unit.



contrast to the average pixel values of its neighbors, a large mask ratio can assist MAE in learning genuine visual semantics for reconstruction. The self-supervised trained encoder can be further fine-tuned based on the labeled images to obtain the final parameter for the classification task.

## 3 MOTIVATION

### 3.1 Limited Resources of Edge Devices

The popular deep neural networks are vital in dealing with data analysis [51]. However, edge devices could be rather resource-limited [5, 17], making it hard to run inference tasks with complex models. Fig. 3 shows the FLOPs of several commonly used models for image recognition and the FLOPS/W of the various raspberry pi devices [36]. The growth of computing capability on edge devices lags behind the increase in model complexity. For example, the computing power of *pi-4B(4G-64bit)* is about 2.0 FLOPS/W, which is around four times that of *pi-zero*, while the FLOPs of the *FixEfficientNet-B7* are about 2700, which is thousands of times that of the *edgeNetx-S*. This suggests that edge devices have struggled to efficiently run growing models. Offloading has therefore been widely exploited for inference tasks. However, bandwidth is a scarce and even volatile resource [22, 31, 33]. The direct transmission of raw data may incur a significant delay. To reduce offloading traffic, some works use small-size selectors to choose and transmit critical regions in the image. However, these selectors are still resource-intensive for resource-limited edge devices. There are also efforts to run a part of the inference model on edge devices and upload the intermediate output to the server for the rest of the inference [22]. These DNN partition-oriented works hypothesize that some intermediate layers in the neural network may have a smaller output size than the raw data, thus saving bandwidth. However, lots of models do not show this characteristic. We list four models with the highest accuracy on the commonly used ImageNet dataset, including CoCa [46], BASIC-L [41], ViT-G/14 [41], and ViT-e [1]. These models are all attention-based with an isotropic architecture [11], which means all main layers contain heavy attention operations and have the same output size. Figure 4 shows the normalized output sizes (with the size of the input image set to one) of intermediate layers for each model. Coca, ViT-G/14, and ViT-e have large intermediate data. For BASIC-L, the size is similar to the input image. However, this output size is obtained after running more than 40 layers, and such a computational burden cannot be afforded by weak-edge devices.

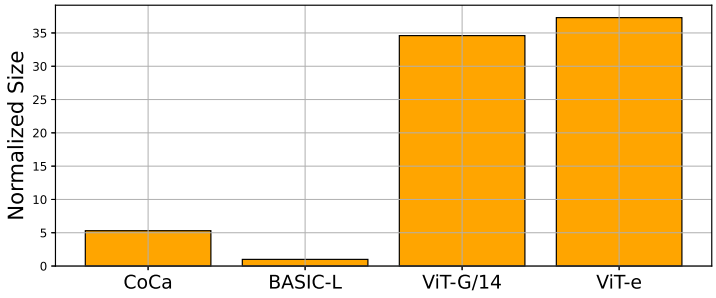



Fig. 4. The output size of various state-of-the-art inference models.

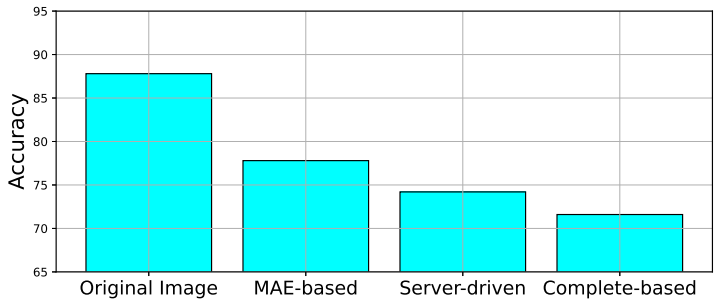
### 3.2 Possibility of MAE-based Bandwidth-saving

Due to MAE's potent reconstruction capabilities, it is reasonable for the device to randomly mask images before transmitting the preserved data to the server for MAE reconstruction and inference. Two related competitors exist. The first option is to replace the MAE with other ways to reconstruct the image, such as the complete-based method in [3]. The other server-driven method [5] initially reduces image resolution on the device and then utilizes server-side computing to identify the target object in the image before retransmitting high-level pixels.

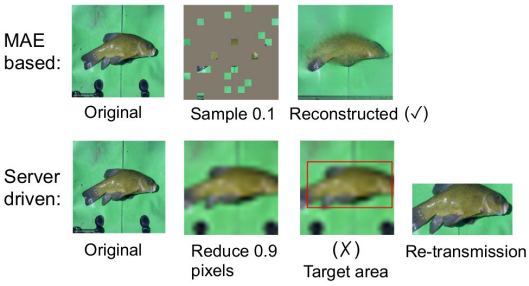
We conducted some preliminary experiments to compare the efforts of different solutions. We employ a subset of the ImageNet dataset for accuracy testing with 10 classes and 50 images for each. We apply the Large-ViT-based encoder in MAE and the DeiT-Small inference model (a variant of ViT) [34]. To conserve bandwidth, all three systems transport only 30% of the original data volume in total. For the server-driven method, we only retransmit part of the target area and prioritize high-attention patches to match the budget constraint. Figure. 5(a) illustrates their accuracy. The MAE-based method has the best performance among all bandwidth-saving solutions. For the complete-based works, they target filling reasonable content with photorealistic appearance into the missing regions [3, 49]. The target is different from our reconstruction process and will generate unrelated content when we have a large mask ratio. For the server-driven work, we give an example as shown in Fig.5(b). With a sample ratio of 0.1, detecting the fish in the masked image is difficult, but reconstruction makes it simpler, and the reconstructed image can be accurately identified. The server-driven method, on the other hand, must retransmit the target region, which is greater than the 0.1 transmission ratio in the MAE-based method. The comparison verifies that MAE may study the deep semantics of a masked image and be utilized for bandwidth savings in offloading. Note that there is a tradeoff between communication and accuracy in our scenario. However, it is difficult to mathematically formulate the relationship between increased communication and improved accuracy since we have an image recovery process.

### 3.3 Different Images Require Various Mask Ratios.

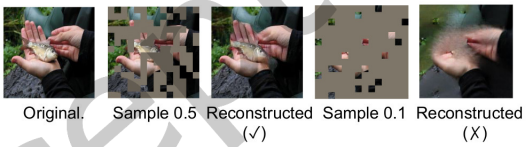
Although the MAE-based method has the best performance among all bandwidth-saving solutions, the challenge, however, is that the MAE-based recovery method still has a significant accuracy gap in comparison to the raw image in Fig. 5(c). Allowing each image to have its optimal sampling ratio for a given transmission budget is a potential method for further enhancing accuracy. We investigated the MAE-based method using additional image instances. As depicted in Fig. 5(c), the image has a complex background, and the object fish in the image is quite small, which cannot be identified with a sample ratio of 0.1 like in Fig. 5(b). When we send more data to bring the mask ratio up to 0.5, the reconstructed image is much clearer, and the fish can be correctly identified. The results show that since the contents of different images are different, their mask ratios should be varied to




(a) The accuracy of different solutions




(b) An image example between two solutions




(c) Complex image demands more data for reconstruction

Fig. 5. MAE can be used for bandwidth-saving in offloading

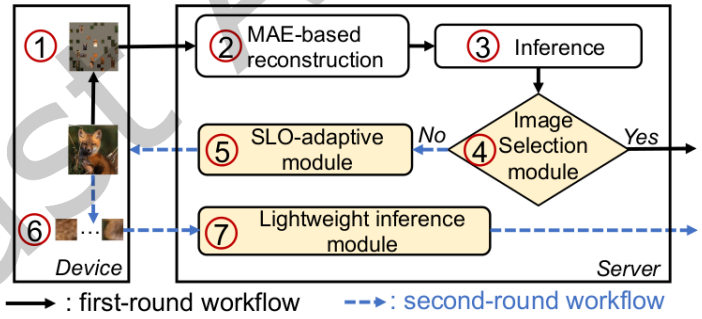
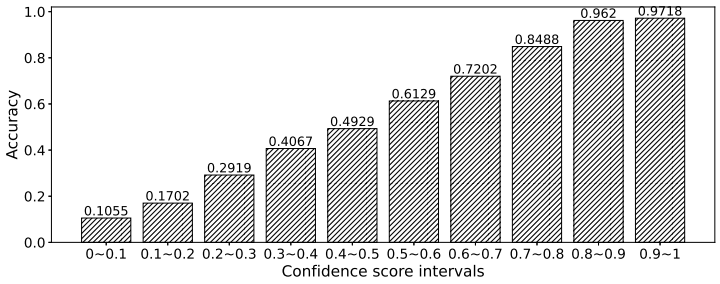
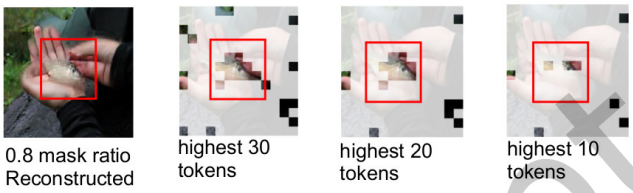



Fig. 6. System overview.

achieve content-aware transmission and bandwidth savings. However, due to limited processing resources on edge devices, we cannot know the best mask ratio for each image, and giving all images the same sample ratio results in duplicate transmission and reduced accuracy.




(a) The relationship between confidence scores and accuracy




(b) High-attention regions in the image are focused on the target object

Fig. 7. Some crucial characteristics in the feedback process.



## 4 SYSTEM DESIGN

### 4.1 Overview

Fig. 6 illustrates the process of the proposed two-round offloading system, A-MOT. There are three key designs in the A-MOT, including the image selection module, the SLO-adaptive module, and the lightweight inference module. Specifically, the device randomly samples the data on each image (①) and transmits it to the server. The server then reconstructs the images using the MAE model (②) and conducts inference (③). The confidence score is compared with a threshold in the image selection module. If the score exceeds the threshold, the inference result is direct output; otherwise, high-attention tokens are chosen to prepare for the second round of offloading (④). The SLO-adaptive module (⑤) then determines and requests additional data volume for each image within the given traffic budget adaptively. The short extra-transmitted data sequence (⑥) will be encoded (⑦) independently in the lightweight inference module, and we combine the output of the encoder and the embeddings of the first inferences to send to an attention-based decoder and obtain the final results. We present the details in the following.

### 4.2 Two-round Offloading with Image Selection

The MAE model can help to recover the sampled data transmitted by the device in order to increase accuracy. It is critical to decide the sample ratios on the device side for each image since the ratio determines the communication costs and the recovery effect. However, weak edge devices have no sufficient hardware resources to run complex algorithms for content recognition to make decisions, and a one-time transmission with a content-random sample operation is not enough to obtain the desired accuracy. We choose to present a two-round offloading scheme. The scheme transmits a low ratio of data for all images first. If these patches are sufficient for recovery and inference, the server returns results and completes the inference service. Otherwise, a second round of offloading is needed.

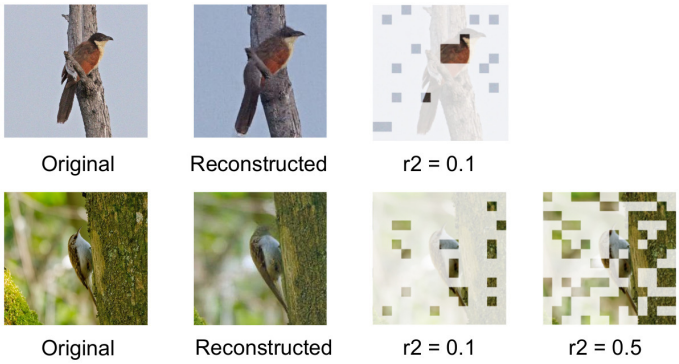



Fig. 8. Different images demand various $r_2$ for the second transmission.



However, it is not a trivial issue to decide whether the second offloading is needed since we cannot know if the image is correctly recognized.

The image selection module is proposed to determine whether the second offloading is needed by exploring the potential of confidence scores. To study the correlation between confidence score and precision, we use the same settings introduced in the previous section. We divide all image samples into ten groups based on their inference confidence scores. The accuracy of the samples at each interval is recorded as in Fig. 7(a). We find that the confidence score correlates highly with accuracy. A similar observation is also made in [8]. Consequently, we can design a threshold-based strategy to focus on the additional data-required images based on the confidence score. The threshold is previously determined by the expected level of precision since it has a direct relationship with accuracy. If the confidence score exceeds the threshold, we finish the inference by generating the result and sending it back to the device; otherwise, the server will demand a second transfer to run the inference process with supplemented data to improve accuracy.

The selection module also selects the contents of the image for the second transmission. Instead of random content chosen, we find that the attention results of the initial inference are related to the image's content. An example is shown in Fig. 7(b). After inferring the reconstructed image with a 0.8 mask ratio, we choose the 30/20/10 tokens that have the highest attention values on the classification token. Note that there are multiple attention heads in the ViT; we average the attention values of all heads in the last attention layer as the final value. We can see that these tokens are most focused on the target object, which is in the red square. Because of this, we can send the high-attention tokens during the second transmission to improve accuracy. Note that two-round offloading is usually enough to obtain good performance with the given communication budget. If we increase the rounds without limitation, the performance may even decrease since it will be complex to decide the bandwidth budget allocation among rounds, and the increased inference operation will bring an extra computation burden to the server.

### 4.3 SLO-adaptive Module

Different images require different numbers of patches for correct inference. Assume $r_1$ and $r_2$ are the transmission ratios compared to the original image size for the two offloading rounds mentioned above. We set all images to have the same $r_1$ since we were unaware of their contents at first. However, $r_2$ should be varied since all images have their own unique content. Fig. 8 shows an example where two reconstructed images have the same $r_2$ as 0.1. The upper image has a simple background, and the re-transmitted content can catch the central part of the target

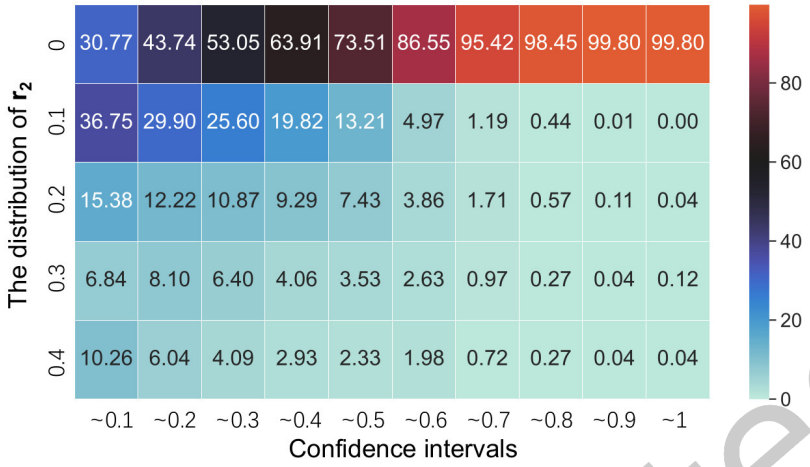



Fig. 9. The distributions of $r_2$ in each confidence score interval.

bird, whereas the other image with a complex background requires a larger $r_2$ of 0.5 to cover the target. In fact, the traffic budget can be limited and constant, so we need to require different images with various amounts of data and achieve the SLO (service level objective) by taking the given traffic budget into account.

We propose the SLO-adaptive module to overcome the challenge by exploring the relationship between the confidence score and the needed $r_2$. We first obtained the smallest $r_2$ values for each image that could be correctly identified by running the inference multiple times. The distributions of $r_2$ in various confidence score intervals are shown in Fig. 9 when the first-time transmission ratio $r_1$ is set as 0.6. We can see that a higher confidence score means a small and more concentrated distribution of $r_2$. This distribution is consistent among all images according to our experiments, which can be used as a priori knowledge.

The above distribution can be represented as $p_{i,j}$, which means the percentage in the $i-th$ raw and $j-th$ interval. Note that $\sum_{i=0}^{R} P_{i,j} = 1$, where $R$ is the number of selections of $r_2$. Since the only information we know is the confidence score of the images, we give the images in the same confidence interval with the same $r_2$. Then we need to decide various $r_2$ for each interval. With the distribution information, we decide $r_2$ by solving a resource allocation problem. We formulate the problem first. Suppose the total transmission budget is $B$ and we have a given first-time transmission ratio $r_1$. The $N$ images in a batch will be naturally distributed in various confidence intervals after the first-time inference. For the $j-th$ interval, the amount of samples is denoted as $d_j$. Suppose $x_j$ is the second-time transmission ratio ($r_2$) for the $j-th$ intervals. We have the following formulations:

$$maximize \sum_{j=0}^{K} \sum_{i=0}^{x_j} p_{i,j}; \tag{3}$$

$$\sum_{j=0}^{K} x_j \cdot d_j \leq B - r_1 \cdot N \tag{4}$$

Our target is to maximize the accuracy, which means maximizing the sum of the probabilities that the images can be correctly classified in each interval as formulated in (3). $K$ is the number of intervals. The transmission budget constraint is given in (4).

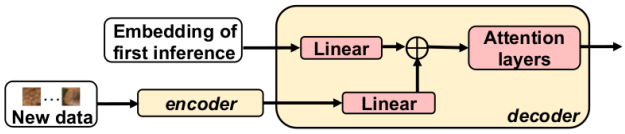



Fig. 10. The process of the lightweight inference.

### 4.4 Lightweight Inference Module

The newly transmitted data contains important information about the image; however, it will be time-consuming if we conduct the reconstruction process again or implement another complete inference operation with the augmented image. Here we propose the lightweight inference module to reduce the computational cost at the server as shown in Fig. 10. First, the data sequence for the second round of offloading is independently encoded by the encoder, and the encoder is configured with the same attention layers as the initial inference model. The overhead of the encoder is low due to the short input sequence. However, the accuracy will be decreased if we only use the new, short data. Then, the lightweight inference module combines the output of the encoder and the embeddings of the first inference. Specifically, they go through the linear layers independently, and the feature values corresponding to the same positions in the image will be added together to generate a new embedding sequence. Finally, the new sequence goes through an attention-based decoder to obtain the final classification result.

The overhead of the second-time inference is proportional to the length of the newly transmitted data and the complexity of the additional decoder. The former is typically shorter than 20% of the original length on average. For the decoder, we set it with three attention layers, and each layer has the same size as the attention layer in the encoder. The complexity of the decoder is about 1.2 GFLOPs. The prior inference model (DeiT-Samll) with the original image input achieves 4.6 GFLOPs. This indicates that the total overhead of the lightweight inference will not exceed 50% of the original inference.

## 5 EVALUATION

### 5.1 Experiment settings

We deploy an A-MOT server on an Ubuntu system with an Intel i7-10700 CPU and a Nvidia Geforce RTX 3080 GPU. We use the ImageNet-1K dataset (1000 classes, each with 50 images for the validation, and around 1200 images for the training) for inference. We apply the Large-ViT-based encoder in MAE [12] for the image recovery and the DeiT model (a variant of ViT) [34] to infer the images. The confidence threshold to select the images for the second round of offloading in the image selection part is set at 0.8. With various total traffic budget conditions, we compare our system to the following baselines in accuracy, which also don't require a lot of computing on the device side:

- **Non reconstruction (Non-R).** A natural baseline is to directly infer the transmitted data without any reconstruction operations.
- **Server-driven transmission (SDT).** The DDS [5] is configured to transmit a low-quality image to the server and retransmit high-quality content of the chosen areas again for video analysis. We apply this method to transmit low-quality images first and then offload high-quality content of the target area in the image. We choose the best performance among various combinations of $r_1$ and $r_2$ as the final result.

- **Image Super resolution (ISR).** Another possible solution is to transmit a low-quality image to the server, and the server reconstructs a high-resolution (HR) image with a trained model, which is also called the super-resolution (SR) method. The HR image can be further inferred. We also follow this principle to generate HR images with the commonly used SR model proposed in [39].

Besides, we also compare A-MOT with some computation-required solutions in both accuracy and communication overhead:

- **Partition-oriented inference (PO)**. The inference model contains multiple attention layers, and each layer has the same structure. We let the device complete the first attention layer and transmit the intermediate results to the server in a split form like in[22].
- **Data filtering on the device (DF)**. The image can contain lots of redundant content, we let the device run a MobileNet-SSD [13] based model locally, to select the crucial area in the image first. Only the selected region is transmitted to the server, like in [6].
- **Model-based compression (MBC)**. After splitting the inference model between the edge device and the server, it also has an encoder-decoder scheme to compress the intermediate results to reduce communication [25].

## 5.2 Results

*5.2.1 With Computation-free Baselines.* Fig. 11 shows the overall results when compared with these computation-free baselines. The number on the x-axis represents the total traffic budget. Suppose 100% means to transmit all the raw images to the server, we compare the performances of solutions under various given traffic budgets (different percentages). Our solution improves the accuracy by even more than 10% at a 40% transmission budget compared with the Non-R solution. The gap is gradually reduced when the total transmission ratio gets larger, which is reasonable since all solutions should have similar performance when transmitting the total images. It is worth noting that in A-MOT, when the total budget is small (such as 40%), we allocate 10% of the raw data volume as the budget for $r_2$ in the second round of offloading, and it becomes 20% for the other larger budget settings. This heuristic can bring the best performance in our experiments, and we will discuss more details in the latter subsection. The SDT does better than the Non-R solution and still lags behind A-MOT since the MAE model in A-MOT can help to learn genuine visual semantics. The gap is also more noticeable when the total transmission budget is low. ISR achieves only a slight advantage over Non-R. Its one-time transmission strategy is far from content-aware offloading.

*5.2.2 With computation-required Baselines.* Consider transmitting the raw images with a communication cost of one, we normalize all communication costs to the raw data. Fig. 12 shows the results when compared with the baseline that contains computation on the device side. For computation, PO demands about 400 MFLOPs, DF demands about 1200 MFLOPs, and MBC demands about 500 MFLOPs, while our method (A-MOT) almost demands no computation on the device. For communication, PO has the highest data transmission requirements, and its accuracy is also the highest. A-MOT has a small data transmit requirement and the accuracy gaps between PO and DF are slight. MBC has a small communication overhead. When the device is weak, A-MOT offers significant advantages in terms of power savings while guaranteeing accuracy.

## 5.3 The Influence of Various Strategies in A-MOT

*5.3.1 About Image Selection and SLO-adaptive Modules.* There are some key designs in A-MOT, including the image selection module and the SLO-adaptive module. We chose some different settings in A-MOT to evaluate the effort of the above designs.

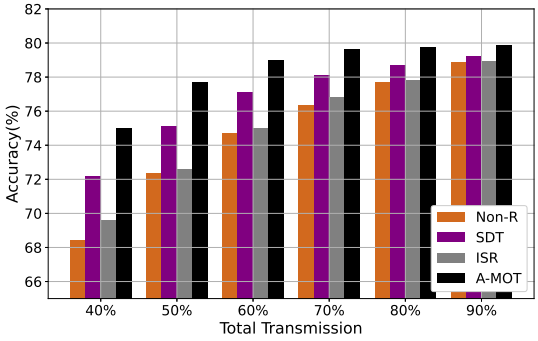



Fig. 11. The overall results with computation-free baselines.

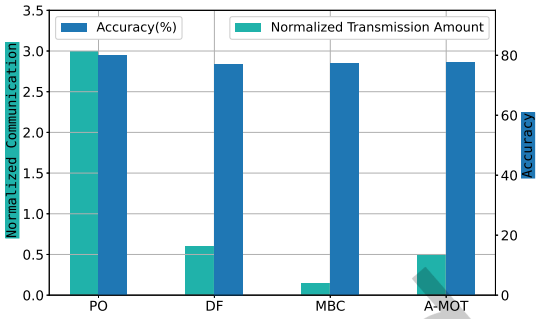



Fig. 12. The results with computation-contained baselines.

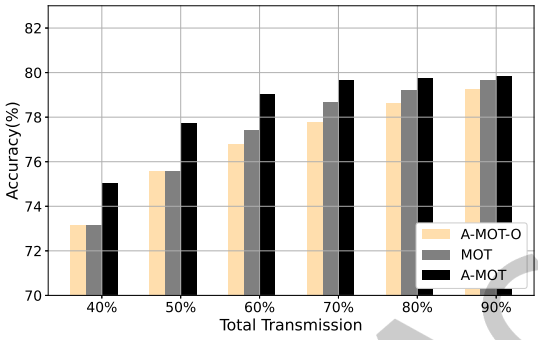



Fig. 13. The Influence of Various Strategies in A-MOT.

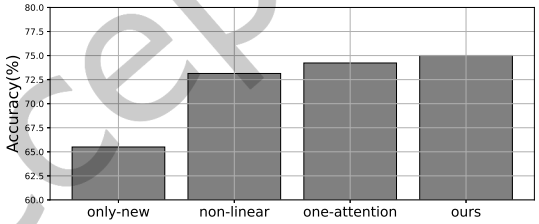



Fig. 14. Accuracy of different settings in the lightweight inference modules.

**A-MOT-O**. We propose a two-round offloading scheme in A-MOT. We also compare the accuracy when there is only a one-time transmission process, called A-MOT-O. The images also need to be reconstructed and inferred, and the result will be directly transmitted to the device as the final outcome.

**MOT**. We propose the SLO-adaptive module to set different $r_2$ for different images. We also compare our solution to the same $r_2$ setting, which is our work in the previous conference version (MOT) [24].

The results are shown in Fig. 13. The OFTT solution and the MOT solution have similar accuracy when the total transmitted data is small, and the benefit of the feedback strategy in the MOT solution will be obvious when the data amount gets larger since it achieves content-aware transmission. Basically, the key designs can further help the mask-reconstruct scheme in A-MOT to improve the accuracy by about 2% when the traffic budget is limited.

*5.3.2 About Lightweight Inference Modules.* We also evaluate the lightweight inference module by comparing it with various alternative settings, such as only inferring newly offloaded data (only-new), combining the result of the encoder and embeddings from the first inference without linear layers (non-linear), and employing the decoder with only one attention layer (one-attention). Fig. 14 shows the accuracy results of these settings when the total traffic budget is 40%. We can see that using only new data can result in a significant decrease in accuracy.

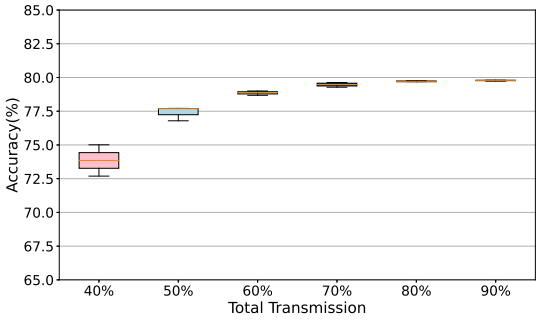



Fig. 15. The accuracy and various $r_1$.

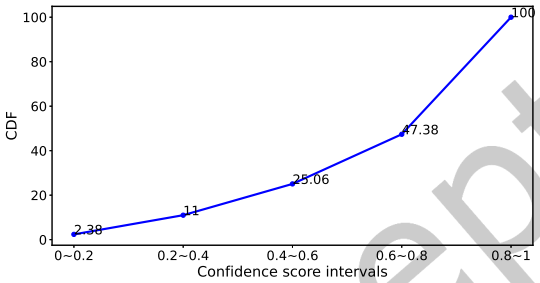



Fig. 16. Percentage of samples that can be correctly classified in each interval.

The non-linear method lags behind our solution as well. Usually, the accuracy is proportional to the number of layers in the decoder model; thus, the one-attention setting is insufficient to achieve the desired accuracy.

## 5.4 The Choice of $r_1$

With a constant transmission budget, there are multiple $r_1$, $r_2$ combinations that can be chosen in A-MOT. Different $r_1$ can lead to various inference accuracy. Fig. 15 shows the distribution of accuracy among various $r_1$ with different transmission budgets. We can see that the accuracy has a more concentrated distribution when the total budget is large. In A-MOT, allocating 10% or 20% of the original data volume as $r_2$ for the second transfer can achieve the best accuracy. This small amount of $r_2$ also means a more lightweight inference for the second round of offloaded data. As we mentioned before, even considering the extra attention-based decoder, the total overhead of the lightweight inference will not exceed 50% of the original inference.

## 5.5 The Choice of Threshold

In A-MOT, we set a threshold to decide if the images need a second round of offloading according to their confidence score. The threshold setting can help reduce transmission since we can obtain the output of the high-confidence image directly. We show the distribution of the raw image samples in the ImageNet data set among various confidence scores in Fig. 16. With a set threshold of 0.8 in A-MOT, the upper bound percentage of only once-time transmission samples can be about 53%. This means that many images only require once offload and once inference, allowing the limited budget to be focused on improving the identification accuracy of low-confidence images. The specific percentage of these one-time samples is related to the setting of the first-time

transmission ratio $r_1$ since $r_1$ can decide the distribution of the confidence scores. The percentage is increased as $r_1$ becomes larger basically. In A-MOT, we have a larger $r_1$ when there is more traffic budget; thus, more samples will be inferred only once, and the resource will focus on the low-confidence image.

## 6 RELATED WORK

### 6.1 Offloading

For edge devices, their data can be trained with federated learning [50, 52], and the data inference with offloading to the server also exhibits an increasing pattern. Li et al. [22] proposed to adaptive split DNN and model right-sizing with an on-demand inference framework to tackle the network latency. Shi et al. [32] further consider the privacy issue during the partition process. The energy consumption situations and task deadline constraints are also added to the partition problem [2, 42]. Jeong [16] et al. considered the incremental offloading problem and proposed to partition a DNN into various subtasks. The prediction of energy consumption for each DNN's layer is proposed in [19], and then a partition scheme is adopted. These partition-based methods have an irreparable limitation when applied to real computation-constrained edge devices. The drawback is that the intermediate data sizes of the first several layers are still large [44], even larger than the raw data [33]. The device does not have enough capacity or energy budget to compute the whole layers before the small output point. There are also some works considered from data compression [25, 26, 44] with extra-trained encoder and decoder, and the encoder is conducted by the edge device.

In our scheme, the decoder and encoder are all moved into the server for time-saving. [10] and [9] consider accelerating the inference with the cooperation of multiple edge devices, which is a different scene from ours.

There are some works that focus on the task of video analysis in an offloading scheme. zhang et al. [48] proposed only uploading the frame/camera that has the best view to capture the scene. [23] adjusts the encoding quality on each frame to reduce the transmission latency based on the Regions of Interest (RoIs) detected in the last offloaded frame. The DDS [5] is present to transmit a low-quality image to the server in video analysis. The server conducts an inference model for tasks like object detection or semantic segmentation and chooses the uncertain area. The device transmits the high-quality content of the chosen area again to achieve high inference accuracy. This video-focused work is inefficient to apply to image recognition since the target and input are different. Similar input data redundant reduction works have also been proposed [15, 18]. However, they all demand computing power from the device.

### 6.2 Image Sparsity and Completion

In contrast to language that contains high semantics and information density, an image usually contains heavy spatial redundancy [12]. There are some works trying to reduce the redundancy in the inference of images to accelerate the process and improve accuracy. Different from the common operation in ViT that divides an image into $14 \times 14$ tokens, the DVT [40] claims that some images are suitable for $4 \times 4$ tokens, thus reducing computation overhead. It processes the image by sequentially activating a cascade of Transformers with increasing numbers of tokens. DynamicVit uses a prediction module to prune the unimportant tokens in the training to accelerate the process. Similar image sparsity ideas have also been proposed in [28, 37, 43]. These works are orthogonal to ours since they reduce the redundancy during the conducting process rather than at the beginning.

There are some works related to image completion or image inpainting that target filling reasonable content with photorealistic appearance into missing regions [3, 49]. These works have different targets for our reconstruction process; they may reconstruct totally unrelated content when we have a large mask ratio.

## 7 CONCLUSION

In this work, we propose a two-round offloading scheme to save bandwidth resources for weak edge devices. The MAE model is utilized to recover the sampled images transmitted from the edge. All images transmit a small volume of data at the first round; the confidence and attention results from the inference will determine if the second offloading is needed and what content should be offloaded. The SLO-adaptive module is also made to look into how the confidence score and the amount of data sent are related to determine the transmission volume for each image independently. Finally, the lightweight inference module is proposed to save inference time and improve accuracy. We compare our system with different baselines to verify its effectiveness.

## ACKNOWLEDGMENTS

This research is supported by Japan Society for the Promotion of Science (JSPS) KAKENHI No. 21H03424, and Japan Science and Technology Agency (JST) PRESTO No. 23828673.



## REFERENCES

[1] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. 2022. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794* (2022).

[2] Xing Chen, Jianshan Zhang, Bing Lin, Zheyi Chen, Katinka Wolter, and Geyong Min. 2021. Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments. *IEEE Transactions on Parallel and Distributed Systems* 33, 3 (2021), 683–697.

[3] Qiaole Dong, Chenjie Cao, and Yanwei Fu. 2022. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11358–11368.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[5] Kuntai Du, Ahsan Pervaiz, Xin Yuan, Aakanksha Chowdhery, Qizheng Zhang, Henry Hoffmann, and Junchen Jiang. 2020. Server-driven video streaming for deep learning inference. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 557–570.

[6] Kuntai Du, Qizheng Zhang, Anton Arapin, Haodong Wang, Zhengxu Xia, and Junchen Jiang. 2022. AccMPEG: Optimizing Video Encoding for Video Analytics. *arXiv preprint arXiv:2204.12534* (2022).

[7] Boqian Fu, Fahao Chen, Peng Li, and Deze Zeng. 2022. TCB: Accelerating Transformer Inference Services with Request Concatenation. In *Proceedings of the 51st International Conference on Parallel Processing*. 1–11.

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[9] Ramyad Hadidi, Bahar Asgari, Jiashen Cao, Younmin Bae, Da Eun Shim, Hyojong Kim, Sung-Kyu Lim, Michael S Ryoo, and Hyesoon Kim. 2020. LCP: A low-communication parallelization method for fast neural network inference in image recognition. *arXiv preprint arXiv:2003.06464* (2020).

[10] Ramyad Hadidi, Jiashen Cao, Matthew Woodward, Michael S Ryoo, and Hyesoon Kim. 2018. Real-time image recognition using collaborative iot devices. In *Proceedings of the 1st on Reproducible Quality-Efficient Systems Tournament on Co-designing Pareto-efficient Deep Learning*. 1.

[11] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. 2022. Vision GNN: An Image is Worth Graph of Nodes. *arXiv preprint arXiv:2206.00272* (2022).

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16000–16009.

[13] 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).

[14] Ling Hu and Qiang Ni. 2017. IoT-driven automated object detection algorithm for urban surveillance systems in smart cities. *IEEE Internet of Things Journal* 5, 2 (2017), 747–754.

[15] Kai Huang and Wei Gao. 2022. Real-time neural network inference on extremely weak devices: agile offloading with explainable AI. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 200–213.

[16] 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. In *Proceedings of the ACM Symposium on Cloud Computing*. 401–411.

[17] Joo Seong Jeong, Jingyu Lee, Donghyun Kim, Changmin Jeon, Changjin Jeong, Youngki Lee, and Byung-Gon Chun. 2022. Band: coordinated multi-DNN inference on heterogeneous mobile processors. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. 235–247.
[18] Shanyang Jiang and Lan Zhang. 2022. Quality-aided Annotation Service Selection in MLaaS Market. In *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–11.
[19] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News* 45, 1 (2017), 615–629.
[20] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.
[21] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu, and Bharat Bhargava. 2013. A survey of computation offloading for mobile systems. *Mobile networks and Applications* 18, 1 (2013), 129–140.
[22] En Li, Liekang Zeng, Zhi Zhou, and Xu Chen. 2019. Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications* 19, 1 (2019), 447–457.
[23] Luyang Liu, Hongyu Li, and Marco Gruteser. 2019. Edge assisted real-time object detection for mobile augmented reality. In *The 25th annual international conference on mobile computing and networking*. 1–16.
[24] Tao Liu, Peng Li, Yu Gu, and Peng Liu. 2023. Efficient Transformer Inference for Extremely Weak Edge Devices using Masked Autoencoders. In *2023 IEEE International Conference on Communications (ICC)*.
[25] Yoshitomo Matsubara, Davide Callegaro, Sameer Singh, Marco Levorato, and Francesco Restuccia. 2022. BottleFit: Learning Compressed Representations in Deep Neural Networks for Effective and Efficient Split Computing. *arXiv preprint arXiv:2201.02693* (2022).
[26] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. 2018. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4394–4402.
[27] Minghui Min, Liang Xiao, Ye Chen, Peng Cheng, Di Wu, and Weihua Zhuang. 2019. Learning-based computation offloading for IoT devices with energy harvesting. *IEEE Transactions on Vehicular Technology* 68, 2 (2019), 1930–1941.
[28] Mahyar Najibi, Bharat Singh, and Larry S Davis. 2019. Autofocus: Efficient multi-scale inference. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9745–9755.
[29] Jiaming Qiu, Ruiqi Wang, Ayan Chakrabarti, Roch Guérin, and Chenyang Lu. 2022. Adaptive edge offloading for image classification under rate limit. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 11 (2022), 3886–3897.
[30] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
[31] Farah Ait Salaht, Frédéric Desprez, and Adrien Lebre. 2020. An overview of service placement problem in fog and edge computing. *ACM Computing Surveys (CSUR)* 53, 3 (2020), 1–35.
[32] Chengshuai Shi, Lixing Chen, Cong Shen, Linqi Song, and Jie Xu. 2019. Privacy-aware edge computing based on adaptive DNN partitioning. In *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.
[33] Wenqi Shi, Yunzhong Hou, Sheng Zhou, Zhisheng Niu, Yang Zhang, and Lu Geng. 2019. Improving device-edge cooperative inference of deep learning via 2-step pruning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 1–6.
[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, Vol. 139. 10347–10357.
[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[36] VMW Research Group. 2022. The GFLOPS of the various machines. https://web.eece.maine.edu/~vweaver/group/green_machines.html.
[37] Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. 2021. Exploring sparsity in image super-resolution for efficient inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4917–4926.
[38] Robert J Wang, Xiang Li, and Charles X Ling. 2018. Pelee: A real-time object detection system on mobile devices. *Advances in neural information processing systems* 31 (2018).
[39] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1905–1914.
[40] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. 2021. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems* 34 (2021), 11960–11973.
[41] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*. PMLR, 23965–23998.
[42] 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.

[43] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. 2020. Resolution adaptive networks for efficient inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2369–2378.

[44] Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Shengzhong Liu, Huajie Shao, and Tarek Abdelzaher. 2020. Deep compressive offloading: Speeding up neural network inference by trading edge computation for network latency. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 476–488.

[45] Juheon Yi and Youngki Lee. 2020. Heimdall: mobile GPU coordination platform for augmented reality applications. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.

[46] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917* (2022).

[47] Yufeng Zhan, Song Guo, Peng Li, and Jiang Zhang. 2020. A deep reinforcement learning based offloading game in edge computing. *IEEE Trans. Comput.* 69, 6 (2020), 883–893.

[48] Tan Zhang, Aakanksha Chowdhery, Paramvir Bahl, Kyle Jamieson, and Suman Banerjee. 2015. The design and implementation of a wireless video surveillance system. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 426–438.

[49] Chuanxia Zheng, Tat-Jen Cham, Jianfei Cai, and Dinh Phung. 2022. Bridging Global Context Interactions for High-Fidelity Image Completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11512–11522.

[50] Xiaokang Zhou, Wei Liang, I Kevin, Kai Wang, Zheng Yan, Laurence T Yang, Wei Wei, Jianhua Ma, and Qun Jin. 2023. Decentralized P2P Federated Learning for Privacy-Preserving and Resilient Mobile Robotic Systems. *IEEE Wireless Communications* 30, 2 (2023), 82–89.

[51] Xiaokang Zhou, Wei Liang, I Kevin, Kai Wang, and Laurence T Yang. 2020. Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations. *IEEE Transactions on Computational Social Systems* 8, 1 (2020), 171–178.

[52] Xiaokang Zhou, Xiaozhou Ye, I Kevin, Kai Wang, Wei Liang, Nirmal Kumar C Nair, Shohei Shimizu, Zheng Yan, and Qun Jin. 2023. Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications. *IEEE Transactions on Computational Social Systems* (2023).