Multi-View Edge Inference in Low-Altitude Airspace: Joint UAV Deployment and Model Architecture Optimization

Yao Tang, Guangxu Zhu, Chao Shen, Tse-Tin Chan, and Tat-Ming Lok

***Abstract*—The rapid emergence of the low-altitude economy (LAE) is enabling efficient and flexible applications via collaborative unmanned aerial vehicles (UAVs) operating in low-altitude airspace. In multi-view edge inference, UAVs capture data from diverse viewpoints and transmit features to an edge server, which fuses multi-view features for accurate recognition. However, the practical implementation is challenged by UAV-specific constraints and model architecture, notably UAV deployment and the view-pooling layer. In this paper, we investigate the joint design of UAV deployment and model architecture for multi-view edge inference in a target recognition task. To accommodate the UAV's limited onboard resources, we adopt split inference. We first derive a closed-form accuracy bound that incorporates both the classification margin and the quantization error. We establish a closed-form relation that explicitly links UAV deployment and view-pooling layer placement to the classification margin. Then, we characterize the quantization error introduced by feature quantization during wireless transmission. Our analysis shows that jointly optimizing these factors yields more discriminative features and a more effective model architecture, thereby improving inference accuracy. Moreover, these design choices collectively shape inference latency, revealing a tradeoff between accuracy and latency. We formulate a joint UAV deployment and model architecture optimization problem that minimizes inference latency subject to an accuracy constraint. To solve the resulting mixed-integer non-convex problem, we develop an alternating optimization approach that decomposes it into tractable subproblems. The proposed split, view-pooling, quantization, and position (SVQP) scheme iteratively optimizes the corresponding decision variables. Simulation results demonstrate that the proposed SVQP scheme achieves the best tradeoff between inference accuracy and latency compared with baselines.**

***Index Terms*—Low-altitude economy, edge inference, multi-view 3D shape recognition, UAV deployment design.**

I. Introduction

The low-altitude economy (LAE) is an emerging ecosystem characterized by unmanned aerial vehicles (UAVs), ground stations, and other stakeholders conducting economic activities within open low-altitude airspace [1]–[3]. This paradigm significantly enhances the speed, efficiency, and flexibility of operations in industries. Its realization is enabled by 6G networks, which combine distributed sensing with artificial intelligence (AI), providing an intelligent platform for diverse inference tasks [4]–[7]. The integration of UAVs further enhances this platform through multi-view sensing, making it especially effective for applications that require comprehensive situational awareness, such as target recognition, autonomous control, and autonomous driving. In a typical multi-view inference system, each UAV employs a lightweight sub-model to extract features from its observations and wirelessly transmits them to a server. The server then aggregates these features through an sub-model, enabling efficient and accurate inference by pooling information from multiple UAV viewpoints.

Despite these benefits, the effective implementation of multi-view edge inference in low-altitude airspace faces significant challenges arising from the UAV deployment and limited onboard resources. *First, UAV deployment affects inference accuracy.* Owing to practical constraints in positioning, the data (e.g., images) collected by UAVs are strongly dependent on their viewpoints, which are dictated by their positions. Certain viewpoints capture more representative features of the target and thus improve inference accuracy, whereas others contribute less effectively. *Second, UAV deployment affects inference latency.* The transmission rate of a UAV depends on the quality of the wireless channel between the UAV and the server. Larger elevation angles favor line-of-sight (LOS) links, resulting in higher transmission rate [8]. However, prioritizing communication efficiency can constrain the UAVs to suboptimal sensing viewpoints, thereby degrading inference accuracy. Consequently, UAV deployment should balance a fundamental tradeoff between inference accuracy and inference latency. *Third, limited onboard resources on UAVs constrain inference efficiency.* The UAVs operate under strict power and hardware constraints that limit both intermediate feature transmission and onboard computation, ultimately reducing inference efficiency. Split inference for multi-view models is an effective approach to alleviate communication and computation burdens. However, the model architecture, specifically the choice of split layer and view-pooling layer, influences the inference accuracy. Therefore, achieving efficient and accurate multi-view edge inference requires joint optimization of UAV deployment and the model architecture.

Prior works [9]–[14] have primarily focused on mitigating communication and computation bottlenecks arising from high-dimensional feature transmission and the time-consuming loading of models. For instance, efficient feature

Y. Tang is with the AI for Science and Engineering Center, Shenzhen Loop Area Institution, Shenzhen 518000, China (e-mail: yaotang@slai.edu.cn).

G. Zhu and C. Shen are with Shenzhen Research Institute of Big Data, Shenzhen, China, and the Chinese University of Hong Kong, Shenzhen, Shenzhen, China (e-mail: {gxzhu, chaoshen}@sribd.cn).

Tse-Tin Chan is with the Department of Mathematics and Information Technology, The Education University of Hong Kong, Hong Kong (e-mail: tsetinchan@eduhk.hk).

T.-M. Lok is with the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong (e-mail: tmlok@ie.cuhk.edu.hk).

extraction [9]–[11] ensures that only task-relevant information is transmitted, reducing payload size and computational overhead. Sensor scheduling strategies [12] further decrease bandwidth usage by minimizing the number of concurrently active transmitters. Over-the-air computation (AirComp) [13], [14] enables air aggregation, improving the efficiency of transmission. Moreover, inference accuracy characterization has been studied in task-oriented edge AI systems [15]–[20]. These studies aim to establish analytical relationships between inference performance and key system factors, such as sensing quality, communication distortion, computation capability, and feature compression. However, these studies rarely account for UAV mobility when UAVs are used as sensing assets. The joint impact of UAV deployment and model architecture on inference performance remains unexplored. Therefore, we aim to *optimize UAV deployment and model architecture to enhance performance in multi-view edge inference systems*.

A. Contributions

In this paper, we propose a novel target recognition approach for a multi-view edge inference system comprising multiple UAVs and an edge server. Each UAV acts as a distributed sensor, capturing the same target from different viewpoints. We employ a multi-view convolutional neural network (MVCNN) within a split inference framework. Each UAV runs a lightweight sub-model to extract features from its observations, applies quantization to compress the features, and transmits them to the server. The server hosts an additional sub-model with a view-pooling layer that aggregates the received features to perform the final inference. In this framework, inference accuracy is determined by the sensing data and the model architecture, which depends on the UAV deployment, split layer, and view-pooling layer. We analytically establish a closed-form relationship between these factors and the classification margin, a central factor influencing inference accuracy. In addition, we evaluate how feature quantization in transmission deteriorates inference accuracy. These design parameters also affect inference latency, revealing the tradeoff between accuracy and latency. We minimize latency under an accuracy constraint by jointly optimizing UAV deployment, split layer, view-pooling layer, and quantization level.

We summarize the key results and contributions as follows:

- **Explicit characterization of inference accuracy:** We analyze inference accuracy via the classification margin and quantization error. First, we derive a closed-form relation that explicitly connects UAV deployment and view-pooling layer placement to the classification margin. Second, we characterize the quantization error introduced by feature quantization during wireless transmission. Our analysis reveals how these factors jointly influence inference accuracy, and shows that optimizing them yields more discriminative features and a more effective model architecture, thereby improving inference accuracy.
- **Inference accuracy-latency tradeoff:** Based on the above theoretical results of inference accuracy, we formulate a joint UAV deployment and model architecture optimization problem that minimizes inference latency subject to an accuracy constraint. To address this mixed-integer non-convex problem, we adopt an alternating optimization approach and decompose it into three subproblems that optimize: (i) the split layer and view-pooling layer placement, (ii) the quantization level, and (iii) the UAV positions. The proposed scheme can efficiently compute high-quality suboptimal solutions.
- **Performance evaluation:** We validate our approach on selected real-world datasets with extensive simulations for a multi-view target recognition task. The results show that our split, view-pooling, quantization, and position optimization (SVQP) scheme achieves a superior tradeoff between inference accuracy and latency compared to other baseline schemes.

The rest of this paper is organized as follows. Section II reviews the related work. Section III introduces the system model. Section IV presents the theoretical analysis of the inference accuracy. Section V establishes the problem formulation and optimization. Section VI discusses the simulation results, and Section VII concludes this paper.

II. Related Works

Studies on multi-view edge inference commonly adopt the MVCNN, which fuses features from distributed sensors into a global feature map for server side inference [12], [21], [22]. Most existing work targets the communication and computation bottlenecks that arise during inference. Typical mitigation strategies include efficient feature extraction [9]–[11], sensor scheduling [12], and over-the-air computation (AirComp) [13], [14], [17]. *Feature extraction:* In [9], the feature extraction and encoding process is optimized to ensure that only inference-relevant information is transmitted. The work in [10] introduced a significance-aware approximation across multiple nodes, reducing the design space by accounting for the nonuniform contributions of different views to the inference. Similarly, [11] proposed significance-aware feature selection to improve the efficiency of multi-view object detection. *Sensor scheduling:* In [12], a bandwidth-efficient framework is presented that learns optimal group construction and communication timing. *AirComp:* In [13], importance-aware beamforming is enabled by considering the average discrimination gain per feature dimension. The work in [14] employed AirComp for feature aggregation and adopted batch processing to reduce memory access during the inference.

Inference accuracy characterization has recently been studied in task-oriented edge AI systems. In [15], the authors characterized the inference performance of multi-device edge AI by adopting discriminant gain and derived its closed-form relationship with sensing, communication, computation, and quantized feature transmission. The work in [16] investigated energy-efficient edge inference with adjustable split inference, model pruning, and feature quantization, and derived an explicit approximation of inference accuracy. In [17], the authors proposed over-the-air multi-view pooling and applied classification-margin theory to relate aggregation errors to classification accuracy. In [18], they analyzed the view-and-channel aggregation gain in integrated sensing and edge AI by

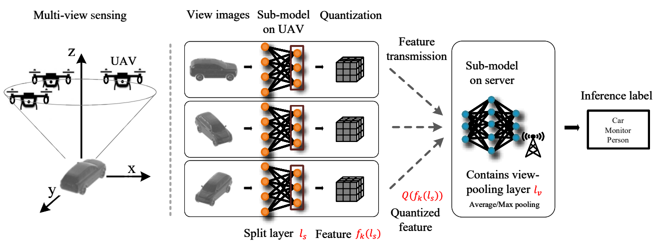

Fig. 1. Multi-view edge inference system in low-altitude airspace.

characterizing end-to-end inference uncertainty and discriminant gain. Moreover, split-inference studies such as [19], [20] investigated feature transmission and feature compression to improve the latency-accuracy tradeoff.

However, these works do not explicitly characterize the inference accuracy under the impact of UAV positions. This limitation can substantially impact inference accuracy, as UAV deployment directly shapes the sensing data. Moreover, the joint impact of UAV deployment and model architecture on inference performance remains unexplored. Therefore, we aim to address the open problem of optimizing UAV deployment and model architecture to enhance performance in multi-view edge inference systems.

III. System Model

As shown in Fig. 1, we consider a multi-view edge inference system operating in low-altitude airspace with K UAVs and an edge server. Let the UAV set be $\mathcal{K} = \{1, 2, \cdots, K\}$. The UAVs collaborate with the edge server to execute a joint inference task. Specifically, each UAV acquires data of a common target from its own viewpoint and extracts local features using a lightweight sub-model. Then, the edge server aggregates the received features via its sub-model to perform inference by fusing information across UAV perspectives. This architecture supports accurate and efficient inference for applications, such as multi-view object detection in security networks and vehicle re-identification in urban surveillance.

A. *Multi-view & Split Inference Model*

We consider the multi-view convolutional neural network (MVCNN) architecture for the edge inference [21]. We employ a pre-trained MVCNN model with a view-pooling layer at $l_v \in \mathcal{L}$, where $\mathcal{L} = \{1, \dots, L\}$ indexes the model layers. The view-pooling layer aggregates features across views using average, max, or other pooling strategies. Moreover, limited onboard computation on UAVs and wireless resource constraints for transmitting high-dimensional data to the server make low-latency inference challenging. To mitigate these issues, we split the MVCNN at a split layer $l_s \in \mathcal{L}$. The sub-model comprising layers $\mathcal{L}^{UAV} = \{1, \cdots, l_s\}$ is deployed on the UAV, while the remaining layers $\mathcal{L}^{server} = \{l_s + 1, \cdots, L\}$ is executed on the edge server. The split is chosen to occur before the view-pooling layer, i.e., $l_s \leq l_v, l_s, l_v \in \mathcal{L}$.

B. *Sensing Model*

The UAVs are uniformly deployed around the target and simultaneously sense data from distinct viewpoints. We consider that the UAVs remain stationary during the sensing process. This assumption is reasonable for the considered multi-view target recognition task, since the UAVs only need to capture images of the target from different viewpoints within a short sensing duration. Therefore, even when the target is moving, e.g., a vehicle or pedestrian, the proposed method can recognize the target based on its instantaneous state at the sensing time. This model can also be extended to more dynamic scenarios, such as target tracking and continuous recognition, by incorporating UAV trajectory control [23].

The sub-model deployed on UAV k takes the captured image as input and produces a feature vector $\boldsymbol{f}_k(l_s) = [f_{k,1}, \cdots, f_{k,N(l_s)}] \in \mathbb{R}^{N(l_s)}$, where $N(l_s)$ denotes the dimensionality of the feature at the split layer l_s. Let $\mathcal{N}(l_s) = \{1, \cdots, N(l_s)\}$ denote the index set of feature dimensions at l_s. Moreover, each element of the output feature is bounded as $|f_{k,n}| \in [\underline{f}_{k,n}, \overline{f}_{k,n}], \forall n \in \mathcal{N}(l_s)$. With q quantization bits, we define the set of quantization levels $\{\delta_0, \cdots, \delta_{2^q-1}\}$, which are uniformly spaced over the interval $[\underline{f}_{k,n}, \overline{f}_{k,n}]$. Specifically, each level is given by $\delta_i = \underline{f}_{k,n} + i \times \frac{\underline{f}_{k,n} - \overline{f}_{k,n}}{2^q - 1}$, $i = 0, \cdots, 2^q - 1$. Given an arbitrary feature $f_{k,n} \in [\delta_j, \delta_{j+1})$, its quantized representation is assigned as [24], [25]

$$Q(f_{k,n}) = \begin{cases} \text{sign}(f_{k,n}) \times \delta_i, & \text{w.p.} \frac{\delta_{i+1} - |f_{k,n}|}{\delta_{i+1} - \delta_i}, \\ \text{sign}(f_{k,n}) \times \delta_{i+1}, & \text{w.p.} \frac{|f_{k,n}| - \delta_i}{\delta_{i+1} - \delta_i}, \end{cases} \tag{1}$$

where sign$(\cdot)$ denotes the signum function. Based on (1), the quantized feature of UAV k is expressed as $Q(\boldsymbol{f}_k(l_s)) \triangleq [Q(f_{k,1}), \cdots, Q(f_{k,N(l_s)})]$. The server then receives the set of quantized feature $\{Q(\boldsymbol{f}_k(l_s)), \forall k \in \mathcal{K}\}$ and processes them through the subsequent layers of the model.

C. *Communication Model*

We denote the three-dimensional positions of UAV k and the server as $\boldsymbol{u}_k = (x_k, y_k, z_k)$ and $\boldsymbol{s} = (x_s, y_s, z_s)$, respectively, where z_k and z_s denote their respective altitudes. The channel between UAV k and the server can be either line-of-sight (LOS) or non-line-of-sight (NLOS). The LOS probability between UAV k and the server is given by [8]

$$q_k(\boldsymbol{u}_k) = \frac{1}{1 + \psi \exp(-\zeta[\theta_{c,k}(\boldsymbol{u}_k) - \psi])}, \tag{2}$$

where $\theta_{c,k}(\boldsymbol{u}_k) = \frac{180°}{\pi} \times \sin^{-1} \left| \frac{z_{u,k} - z_s}{d_{c,k}} \right|$ is the communication elevation angle, and $d_{c,k} = \|\boldsymbol{u}_k - \boldsymbol{s}\|$ is the Euclidean distance between UAV k and the server. Moreover, the NLOS probability is $1 - q_k(\boldsymbol{u}_k)$. The average channel gain between UAV k and the server is [26], [27]

$$h_k(\boldsymbol{u}_k) = \frac{(K_0 d_{c,k})^{-\alpha}}{\eta_1 q_k(\boldsymbol{u}_k) + \eta_2(1 - q_k(\boldsymbol{u}_k))}, \tag{3}$$

where $K_0 = \frac{4\pi f_c}{c}$, f_c is the carrier frequency, c is the speed of light, and α is the path loss exponent of the link between the UAV and the server. Here, η_1 and η_2 ($\eta_2 > \eta_1 > 1$)

represent the excessive path loss coefficients for LOS and NLOS cases[1], respectively, which account for shadowing and reflection effects [8], [28]. Moreover, we assume that each UAV is allocated a dedicated sub-channel with bandwidth B_k for feature transmission. Suppose that UAV k uploads the feature to the server with power $p_k > 0$. The received rate at the server from UAV k is[2]

$$r_k(\boldsymbol{u}_k, p_k) = B_k \log\left(1 + \frac{p_k h_k(\boldsymbol{u}_k)}{B_k N_0}\right), \tag{4}$$

where N_0 is the noise power spectral density.

D. Inference Latency

In our system, the end-to-end inference latency comprises three components: 1) the delay incurred by model inference on the UAVs, 2) the delay caused by feature transmission between the UAVs and the server, and 3) the delay incurred by model inference on the server.

- ***UAV computation time:*** The computational overhead on each UAV is estimated based on the number of floating-point operations (FLOPs) required by its neural network layers. The computational workload of each layer $i \in \{1, \ldots, l_s\}$ denoted as λ_{ki}, includes operations from convolutional layer f_{conv}, pooling layer f_{pool}, and ReLU layer f_{relu}. We define v_k as the computation capacity of each UAV k. The inference latency on UAV k is [29]

$$T_k^{cp}(l_s, v_k) = \sum_{i \in \mathcal{L}^{UAV}} \frac{\lambda_{ki}}{v_k}, \tag{5}$$

where $\lambda_{ki} = w_{i1} f_{\text{conv}} + w_{i2} f_{\text{pool}} + w_{i3} f_{\text{relu}}$, and w_{i1}, w_{i2}, w_{i3} denote the number of convolutional layers, pooling layers, and ReLU layers, respectively. Here, $w_{i1} + w_{i2} + w_{i3} = l_s$.

- ***Network transmission time:*** Each UAV uploads its quantized feature $Q(\boldsymbol{f}_k(l_s))$ to the server. According to (1), let μ be the number of bits used to represent sign$(f_{k,n})$, $\underline{f}_{k,n}$, and $\overline{f}_{k,n}$. Since each vector contains $N(l_s)$ entries, it requires $N(l_s)q + \mu$ bits to encode these parts [24]. Therefore, the transmission time for UAV k's feature is

$$T_k^{cm}(l_s, q, \boldsymbol{u}_k, p_k) = \frac{N(l_s)q + \mu}{r_k(\boldsymbol{u}_k, p_k)}. \tag{6}$$

- ***Server computation time:*** For the server, the computational workload of each layer $i \in \{l_s, \ldots, L\}$ denoted as $\lambda_{s,i}$, includes operations from f_{conv}, f_{pool}, f_{relu}, fully-connected layer f_{conn}, and a softmax layer f_{softmax}. We define v_s as the computation capacity of the server. The inference latency on the server is [29], [30]

$$T_s^{cp}(l_s) = \sum_{i \in \mathcal{L}^{server}} \frac{\lambda_{si}}{v_s}, \tag{7}$$

where $\lambda_{si} = w_{i4} f_{\text{conv}} + w_{i5} f_{\text{pool}} + w_{i6} f_{\text{relu}} + w_{i7} f_{\text{conn}} + f_{\text{softmax}}$, and $w_{i4}, w_{i5}, w_{i6}, w_{i7}$ denote the number of convolutional layers, pooling layers, ReLU layers, fully-connected layers, and softmax layer, respectively. Here, $w_{i4} + w_{i5} + w_{i6} + w_{i7} = L - l_s$.

Above all, the inference latency is[3]

$$\begin{aligned} &T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\}) \\ &= \max_{k \in \mathcal{K}} \{T_k^{cp}(l_s, v_k) + T_k^{cm}(l_s, q, \boldsymbol{u}_k, p_k) + T_s^{cp}(l_s)\}. \end{aligned} \tag{8}$$

IV. Inference Accuracy Analysis

In this section, we analyze the inference accuracy and derive a closed-form lower bound. In Section IV-A, we introduce an inference accuracy metric that jointly captures the effects of the split layer, view-pooling layer, quantization, and UAV positions. In Sections IV-B and IV-C, we present lower bounds for the quantization error and the classification margin, respectively. In Section IV-D, we combine these results to provide a closed-form bound for inference accuracy.

A. Metric of Inference Accuracy

We analyze factors affecting inference accuracy by focusing on classification tasks as a representative case. Specifically, the UAVs quantize features before transmission to the server, introducing quantization errors. The quantized feature $\{Q(\boldsymbol{f}_k(l_s)), \forall k \in \mathcal{K}\}$ are first processed by the intermediate layers from l_s to l_v. The resulting features are then aggregated by the view-pooling operation to produce a single feature vector at layer l_v. Let $c_{par}(\cdot, l_v)$ denote the partial MVCNN from layer l_s up to l_v (exclusive). Feature pooling is implemented via average-pooling given by[4]

$$\hat{\boldsymbol{g}}(l_v) \triangleq \frac{1}{K} \sum_{k \in \mathcal{K}} c_{par}(Q(\boldsymbol{f}_k(l_s)), l_v), \tag{9}$$

where $Q(\boldsymbol{f}_k(l_s))$ serves as the input to the partial model $c_{par}(\cdot, l_v)$. The pooled feature vector is then processed by the remaining layers to produce the final inference result.

To assess the theoretical impact on inference accuracy, we introduce the classification margin, defined as the minimum distance from any feature vector to the decision boundary [17]. Accordingly, a sufficient condition for correct classification of $\hat{\boldsymbol{g}}(l_v)$ is that its error induced by quantization and intermediate processing remains strictly smaller than the margin, i.e.,

$$\|\hat{\boldsymbol{g}}(l_v) - \boldsymbol{g}(l_v)\|_2 \le \Delta(l_v, \{\boldsymbol{u}_k\}), \tag{10}$$

where $\boldsymbol{g}(l_v)$ denotes the unquantized feature, defined analogously to (9). The quantization error $\|\hat{\boldsymbol{g}}(l_v) - \boldsymbol{g}(l_v)\|_2$ depends on the split layer l_s, the view-pooling layer l_v, and the quantization level q. The classification margin $\Delta(l_v, \{\boldsymbol{u}_k\})$ depends on the model architecture determined by l_v, and on the sensed data determined by the UAV position $\{\boldsymbol{u}_k\}$. We present a lower bound for the inference accuracy as follows.

Lemma 1. *The lower bound of inference accuracy under multi-view edge inference system is*

$$R_{view}(l_s, l_v, q, \{\boldsymbol{u}_k\})$$

[1] The parameters η_1 and η_2 can be adjusted to simplify the model to the LoS case.

[2] Each UAV uploads its quantized feature within one uplink transmission phase using OFDM-based wireless transmission. In practice, the payload can be packetized and transmitted over multiple OFDM symbols or frames.

[3] For brevity, we write $\{\boldsymbol{u}_k\}$, $\{p_k\}$, and $\{v_k\}$ to denote $\{\boldsymbol{u}_k, \forall k \in \mathcal{K}\}$, $\{p_k, \forall k \in \mathcal{K}\}$, and $\{v_k, \forall k \in \mathcal{K}\}$, respectively.

[4] The analysis readily extends to max-pooling and other pooling strategies; average-pooling is adopted here for concreteness.

$$\geq R_0\left(l_v,\{\boldsymbol{u}_k\}\right)\Pr\left[\|\hat{\boldsymbol{g}}(l_v)-\boldsymbol{g}(l_v)\|_2\leq\Delta(l_v,\{\boldsymbol{u}_k\})\right]$$
$$\geq R_0\left(l_v,\{\boldsymbol{u}_k\}\right)\left(1-\frac{\mathbb{E}\left[\|\hat{\boldsymbol{g}}(l_v)-\boldsymbol{g}(l_v)\|_2^2\right]}{\Delta^2(l_v,\{\boldsymbol{u}_k\})}\right),\qquad(11)$$

where $R_0\left(l_v,\{\boldsymbol{u}_k\}\right)$ denotes the correct classification probability for a input sample sensed by UAVs at $\{\boldsymbol{u}_k\}$ using a MVCNN model with view-pooling at layer l_v. This corresponds to the ideal scenario in which no quantization error occurs during task inference. Moreover, the lower bound in (11) *is applicable to a broad class of multi-view target recognition models that map multiple rendered views into a feature representation followed by a classifier, including those using visual geometry group network (VGG), MobileNet, or vision transformer (ViT) as the backbone network.*

Proof. By applying the well-known Markov's inequality, the inequality can be derived. □

Next, we analyze the quantization error and the classification margin in the following.

B. Quantization Error

According to the stochastic quantization method [24], the expected squared norm of quantization error is bounded by

$$\mathbb{E}\left[\|\hat{\boldsymbol{g}}(l_v)-\boldsymbol{g}(l_v)\|_2^2\right]\leq\frac{1}{K}\sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{N}(l_s)}\frac{(\overline{f}_{k,n}-\underline{f}_{k,n})^2}{4(2^q-1)^2}\sigma,\quad(12)$$

where $\overline{f}_{k,n}$ and $\underline{f}_{k,n}$ denote the maximum and the minimum of feature element $f_{k,n}$ for $n\in\mathcal{N}(l_s)$, respectively. Here, $\sigma\triangleq\sup_{x\in\Omega}\|\mathbf{J}_{\rm par}(x,l_v)\|_2^2$, and $\mathbf{J}_{\rm par}(x,l_v)=\prod_{l=l_s+1}^{l_v}\frac{\partial\phi_l(\boldsymbol{w}_{l-1})}{\partial\boldsymbol{w}_{l-1}}$ denotes the Jacobian matrix (JM) of the partial model $c_{\rm par}(\cdot,l_v)$. The function $\phi_l(\cdot)$ represents the l-th layer mapping, and $\boldsymbol{w}_l$ is the output of the l-th layer. The dataset $\Omega=\operatorname{conv}(\mathcal{X}(\{\boldsymbol{u}_k\}))$ consists of samples sensed by UAVs positioned at $\{\boldsymbol{u}_k\}$.

Proof. According to (9), the expected squared norm of the quantization error can be written as [31]

$$\mathbb{E}\left[\|\hat{\boldsymbol{g}}(l_v)-\boldsymbol{g}(l_v)\|_2^2\right]$$
$$=\mathbb{E}\left[\left\|\frac{1}{K}\sum_{k\in\mathcal{K}}\left(c_{par}(Q(\boldsymbol{f}_k(l_s)),l_v)-c_{par}(\boldsymbol{f}_k(l_s),l_v)\right)\right\|_2^2\right]$$
$$\leq\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\|c_{par}(Q(\boldsymbol{f}_k(l_s)))-c_{par}(\boldsymbol{f}_k(l_s))\|_2^2\right]\qquad(13)$$

where the inequality follows from Jensen's inequality. Based on Theorem 3 in [31], we have

$$\mathbb{E}\left[\|c_{par}(Q(\boldsymbol{f}_k(l_s)))-c_{par}(\boldsymbol{f}_k(l_s))\|_2^2\right]$$
$$\leq\sigma\cdot\mathbb{E}\left[|c_{par}(Q(f_{k,n}),l_v)-c_{par}(f_{k,n},l_v)|^2\right],\qquad(14)$$

where $\sigma=\sup_{x\in\Omega}\|\mathbf{J}_{\rm par}(x,l_v)\|_2^2$, $\mathbf{J}_{\rm par}(x,l_v)=\prod_{l=l_s+1}^{l_v}\frac{\partial\phi_l(\boldsymbol{w}_{l-1})}{\partial\boldsymbol{w}_{l-1}}$ is the JM of the partial model $c_{\rm par}(\cdot,l_v)$, comprising layers from l_s up to (but excluding) l_v. Here, $\phi_l(\cdot)$ represents the function at the l-th layer, and $\boldsymbol{w}_l$ denotes the output of the l-th layer. The dataset $\Omega=\operatorname{conv}(\mathcal{X}(\{\boldsymbol{u}_k\}))$ consists of data acquired by UAVs positioned at $\{\boldsymbol{u}_k\}$. Moreover, with the stochastic quantization in (1), the quantization error for each element n is bounded by [24]

$$\mathbb{E}\left[|Q(f_{k,n})-f_{k,n}|^2\right]\leq(\frac{\delta_i-\delta_{i+1}}{2})^2,\qquad(15)$$

where $\delta_i-\delta_{i+1}=\frac{\overline{f}_{k,n}-\underline{f}_{k,n}}{2^q-1}$. Substituting (14) and (15) into (13), we have (12). Given the complexity of σ, we treat it as a constant, estimated empirically through simulation. □

C. Classification Margin

Let the set of images captured from different views of target j be $\{(\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj}),\forall k\in\mathcal{K}\}$, where $\boldsymbol{x}_{kj}(\boldsymbol{u}_k)$ is the image acquired by UAV k at position $\boldsymbol{u}_k$, and y_{kj} is its classification label. The intuition is to identify a single most representative data point $s_j=(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}),y_j)$ for the set $\{(\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj}),\forall k\in\mathcal{K}\}$, ensuring consistent inference labels across all views, i.e., $y_j=y_{kj},\forall k\in\mathcal{K}$. We apply the score concept from [31] to derive a lower bound on the classification margin. This facilitates analysis of how the choices of l_v and $\{\boldsymbol{u}_k\}$ affect the margin. Based on this framework, we present the following proposition.

Proposition 1. *In the multi-view edge inference system, the classification margin of the MVCNN model with view-pooling layer l_v on the entire dataset Ω is bounded by*

$$\Delta(l_v,\{\boldsymbol{u}_k\})>\frac{A-B\times\max\limits_{\boldsymbol{x}_{kj}(\boldsymbol{u}_k)\in\Omega}\|\boldsymbol{x}_{kj}(\boldsymbol{u}_k)-\boldsymbol{x}_j(\{\boldsymbol{u}_k\})\|_2}{\sup\limits_{\boldsymbol{x}\in\Omega}\|\mathbf{J}(\boldsymbol{x},l_v)\|_2},\ (16)$$

where $A=\min\limits_{\boldsymbol{x}_{kj}(\boldsymbol{u}_k)\in\Omega}o((\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj}),l')-2\|\boldsymbol{\xi}\|_2$ and $B=2\sup\limits_{\boldsymbol{x}\in\Omega}\|\mathbf{J}(\boldsymbol{x},l')\|_2$. Here, $o((\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj}),l')$ denotes the score of sample $(\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj})$ under the reference (no pooling) model, i.e., $l'=0$. The dataset's minimum score is lower bounded by a constant, and the vector $\boldsymbol{\xi}$ is also constant. The Jacobians $\mathbf{J}(\mathbf{x},l')$ and $\mathbf{J}(\mathbf{x},l_v)$ are those of the reference and MVCNN models, respectively.

Proof. To derive a closed-form expression for the classification margin, we adopt the score concept from [31]. Consider an MVCNN model $c(\boldsymbol{x},l_v)$ with a view-pooling layer at l_v, which classifies target j using the sample set $\{(\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj}),\forall k\in\mathcal{K}\}$ with the score

$$o(s_j,l_v,\boldsymbol{u}_k)=\min_{\tau\neq y_j}\sqrt{2}(\delta_{y_j}-\delta_\tau)^T c\left(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}),l_v\right),\quad(17)$$

where $\delta_\tau\in\mathbb{R}^C$ denotess the Kronecker delta vector with $(\delta_\tau)_\tau=1$. Since the features are pooled at layer l_v, all samples in the set $\{(\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj}),k\in\mathcal{K}\}$ share the same score. Accordingly, we define $o(s_j,l_v,\boldsymbol{u}_k)$ as the collective score for this set. Based on Theorem 4 in [31], the classification margin of the set $\{(\boldsymbol{x}_{kj}(\boldsymbol{u}_k),y_{kj}),\forall k\in\mathcal{K}\}$ is

$$\Delta(s_j,l_v,\{\boldsymbol{u}_k\})\geq\frac{o(s_j,l_v,\{\boldsymbol{u}_k\})}{\sup\limits_{\boldsymbol{x}\in\Omega}\|\mathbf{J}(\boldsymbol{x},l_v)\|_2},\qquad(18)$$

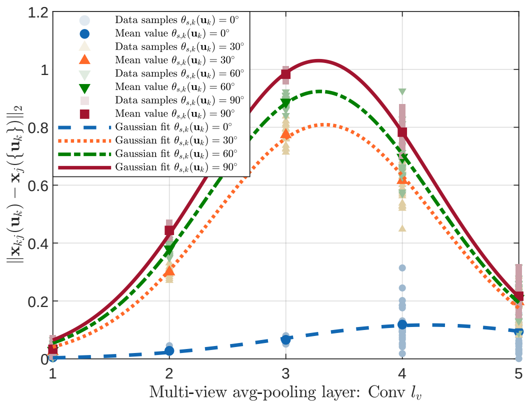

Fig. 2. The disturbance vs. average-pooling layer l_v under different elevation angle $\theta_{s,k}(\boldsymbol{u}_k)$.

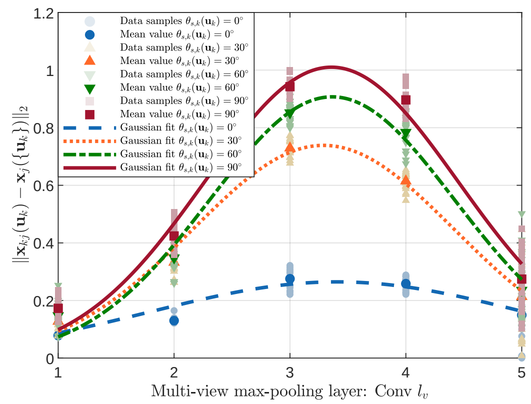

Fig. 3. The disturbance vs. max-pooling layer l_v under different elevation angle $\theta_{s,k}(\boldsymbol{u}_k)$.

where $\mathbf{J}(\boldsymbol{x}, l_v) = \prod_{l=l_v+1}^{L} \frac{d\,\phi_l(\boldsymbol{w}_{l-1}, l_v)}{d\,\boldsymbol{w}_{l-1}}$ denote the JM of the server-side sub-model. Here, $\phi_l(\cdot)$ is the mapping implemented by the l-th layer, and $\boldsymbol{w}_l$ denotes the output at layer l.

Next, we introduce a reference score $o((\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj}), l')$ to facilitate deriving the closed-form in (18). This score is the output for the sample $(\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj})$ produced by the reference model without view pooling layer , i.e., $l' = 0$. Then, (18) can be reformulated as

$$\begin{aligned}&\Delta(s_j, l_v, \{\boldsymbol{u}_k\})\\ &\geq \frac{o\left((\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj}), l'\right) + o(s_j, l_v, \{\boldsymbol{u}_k\}) - o\left((\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj}), l'\right)}{\sup\limits_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l_v)\|_2}. \quad (19)\end{aligned}$$

Substituting (17) into (19), we have

$$\begin{aligned}&o(s_j, l_v, \{\boldsymbol{u}_k\}) - o\left((\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj}), l'\right)\\ &= \min_{\tau\neq y_j} \sqrt{2}(\delta_{y_j} - \delta_\tau)^T (c(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}), l_v) - c(\boldsymbol{x}_{kj}(\boldsymbol{u}_k), l')). \quad (20)\end{aligned}$$

Moreover, (20) is equivalent to

$$\begin{aligned}&-\max_{\tau\neq y_j} \sqrt{2}(\delta_{y_j} - \delta_\tau)^T (c(\boldsymbol{x}_{kj}(\boldsymbol{u}_k), l') - (c(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}), l') + \boldsymbol{\xi}))\\ &\geq -\max_{\tau\neq y_j} \|\sqrt{2}(\delta_{y_j} - \delta_\tau)^T\|_2 \|c(\boldsymbol{x}_{kj}(\boldsymbol{u}_k), l') - c(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}), l') - \boldsymbol{\xi}\|_2\\ &\geq -\max_{\tau\neq y_j} \|\sqrt{2}(\delta_{y_j} - \delta_\tau)^T\|_2 (\|c(\boldsymbol{x}_{kj}(\boldsymbol{u}_k), l') - c(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}), l')\|_2 + \|\boldsymbol{\xi}\|_2)\\ &\geq -2\|\boldsymbol{\xi}\|_2 - 2\sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l')\|_2 \, \|\boldsymbol{x}_{kj}(\boldsymbol{u}_k) - \boldsymbol{x}_j(\boldsymbol{u}_k)\|_2. \quad (21)\end{aligned}$$

The above approximation first is obtained by setting $c(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}), l_v) = c(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}), l') + \boldsymbol{\xi}$, where $\boldsymbol{\xi}$ captures the logit discrepancy between the model with view-pooling layer l_v and the reference model. Then, it follows directly from the Cauchy-Schwarz inequality, $\boldsymbol{a}^\top \boldsymbol{b} \leq \|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2$. Next, it follows from the triangle inequality, $\|\boldsymbol{a} + \boldsymbol{b}\|_2 \leq \|\boldsymbol{a}\|_2 + \|\boldsymbol{b}\|_2$. According to Corollary 2 in [31], the difference between the model outputs satisfies $\|c(\boldsymbol{x}_{kj}(\boldsymbol{u}_k), l') - c(\boldsymbol{x}_j(\{\boldsymbol{u}_k\}), l')\|_2 \leq \sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l')\|_2 \, \|\boldsymbol{x}_{kj}(\boldsymbol{u}_k) - \boldsymbol{x}_j(\boldsymbol{u}_k)\|_2$, which yields (21). Moreover, $o((\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj}), l')$, $2\|\boldsymbol{\xi}\|_2$, and $2\sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l')\|_2$ are constants that can be determined via simulation. Since the lower bound in (21) is derived over the set $\{(\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj}), \forall k \in \mathcal{K}\}$, all data samples must be considered. Thus, the classification margin for the entire dataset is (16), where $A = \min_{\boldsymbol{x}_{kj}(\boldsymbol{u}_k)\in\Omega} o\left((\boldsymbol{x}_{kj}(\boldsymbol{u}_k), y_{kj}), l'\right) - 2\|\boldsymbol{\xi}\|_2$, $B = 2\sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l')\|_2$. □

Remark 1. *From* (16)*, the classification margin is dominated by two terms: 1) The disturbance* $\max_{\boldsymbol{x}_{kj}(\boldsymbol{u}_k)\in\Omega} \|\boldsymbol{x}_{kj}(\boldsymbol{u}_k) - \boldsymbol{x}_j(\{\boldsymbol{u}_k\})\|_2$ *quantifies the fidelity of the feature representation after the view-pooling layer. Smaller values indicate that the pooled features preserve more information from the inputs and are therefore more representative. It is determined by the data collected by UAVs at position* $\{\boldsymbol{u}_k\}$*. 2) The supremum of the Jacobian norm* $\sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, \ell_v)\|_2$ *characterizes the worst-case local sensitivity of the model. Smaller values correspond to increased robustness to infinitesimal input perturbations. Otherwise, the model is more sensitive (i.e., less robust). It depends on the model, particularly on where view-pooling layer* l_v *is applied.*

Although the key factors shaping the classification margin are identified, establishing a closed-form relationship among l_v, $\{\boldsymbol{u}_k\}$, and the margin remains nontrivial, given the lack of explicit analytical expressions. Accordingly, we complement our theoretical discussion with qualitative analysis and use simulations to elucidate the functional dependencies.

1) $\max_{\boldsymbol{x}_{kj}(\boldsymbol{u}_k)\in\Omega} \|\boldsymbol{x}_{kj}(\boldsymbol{u}_k) - \boldsymbol{x}_j(\{\boldsymbol{u}_k\})\|_2$*:* From (16), a smaller value of this term increases the lower bound of $\Delta(l_v, \{\boldsymbol{u}_k\})$, thereby enhancing the model's ability to separate classes. We evaluate multiple view-pooling strategies, including average-pooling and max-pooling, spanning multi-view data generation, model training, and curve fitting (see Appendix in the online supplemental material for details).

Fig. 2 and Fig. 3 illustrate how $\max_{\boldsymbol{x}_{kj}(\boldsymbol{u}_k)\in\Omega} \|\boldsymbol{x}_{kj}(\boldsymbol{u}_k) - \boldsymbol{x}_j(\{\boldsymbol{u}_k\})\|_2$ varies with the view-pooling layer l_v and the sensing elevation angle $\theta_{s,k}(\boldsymbol{u}_k) = 90^\circ - \frac{180^\circ}{\pi} \times \sin^{-1}|\frac{z_{u,k} - z_v}{d_0}|$, where the target is located at $\boldsymbol{v} = (x_v, y_v, z_v)$, z_k and z_v denote the altitudes of UAV k and the target, respectively, and d_0 denotes the distance between UAV k and the target. The results reveal two key observations: 1) It increases at first and

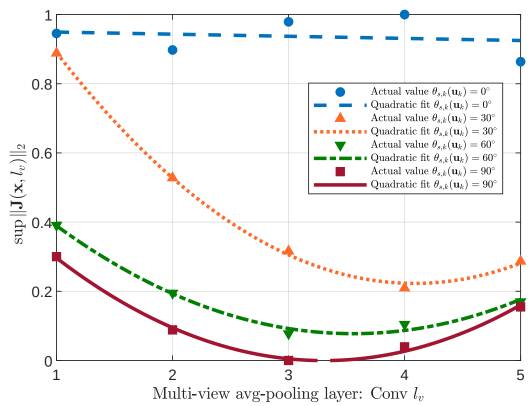

Fig. 4. The supremum of the Jacobian norm vs. average-pooling layer l_v under different elevation angle $\theta_{s,k}(\boldsymbol{u}_k)$.

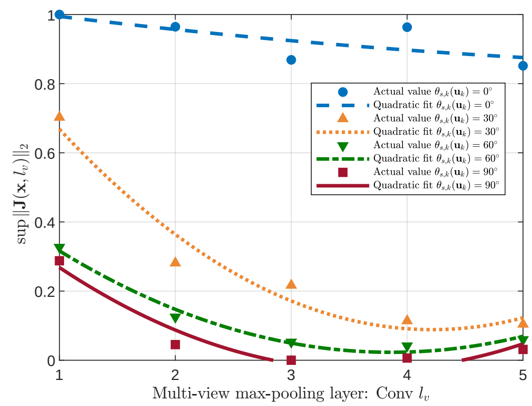

Fig. 5. The supremum of the Jacobian norm vs. max-pooling layer l_v under different elevation angle $\theta_{s,k}(\boldsymbol{u}_k)$.

then decreases as l_v moves deeper. This indicates that both early ($l_v = \text{Conv1}$) and late ($l_v = \text{Conv5}$) view-pooling layers yield more representative features: shallow layers preserve richer input details, while deeper layers capture higher-level abstractions[5]. 2) It decreases as $\theta_{s,k}(\boldsymbol{u}_k)$ decreases. Smaller angles lead to more similar multi-view images, reducing cross-view diversity and thus lowering the disturbance. In the limiting case $\theta_{s,k}(\boldsymbol{u}_k) = 0°$, the multi-view setting collapses to a single view, making the features nearly identical. Based on these observations, the disturbance is well approximated by a Gaussian function

$$\max_{\boldsymbol{x}_{kj}(\boldsymbol{u}_k)\in\Omega} \|\boldsymbol{x}_{kj}(\boldsymbol{u}_k) - \boldsymbol{x}_j(\{\boldsymbol{u}_k\})\|_2$$
$$\triangleq \alpha_{Dist} \exp\left(-\left(\frac{l_v - \beta_{Dist}}{\gamma_{Dist}}\right)^2\right), \tag{23}$$

where $\alpha_{\text{Dist}} > 0$, $\beta_{\text{Dist}} > 0$, and $\gamma_{\text{Dist}} > 0$ are fitting parameters dependent on $\theta_{s,k}(\boldsymbol{u}_k)$. For any fixed $\theta_{s,k}(\boldsymbol{u}_k)$, they can be estimated via minimum mean square error (MMSE) method.

2) $\sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l_v)\|_2$*:* From (16), a smaller value of this term raises the lower bound of $\Delta(l_v, \{\boldsymbol{u}_k\})$, thereby enhancing the model's ability to separate classes. We empirically estimate the supremum of the Jacobian norm (see Appendix in the online supplemental material for details).

Fig. 4 and Fig. 5 show how $\sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l_v)\|_2$ varies with l_v and $\theta_{s,k}(\boldsymbol{u}_k)$. The results reveal two key observations: 1) It first decreases and then increases as l_v deepens. 2) It decreases as $\theta_{s,k}(\boldsymbol{u}_k)$ increases[6]. As $\theta_{s,k}(\boldsymbol{u}_k)$ increases, reflecting greater multi-view diversity, these results indicate that training with more diverse samples enhances generalization and robustness. Based on these observations, the supremum of Jacobian norm is well approximated by a quadratic function

$$\sup_{\boldsymbol{x}\in\Omega} \|\mathbf{J}(\boldsymbol{x}, l_v)\|_2 \triangleq \alpha_{\text{JM}} \times l_v^2 + \beta_{\text{JM}} \times l_v + \gamma_{\text{JM}}, \tag{24}$$

where $\alpha_{\text{JM}} \geq 0$, β_{JM}, and $\gamma_{\text{JM}} > 0$ are fitting parameters dependent on $\theta_{s,k}(\boldsymbol{u}_k)$. For any fixed $\theta_{s,k}(\boldsymbol{u}_k)$, they can be estimated via the MMSE method.

Next, substituting (23) and (24) into (16), we obtain the lower bound on the classification margin

$$\Delta(l_v, \{\boldsymbol{u}_k\}) > \frac{A - B \times \alpha_{\text{Dist}} \times \exp\left(-\left(\frac{l_v - \beta_{\text{Dist}}}{\gamma_{\text{Dist}}}\right)^2\right)}{\alpha_{\text{JM}} \times l_v^2 + \beta_{\text{JM}} \times l_v + \gamma_{\text{JM}}}. \tag{25}$$

D. Inference Accuracy

Leveraging (12) for the quantization error and (25) for the classification margin, we derive the lower bound on the classification accuracy in (11) as (22). We approximate $\hat{R}_0$ by the minimum of $R_0(l_v, \{\boldsymbol{u}_k\})$ obtained from simulations. Moreover, due to model normalization leading to comparable output-feature statistics across layers, we adopt $(\overline{f}_{k,n} - \underline{f}_{k,n})^2 = (f_{\max} - f_{\min})^2$, $\forall l_s \in \mathcal{L}$, $\forall k \in \mathcal{K}$ [16]. This assumption supports the simplified inference accuracy expression in (22).

[5]For analytical convenience, we assume that both splitting and view pooling are applied after the convolutional blocks in the MVCNN model; that is, $l_s, l_v \in \mathcal{L} = \{\text{Conv1}, \text{Conv2}, \ldots\}$ [21].

[6]It depends on the training dataset; our analysis extends to other settings.

$$R_{\text{view}}(l_s, l_v, q, \{\boldsymbol{u}_k\}) \geq \hat{R}_0 \left(1 - \frac{N(l_s)(f_{\max} - f_{\min})^2}{4(2^q - 1)^2} \times \frac{\sigma(\alpha_{\text{JM}} l_v^2 + \beta_{\text{JM}} l_v + \gamma_{\text{JM}})^2}{\left(A - B \times \alpha_{\text{Dist}} \exp\left(-\left(\frac{l_v - \beta_{\text{Dist}}}{\gamma_{\text{Dist}}}\right)^2\right)\right)^2}\right). \tag{22}$$

V. Problem Formulation and Optimization

In this section, we formulate the inference latency minimization problem and solve it via alternating optimization. In Section V-A, we present the formulation that minimizes the inference latency subject to an inference accuracy constraint. In Sections V-B, V-C, and V-D, we decompose the problem into three subproblems and sequentially optimize their respective variables. In Section V-E, we introduce the split, view-pooling, quantization, and position (SVQP) optimization scheme, which efficiently yields a suboptimal solution.

A. Problem Formulation

We aim to minimize the inference latency subject to an accuracy threshold by jointly optimizing split layer l_s, view-pooling layer l_v, quantization level q, UAV position $\{\boldsymbol{u}_k\}$, transmission power $\{p_k\}$, and computation capacity $\{v_k\}$.

$$\textbf{P1:} \min_{\substack{l_s, l_v, q,\\ \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\}}} T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$$

$$\text{s.t.} \quad R_{\text{view}}(l_s, l_v, q, \{\boldsymbol{u}_k\}) \geq R_c, \tag{26a}$$

$$|\varphi_{s,k}(\boldsymbol{u}_k) - \varphi_{s,k+1}(\boldsymbol{u}_{k+1})| = \frac{2\pi}{K}, k \in \mathcal{K} \backslash \{K\}, \tag{26b}$$

$$\theta_{s,k}(\boldsymbol{u}_k) = \theta_{s,k'}(\boldsymbol{u}_{k'}), \forall k, k' \in \mathcal{K}, \tag{26c}$$

$$p_k \leq P_{k,\max}, \forall k \in \mathcal{K}, \tag{26d}$$

$$v_k \leq V_{k,\max}, \forall k \in \mathcal{K}, \tag{26e}$$

$$l_s \leq l_v, \forall l_s, l_v \in \mathcal{L}, \tag{26f}$$

$$q \in \mathbb{Z}^+. \tag{26g}$$

Constraint (26a) enforces that the inference accuracy $R_{\text{view}}(l_s, l_v, q, \{\boldsymbol{u}_k\})$ meets the threshold R_c. Constraint (26b) ensures that the UAVs are uniformly distributed around the target, i.e., the azimuth angle $\varphi_{s,k}(\boldsymbol{u}_k)$ between UAV k and the target are equally spaced at intervals of $\frac{2\pi}{K}$ radians. Constraint (26c) imposes that the elevation angle $\theta_{s,k}(\boldsymbol{u}_k)$ between UAV k and the target are identical for all k. Constraints (26d) and (26e) bound each UAV's transmit power and computation capacity by $P_{k,\max}$ and $V_{k,\max}$. Constraint (26f) restricts the choices of split layer l_s and view-pooling layer l_v, and (26g) enforces quantization level q to be an integer variable.

The objective in **P1** is consistent with reducing the energy consumption of UAVs. The reason is that, during inference, the energy consumption of onboard computation and wireless communication is usually on the order of a few watts, which is much smaller than the hovering power of UAVs, typically on the order of hundreds of watts [32]. Since each UAV hovers at its optimized position with a fixed power setting during the inference process, the total UAV energy consumption is mainly dominated by the hovering duration. Therefore, minimizing the inference latency can effectively reduce the energy consumption of UAVs.

Moreover, the objective and constraint (26a) in **P1** are complicated by the coupling among l_s, l_v, q, and $\{\boldsymbol{u}_k\}$, with q further restricted to integers. Therefore, **P1** is a mixed-integer non-convex problem, making global optimization challenging. To address this, we adopt an alternating optimization approach, sequentially optimizing the decision variables to minimize the objective. Next, we detail how these variables impact inference accuracy and latency.

Algorithm 1: Split and View-pooling Layer Optimization

Input: Quantization level q; UAV position $\{\boldsymbol{u}_k\}$; transmission power $p_k^* = P_{k,\max}, \forall k \in \mathcal{K}$; computation capacity $v_k^* = V_{k,\max}, \forall k \in \mathcal{K}$; threshold R_c; probability $\hat{R}_0$; parameters $A, B, \alpha_{\text{Dist}}, \beta_{\text{Dist}}, \gamma_{\text{Dist}}, \alpha_{\text{JM}}, \beta_{\text{JM}}, \gamma_{\text{JM}}$;

```
for l_v ∈ L do
    Initialize iteration index j;
    Compute lower bound of l_s as l_low^j based on (28);
    Set upper bound of l_s as l_up^j based on (26f);
    if l_low^j ≤ l_up^j then
        Determine the optimal l_s^j;
        Calculate inference latency
          T^j(l_s^j, q, {u_k}, {p_k^*}, {v_k^*});
Determine l_s^* = l_s^j, l_v^* = l_v^j,
  j = argmin_{j=1,...,L} T^j(l_s^j, q, {u_k}, {p_k^*}, {v_k^*});
```

Output: Optimal split layer l_s^*, view-pooling layer l_v^*.

Remark 2. ***The tradeoff between inference accuracy and inference latency.*** *From* (22)*, the split layer* l_s*, quantization level* q*, and UAV position* $\{\boldsymbol{u}_k\}$ *jointly determine the accuracy* $R_{\textit{view}}(l_s, l_v, q, \{\boldsymbol{u}_k\})$ *and the latency* $T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$*. 1) Increasing either* l_s *or* q *mitigates quantization error and thus enhances accuracy. However, a larger* l_s *relocates more layers to the onboard submodel, elevating the UAV's computation burden and incurring longer UAV inference time. While a higher* q *expands the feature payload size, prolonging uplink transmission and overall delay. 2) The well-chosen* $\{\boldsymbol{u}_k\}$ *that captures more diverse sensing views can boost accuracy. However, such viewpoints may degrade the uplink channel to the server, thereby increasing communication time. 3) The view-pooling layer* l_v *also affects this tradeoff. An early* l_v *may fuse low-level features with limited semantic information, degrading accuracy, whereas a late* l_v *requires deeper per-view feature extraction and increases the UAV computation burden. Moreover, due to the constraint* $l_s \leq l_v$*,* l_v *further affects the feasible split-layer selection and hence the computation and communication time. Consequently, there is an accuracy-latency tradeoff: achieving higher inference accuracy typically requires greater computation and communication resources, thereby increasing latency.*

B. Split Layer and View-pooling Layer Optimization with Fixed Quantization Level and Position

We consider the subproblem of **P1** for optimizing the split layer l_s, view-pooling layer l_v, transmission power $\{p_k\}$, and computation capacity $\{v_k\}$ by assuming that quantization level q and UAV position $\{\boldsymbol{u}_k\}$ are fixed. Given that $\{\boldsymbol{u}_k\}$ and q

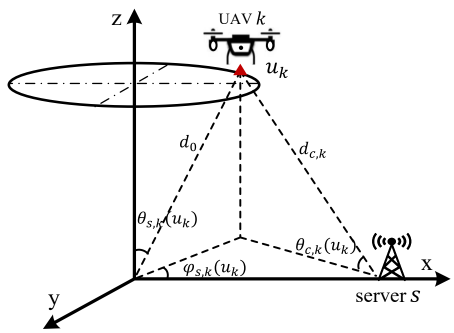

Fig. 6. An illustration of the UAV position $\boldsymbol{u}_k$ and its geometric relationship to the server position s in 3D space.

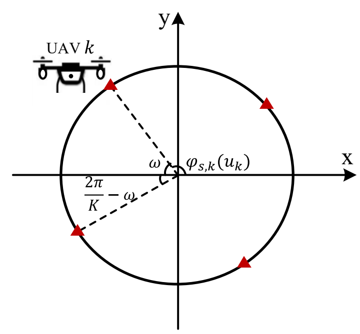

Fig. 7. An illustration of the optimal UAV azimuth angle $\varphi_{s,k}(\boldsymbol{u}_k)$ in 2D space, where $\omega = \frac{\pi}{K}$.

satisfy the constraints (26b), (26c), and (26g), **P1** reduces to

$$\textbf{P2:} \quad \min_{l_s, l_v, \{p_k\}, \{v_k\}} T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$$
$$\text{s.t.} \quad \text{constraints (26a), (26d), (26e), (26f).}$$

From (5) and (6), the computation time $T_k^{cp}(l_s, v_k)$ and transmission time $T_k^{cm}(l_s, q, \boldsymbol{u}_k, p_k)$ for UAV k are both decreasing in v_k and p_k, respectively. Hence, the minimum inference delay is attained by choosing $p_k^* = P_{k,\max}, v_k^* = V_{k,\max}, \forall k \in \mathcal{K}$. By (22), constraint (26a) induces a lower bound on l_s as

$$l_s \geq N^{-1}\left(\left(1 - \frac{R_c}{\hat{R}_0}\right) \times \frac{4(2^q-1)^2}{(f_{\max} - f_{\min})^2} \times \frac{\left(A - B \times \alpha_{\text{Dist}} \exp\left(-\left(\frac{l_v - \beta_{\text{Dist}}}{\gamma_{\text{Dist}}}\right)^2\right)\right)^2}{\sigma(\alpha_{\text{JM}} l_v^2 + \beta_{\text{JM}} l_v + \gamma_{\text{JM}})^2} \right) \tag{28}$$

where $N^{-1}(\cdot)$ is the inverse of the function $N(l_s)$[7]. Additionally, the constraint $l_s \leq l_v$ in (26f) imposes an upper bound on l_s. From (5)-(7), the objective explicitly depends on l_s. Thus, for any fixed l_v, the optimal l_s can be determined. However, since (28) involves both exponential and quadratic terms in l_v, solving **P2** directly is nontrivial. To tackle this, we propose Algorithm 1, which efficiently computes a suboptimal solution for l_s and l_v. Specifically, we perform a one-dimensional search over l_v. For each l_v, we determine the feasible range for l_s by evaluating the lower and upper bounds, denoted l_{low}^j and l_{up}^j, using (28) and (26f). Within this interval, we solve for the optimal l_s corresponding to the fixed l_v.

Lemma 2. *The split and view-pooling operations can occur at the same layer, i.e., $l_s^* = l_v^*$, when the classification accuracy threshold satisfies*

$$R_c \leq \hat{R}_0 \left(1 - \frac{N(\hat{l}_s)(f_{max} - f_{min})^2 \sigma}{4\Delta(\hat{l}_v, \{\hat{\boldsymbol{u}}_k\})^2}\right), \tag{29}$$

where $\hat{l}_s = \text{Conv1}$, and $\hat{l}_v, \{\hat{\boldsymbol{u}}_k\} = \arg\min_{l_v, \{\boldsymbol{u}_k\}} \Delta(l_v, \{\boldsymbol{u}_k\})$.

Proof. The minimum achievable lower bound of $R_{\text{view}}(l_s, l_v, q, \{\boldsymbol{u}_k\})$ is $\hat{R}_0\left(1 - \frac{N(\hat{l}_s)(f_{\max} - f_{\min})^2\sigma}{4\Delta(\hat{l}_v, \{\hat{\boldsymbol{u}}_k\})^2}\right)$, which occurs when $q = 1$, $\hat{l}_s$ =Conv1, and $\hat{l}_v, \{\hat{\boldsymbol{u}}_k\} = \arg\min_{l_v, \{\boldsymbol{u}_k\}} \Delta(l_v, \{\boldsymbol{u}_k\})$. Therefore, when R_c does not exceed this lower bound, any choice of l_s, l_v satisfies the inference accuracy constraint (26a). The case $l_v^* = l_s^*$ is also a potentially optimal solution to **P2**. □

[7]The explicit form of $N(\cdot)$ and $N^{-1}(\cdot)$ can be defined and obtained from the MVCNN architecture.

C. Quantization Level Optimization with Fixed Split and View-pooling Layer and Position

We consider the subproblem of **P1** for optimizing quantization level q by assuming that split layer l_s, view-pooling layer l_v and UAV position $\{\boldsymbol{u}_k\}$ are fixed. Given that $\{\boldsymbol{u}_k\}$, $\{p_k\}$, $\{v_k\}$, and l_s, l_v satisfy constraints (26b)-(26f), **P1** reduces to

$$\textbf{P3:} \quad \min_{q} \quad T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$$
$$\text{s.t.} \quad q > 0, \tag{30a}$$
$$\text{constraint (26a),}$$

where we relax the value of q from integer to real number $q > 0$. Based on (22), the corresponding closed-form lower bound for q can be expressed as

$$q \geq \log_2 \left(1 + \sqrt{\frac{\sigma N(l_s)\hat{R}_0}{(\hat{R}_0 - R_c)}} \frac{(f_{\max} - f_{\min})(\alpha_{\text{JM}} l_v^2 + \beta_{\text{JM}} l_v + \gamma_{\text{JM}})}{2\left(A - B \times \alpha_{\text{Dist}} \exp\left(-\left(\frac{l_v - \beta_{\text{Dist}}}{\gamma_{\text{Dist}}}\right)^2\right)\right)}\right). \tag{31}$$

Since $T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$ is a monotonically increasing function of q, the optimal q^* for **P3** is achieved when the equality in (31) is satisfied.

D. Position Optimization with Fixed Split and View-pooling Layer and Quantization Level

We consider the subproblem of **P1** for optimizing UAV position $\{\boldsymbol{u}_k\}$ by assuming that split layer l_s, view-pooling layer l, and quantization level q are fixed. Given that $\{p_k\}$, $\{v_k\}$, l_s, l_v, q satisfy constraints (26d)-(26g), **P1** reduces to

$$\textbf{P4:} \quad \min_{\{\boldsymbol{u}_k\}} \quad T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$$
$$\text{s.t.} \quad \text{constraint (26a)} - \text{(26c).}$$

The complexity of **P4** primarily arises from the dependence on the azimuth angle $\varphi_{s,k}(\boldsymbol{u}_k)$ between each UAV and the target

Algorithm 2: UAV Position Optimization

Input: Split layer l_s; view-pooling layer l_v; quantization level q; transmission power $p_k^* = P_{k,\max}, \forall k \in \mathcal{K}$; computation capacity $v_k^* = V_{k,\max}, \forall k \in \mathcal{K}$; threshold R_c; probability R_0; parameters $A, B, \alpha_{\text{Dist}}, \beta_{\text{Dist}}, \gamma_{\text{Dist}}, \alpha_{\text{JM}}, \beta_{\text{JM}}, \gamma_{\text{JM}}$; step size ϑ;

Initialize the inference time $T_{\min}$ to infinity.;
for $\theta_{s,k}(\boldsymbol{u}_k) = 0 : \vartheta_1 : \frac{\pi}{2}$ **do**
  // Homogeneous UAVs
  Compute the optimal azimuth angle $\{\varphi_{s,k}(\boldsymbol{u}_k)\}$ using (34) in Proposition 2;
  Compute $T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$ using (8);
  if $T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\}) < T_{\min}$ **then**
    Update $T_{\min} = T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$ and $\{\boldsymbol{u}_k^*\} = \{\boldsymbol{u}_k\}$;
  end
  // Heterogeneous UAVs
  for $\varphi_{s,1}(\boldsymbol{u}_1) = 0 : \vartheta_2 : \frac{2\pi}{K}$ **do**
    Find the azimuth angles $\varphi_{s,k}(\boldsymbol{u}_k) = \varphi_{s,1}(\boldsymbol{u}_1) + (k-1) \times \frac{2\pi}{K}, \forall k \in \mathcal{K} \notin \{1\}$;
    Compute $T_k^{cp}(l_s, v_k)$ and $T_k^{cm}(l_s, q, \boldsymbol{u}_k, p_k)$ using (5), (6);
    Sort $\{T_k^{cp}(l_s, v_k)\}$ and $\{T_k^{cm}(l_s, q, \boldsymbol{u}_k, p_k)\}$, map highest $\{T_k^{cp}\}$ to lowest $\{T_k^{cm}\}$;
    Compute $T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$ using (8);
    if $T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\}) < T_{\min}$ **then**
      Update $T_{\min} = T(l_s, q, \{\boldsymbol{u}_k\}, \{p_k\}, \{v_k\})$ and $\{\boldsymbol{u}_k^*\} = \{\boldsymbol{u}_k\}$;
    end
  end
end

Output: Optimal UAV position $\{\boldsymbol{u}_k^*\}$.

in constraint (26b). However, if all UAVs are homogeneous, the problem admits further simplifications. Accordingly, we consider two cases:

1) Homogeneous UAVs: Assume that all UAVs have identical transmission power and computation capacity, i.e., $P_{k,\max} = P_{k',\max}, V_{k,\max} = V_{k',\max}, \forall k, k' \in \mathcal{K}$. Then, following Section V-B, the optimal choices for each UAV are $p_k^* = P_{k,\max}$ and $v_k^* = V_{k,\max}$. Under these settings, the UAV computation time $T_k^{cp}(l_s, v_k)$ (see (5)) is identical across all k, and the server computation time $T_s^{cp}(l_s)$ (see (7)) is a constant. With l_s, l_v, q, $\{p_k\}$, and $\{v_k\}$ fixed, the only component of the objective that depends on $\{\boldsymbol{u}_k\}$ is the transmission rate $r_k(\boldsymbol{u}_k, p_k)$ appearing in the communication time $T_k^{\text{cm}}(l_s, q, \boldsymbol{u}_k, p_k)$ (see (6)). Thus, **P4** is equivalent to

$$\textbf{P4'}: \quad \max_{\{\boldsymbol{u}_k\}} \min_{k \in \mathcal{K}} \; r_k(\boldsymbol{u}_k, p_k) \qquad \text{s.t.} \quad \text{constraint (26a)} - \text{(26c)}.$$

Since the objective depends on the azimuth angle $\varphi_{s,k}(\boldsymbol{u}_k)$ through the transmission rate $r_k(\boldsymbol{u}_k, p_k)$, we introduce the following lemma to characterize this dependence.

Lemma 3. *For a given $\theta_{s,k}(\boldsymbol{u}_k)$, the transmission rate $r_k(\boldsymbol{u}_k, p_k)$ is a monotonically decreasing function of $\varphi_{s,k}(\boldsymbol{u}_k)$ over $[0, \pi)$, and a monotonically increasing function of $\varphi_{s,k}(\boldsymbol{u}_k)$ over $[\pi, 2\pi)$.*

Proof. As shown in Fig. 6, the squared distance between UAV k and the server is $d_{c,k}^2 = \rho^2 \cos^2(\theta_{s,k}(\boldsymbol{u}_k)) + \rho^2 \sin^2(\theta_{s,k}(\boldsymbol{u}_k)) + \|\boldsymbol{s} - \boldsymbol{v}\|^2 - 2\rho \sin(\theta_{s,k}(\boldsymbol{u}_k)) \times \|\boldsymbol{s} - \boldsymbol{v}\| \times \cos(\varphi_{s,k}(\boldsymbol{u}_k))$, and the elevation angle $\theta_{c,k}(\boldsymbol{u}_k) = \arctan\left(\frac{\rho \cos(\theta_{s,k}(\boldsymbol{u}_k))}{\rho^2 \sin^2(\theta_{s,k}(\boldsymbol{u}_k)) + \|\boldsymbol{s}-\boldsymbol{v}\|^2 - 2\rho \sin(\theta_{s,k}(\boldsymbol{u}_k))\|\boldsymbol{s}-\boldsymbol{v}\| \cos(\varphi_{s,k}(\boldsymbol{u}_k))}\right)$. It follows that as $\varphi_{s,k}(\boldsymbol{u}_k)$ increases, $d_{c,k}^2$ decreases and $\theta_{c,k}(\boldsymbol{u}_k)$ increases. According to (3) and (4), $r_k(\boldsymbol{u}_k, p_k)$ decreases monotonically as $\varphi_{s,k}(\boldsymbol{u}_k)$ increases over $\varphi_{s,k}(\boldsymbol{u}_k) \in [0, \pi)$, and increases monotonically as $\varphi_{s,k}(\boldsymbol{u}_k)$ increases over $\varphi_{s,k}(\boldsymbol{u}_k) \in [\pi, 2\pi)$. □

Based on Lemma 3, we present the following proposition regarding the optimal solution.

Proposition 2. *The optimal solution to **P4'** is achieved when one of $\{\boldsymbol{u}_k\}$ satisfies*

$$\varphi_{s,k}(\boldsymbol{u}_k) = \begin{cases} \frac{2\pi}{K} \times k - \frac{\pi}{K}, & \text{if } K \text{ is even}, \\ \frac{2\pi}{K} \times (k-1), & \text{if } K \text{ is odd}. \end{cases} \tag{34}$$

It highlights that the optimal $\{\boldsymbol{u}_k\}$ depends on whether the number of UAVs K is even or odd, leading to distinct requirements for their angular placement.

Proof. Based on Lemma 3, **P4'** can be reformulated as $\min_{\{\boldsymbol{u}_k\}} \max_{k \in \mathcal{K}} \varphi_{s,k}(\boldsymbol{u}_k)$ for $\varphi_{s,k}(\boldsymbol{u}_k) \in [0, \pi)$, and as $\max_{\{\boldsymbol{u}_k\}} \min_{k \in \mathcal{K}} \varphi_{s,k}(\boldsymbol{u}_k)$ for $\varphi_{s,k}(\boldsymbol{u}_k) \in [\pi, 2\pi)$. As shown in Fig. 7, the optimal value of $\{\varphi_{s,k}(\boldsymbol{u}_k)\}$ is achieved when $\omega = \frac{\pi}{K}$. Therefore, the optimal $\varphi_{s,k}(\boldsymbol{u}_k)$ is given by (34). □

2) Heterogeneous UAVs: Assume that UAVs have heterogeneous transmission power and computation capacity. When $P_{k,\max}$ and $V_{k,\max}$ differ among UAVs, **P4** becomes highly nonlinear due to the coupling introduced by $\varphi_{s,k}(\boldsymbol{u}_k)$. To address this, we propose Algorithm 2. We begin with a one-dimensional discrete search over $\varphi_{s,1}(\boldsymbol{u}_1)$ for UAV 1, then compute $\varphi_{s,k}(\boldsymbol{u}_k)$ for all $k \in \mathcal{K} \setminus \{1\}$. Next, we determine $\{\boldsymbol{u}_k\}$, enabling calculation of both UAV computation and transmission times. We then pair the UAV with the highest computation time T_k^{cp} to the lowest transmission time T_k^{cm} to minimize the total inference time. Finally, we record $\{\boldsymbol{u}_k\}$ and $T_{\min}$ for each $\varphi_{s,k}(\boldsymbol{u}_k)$ and select the optimal $\{\boldsymbol{u}_k^*\}$.

The proposed SVQP scheme determines the optimized UAV positions for inference, Then existing trajectory planning or collision-avoidance methods, such as waypoint planning and safe-distance-constrained trajectory optimization, can be adopted to guide the UAVs to these positions safely [33], [34].

E. Iterative Split, View-pooling, Quantization, and Position Optimization (SVQP)

Based on the solutions to the three subproblems discussed in the preceding sections, we propose an iterative algorithm for **P1**, as summarized in Algorithm 3. Specifically, we alternately solve subproblems **P2**, **P3**, and **P4**, which ensures convergence

Algorithm 3: Iterative Split, View-pooling, Quantization, Position Optimization (SVQP)

Input: UAV set $\mathcal{K}$; transmission power $\{P_{k,\max}\}$; computation capacity $\{V_{k,\max}\}$;

Set the iteration index of the following three steps to $i = 0$;

Initialize $\{\boldsymbol{u}_k^{(i)}\}$ and $q^{(i)}$ to satisfy constraints (26b), (26c), and (26g);

repeat

  Fix $q^{(i)}$ and $\{\boldsymbol{u}_k^{(i)}\}$, optimize split point $l_s^{(i)}$, view-pooling layer $l_v^{(i)}$, transmission power $\{p_k^{(i)}\}$, and computation capacity $\{v_k^{(i)}\}$ by solving **P2** in Section V-B;

  Fix $l_s^{(i)}, l_v^{(i)}, \{\boldsymbol{u}_k^{(i)}\}, \{p_k^{(i)}\}, \{v_k^{(i)}\}$, optimize $q^{(i)}$ by solving **P3** in Section V-C;

  Fix $l_s^{(i)}, l_v^{(i)}, q^{(i)}, \{p_k^{(i)}\}, \{v_k^{(i)}\}$, optimize $\{\boldsymbol{u}_k^{(i)}\}$ by solving **P4** in Section V-D;

  Let $T^{(i)}$ be the objective value of **P4**;

  Update $i = i + 1$;

until $|T^{(i)} - T^{(i-1)}|/T^{(i-1)} \le \tau$;

Output: $l_s^* = l_s^{(i)}$, $l_v^* = l_v^{(i)}$, $q^* = \lceil q^{(i)} \rceil$, $\{\boldsymbol{u}_k^*\} = \{\boldsymbol{u}_k^{(i)}\}$, and $p_k^* = p_k^{(i)}, v_k^* = v_k^{(i)}, \forall k \in \mathcal{K}$.

within a finite number of iterations [35]. Therefore, we state the following proposition.

Proposition 3. ***P1*** *can be efficiently solved by the SVQP scheme (i.e., Algorithm 3), which achieves a polynomial time complexity of* $\mathcal{O}\left(I\left(L + \frac{\pi}{2\vartheta_1} \times K\right)\right)$ *for homogeneous UAVs, and* $\mathcal{O}\left(I\left(L + \frac{\pi^2}{\vartheta_1\vartheta_2}\right)\right)$ *for heterogeneous UAVs.*

Proof. Suppose the alternating optimization algorithm requires I iterations to converge. In each iteration, we solve **P2**-**P4** with the following computational complexities: 1) **P2** can be solved with time complexity $\mathcal{O}(L)$. This is because we perform a one-dimensional search over $l_v \in \mathcal{L}$, while the optimal l_s is obtained analytically. 2) According to (31), **P3** reduces to a linear problem in q, which can be solved directly with time complexity $\mathcal{O}(1)$. 3) Solving **P4** involves two cases. For homogeneous UAVs, we perform a grid search over $\theta_{s,k}(\boldsymbol{u}_k) = 0 : \vartheta_1 : \frac{\pi}{2}$ and compute the optimal $\varphi_{s,k}(\boldsymbol{u}_k)$ based on (34) for all K UAVs. The optimal solution is then selected accordingly. Therefore, the time complexity is $\mathcal{O}\left(\frac{\pi}{2\vartheta_1} \times K\right)$. For heterogeneous UAVs, we search over $\varphi_{s,1}(\boldsymbol{u}_1) = 0 : \vartheta_2 : \frac{2\pi}{K}$. We compute the inference components for all K UAVs, sort both the UAV computation time and the communication time, and apply the optimal mapping procedure. Consequently, the time complexity is $\mathcal{O}\left(\frac{\pi^2}{\vartheta_1\vartheta_2} \times \frac{K + \log K}{K}\right)$. As a result, the overall time complexity of solving **P1** with the SVQP scheme is $\mathcal{O}(I(L + \frac{\pi}{2\vartheta_1} \times K))$ for homogeneous UAVs, and $\mathcal{O}(I(L + \frac{\pi^2}{\vartheta_1\vartheta_2}))$ for heterogeneous UAVs. □

Table I: System Parameters

Parameter	Value
Number of UAVs, K	6
Communication bandwidth, B_k	6 MHz [36]
UAV maximum transmit power, $P_{k,\max}$	20 dBm [36]
UAV computation capacity, $V_{k,\max}$	7×10^8 FLOPs/s
Server computation capacity, v_s	1.0×10^{11} FLOPs/s
Carrier frequency, f_c	60 GHz [37]
Noise power spectral density, N_0	-174 dBm/Hz [37]
Path-loss exponent, α	2 [38]
Path-loss for LOS, NLOS, η_1, η_2	$3, 23$ dB [38]
Environment parameters, ψ, ζ	$11.95, 0.14$ [38]

VI. SIMULATION RESULTS

In this section, we present the numerical results to evaluate the performance of our proposed SVQP scheme. In Section VI-A, we present the basic simulation settings. In Section VI-B, we introduce the baseline schemes. In Section VI-C, we demonstrate and compare the performance of the proposed SVQP scheme with the baseline schemes.

A. Simulation Settings

1) Multi-view edge inference system: We consider a multi-view edge inference system with $K = 6$ UAVs and an edge server. Each UAV captures a distinct view of the target and processes it locally using a sub-model. The resulting features are transmitted to the server, where view pooling and final inference are performed. The target is located at the origin, with all UAVs positioned around it. Each UAV maintains the same elevation angle $\{\theta_{s,k}(\boldsymbol{u}_k)\}$, while their azimuth angle $\{\varphi_{s,k}(\boldsymbol{u}_k)\}$ is uniformly distributed over $[0, 2\pi)$. Additional simulation parameters are in Table I.

2) Inference model and synthetic datasets: We adopt an MVCNN architecture with a VGG11 backbone [21]. The ModelNet40 dataset, comprising 3D CAD models across 40 categories, serves as the source of geometric shapes. Existing public datasets typically contain images captured from fixed viewpoints, often with predetermined elevation angles. To rigorously analyze how UAV positioning affects inference performance, we construct a synthetic dataset with controlled viewing parameters. Specifically, for each ModelNet40 target, $K = 6$ UAVs are positioned at uniform $360°/K = 60°$ azimuthal intervals with distinct elevation angles, producing 6 views per target. All views are rendered with realistic shading, resulting in a comprehensive multi-view dataset for recognition tasks. Further details are provided in Appendix of the online supplemental material.

B. Baseline Schemes

- **Baseline 1:** (Det-UAVposition) In this scheme, the UAV position $\{\boldsymbol{u}_k\}$ is given in Section V-D. The split layer l_s, view-pooling layer l_v, quantization level q, transmission power $\{p_k\}$, and computation capacity $\{v_k\}$ are optimized according to Section V-B and Section V-C.
- **Baseline 2:** (Fix-Split/Viewpooling) In this scheme, the split layer and view-pooling layer are fixed, i.e., $l_s = l_s'$ and $l_v = l_v'$ as defined in Section V-B. The variables q

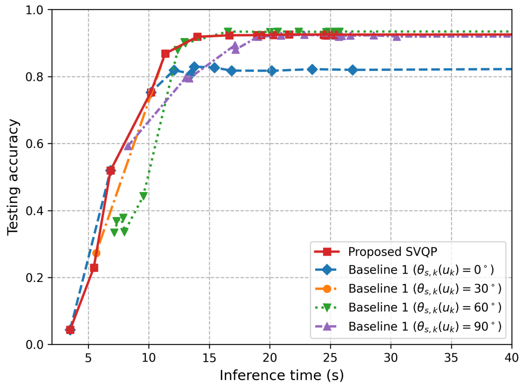

Fig. 8. Testing accuracy versus inference time under the SVQP scheme and Baseline 1.

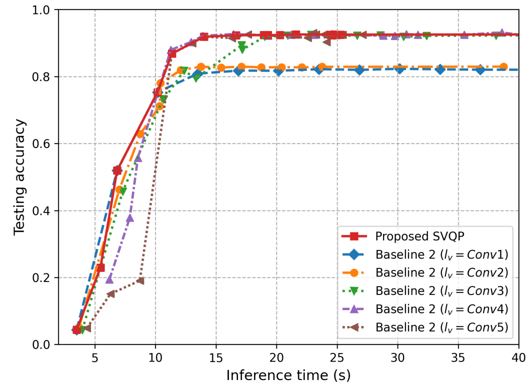

Fig. 9. Testing accuracy versus inference time under SVQP scheme and Baseline 2 with fixed view-pooling layer l_v.

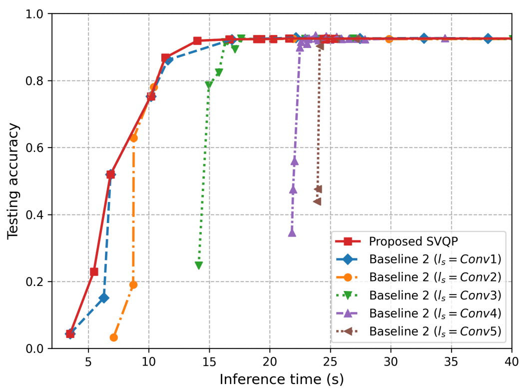

Fig. 10. Testing accuracy versus inference time under the SVQP scheme and Baseline 2 with the fixed split layer l_s.

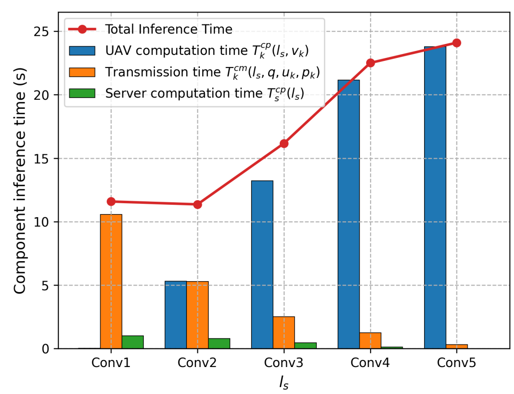

Fig. 11. An illustration of the total inference time and its components when the testing accuracy is set to 0.82.

and $\{\boldsymbol{u}_k\}$ are then optimized according to Section V-C and Section V-D, respectively.

- **Baseline 3:** (Fix-Quantize) This scheme uses the fixed quantization level $q = q_0$ in Section V-C. The variables l_s, l_v, and $\{\boldsymbol{u}_k\}$ are optimized following Section V-B and Section V-D.
- **Baseline 4:** (On-UAV) In this scheme, each UAV conducts the entire inference process locally, including view-pooling. The variables l_v, and q are optimized as detailed in Section V-B and Section V-C. Unlike the previous baseline, transmission time is not included in the inference time. This scenario represents an extreme case for comparative analysis.
- **Baseline 5:** (On-Server) In this scheme, UAVs transmit all raw data to the server, where the entire inference process is conducted centrally. The variables l_v and q are optimized as described in Section V-B and Section V-C. Similar to Baseline 4, transmission time is excluded from the inference time. This also serves as an extreme case for comparison.

C. Performance Evaluation

*1) **Impact of UAV position $\{\boldsymbol{u}_k\}$ on inference performance:*** Fig. 8 compares the inference performance of the proposed SVQP scheme with Baseline 1 (Det-UAVposition) under various UAV positions. All UAVs are assigned fixed azimuthal angles, while the elevation angle $\theta_{s,k}(\boldsymbol{u}_k)$ is varied in Baseline 1 to represent different UAV positions. We plot testing accuracy against total inference time by varying the inference accuracy threshold R_c. The result shows that testing accuracy increases with inference time for both the SVQP scheme and Baseline 1. This is because adopting a deeper split layer l_s and a higher quantization level q enhances inference accuracy (see (22)), but at the cost of increased computation on the UAVs and longer feature transmission times, leading to greater total inference time. This demonstrates a tradeoff between inference accuracy and inference time. Moreover, the SVQP scheme achieves near-optimal testing accuracy among all baseline methods, owing to its ability to optimize UAV positions[8]. Additionally, the results indicate that multi-view inference consistently outperforms single-view inference. Specifically, when $\theta_{s,k}(\boldsymbol{u}_k) = 0^\circ$, corresponding to the single-view scenario where all UAVs capture identical data, the testing accuracy does not exceed 0.82. This demonstrates that multi-view data yields better model performance than with single-view data, consistent with the findings in [21].

*2) **Impact of view-pooling layer l_v on inference performance:*** Fig. 9 shows testing accuracy versus inference time for the SVQP scheme and Baseline 2 (Fix-Split/Viewpooling)

[8]Although the simulations are conducted under a homogeneous UAV setting, the proposed SVQP scheme is also applicable to heterogeneous UAV scenarios and can achieve superior inference performance in both cases compared with the considered baseline schemes.

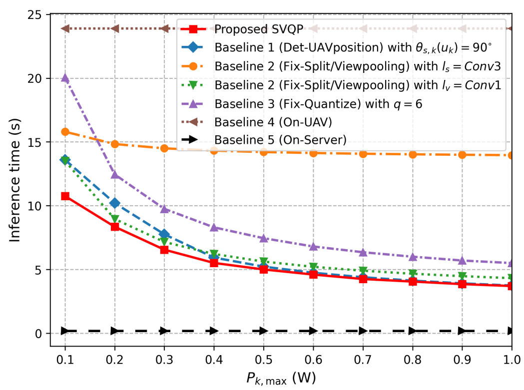

Fig. 12. Inference time versus maximum transmission power $P_{k,\max}$ for different baseline schemes with testing accuracy 0.82.

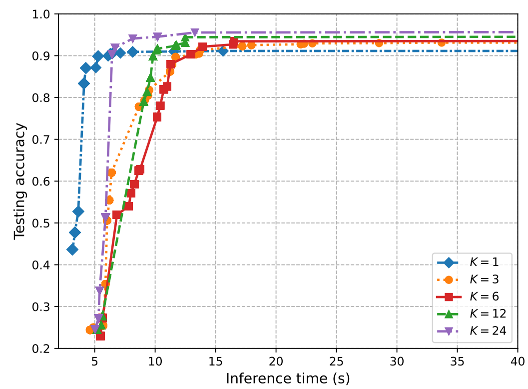

Fig. 13. Testing accuracy versus inference time under the SVQP scheme with different number of UAVs K.

under different view-pooling layer l_v. The SVQP scheme achieves the highest testing accuracy among all baselines. This improvement is attributed to the closed-form relationship derived between $R_{\text{view}}(l_s, l_v, q, \{\boldsymbol{u}_k\})$ and l_v in (22), enabling the optimization of l_v. Furthermore, when view-pooling is performed after Conv1 or Conv2, the testing accuracy does not exceed 0.82. This is because view-pooling at shallow layers alters the model structure and may fail to capture sufficiently representative features. In contrast, performing view-pooling at deeper layers provides more abstract and discriminative features, leading to higher accuracy.

*3) **Impact of split layer l_s on inference performance**:* Fig. 10 shows testing accuracy versus inference latency for the SVQP scheme and Baseline 2 (Fix-Split/Viewpooling) under different split layer l_s. The SVQP scheme achieves the highest testing accuracy among all baselines, as it can select the optimal l_s to balance UAV computation time, feature transmission time, and inference accuracy. To further analyze how the total inference time varies with l_s, Fig. 11 presents the total inference time and its three components when the testing accuracy is 0.82. As l_s increases, more layers are assigned to the UAV, resulting in longer UAV computation time $T_k^{cp}(l_s, v_k)$. In contrast, the transmission time $T_k^{cm}(l_s, q, \boldsymbol{u}_k, p_k)$ decreases, since the feature size $N(l_s)$ becomes smaller with a higher l_s. In addition, the server computation time $T_s^{cp}(l_s)$ decreases because fewer layers remain to be processed at the server[9]. Overall, the total inference time first slightly decreases and then increases as l_s increases, indicating a tradeoff among these three components. Thus, the optimal l_s should be selected to ensure efficient inference.

*4) **The comparison of inference time**:* In Fig. 12, we compare the inference time required to achieve testing accuracy 0.82, across different baseline schemes. The proposed SVQP scheme achieves the shortest inference time among all baselines. In comparison, Baseline 1 (Det-UAVposition) employs deterministic UAV positioning with $\theta_{s,k}(\boldsymbol{u}_k) = 90°$, Baseline 2 (Fix-Split/Viewpooling) uses a fixed split or view-pooling layer with l_s =Conv3 or l_v =Conv1, and Baseline 3 (Fix-Quantize) applies fixed quantization with $q = 6$. Additionally, Baseline 4 (On-UAV) and Baseline 5 (On-Server) represent the worst and best cases, respectively. Since the computation capacity of the server v_s is significantly greater than that of the UAV v_k, performing inference entirely on the server achieves the minimum inference time, while performing inference entirely on the UAV results in the maximum inference time.

*5) **Impact of the number of UAVs K on inference performance**:* As shown in Fig. 13, we plot the testing accuracy versus inference time under different numbers of UAVs K. The results show that the achievable final testing accuracy generally increases as K becomes larger. This is because involving more UAVs provides more viewing perspectives for training the multi-view target recognition model. It enhances the robustness of inference and enables the target to be recognized from diverse viewpoints. However, under stringent inference time constraints, e.g., when the inference time is less than 15 s, a smaller number of UAVs such as $K = 1$ can achieve comparable testing accuracy with shorter inference time than larger numbers of UAVs, such as $K = 3, 6, 12, 24$. This is because all UAVs share the same bandwidth budget. A smaller K allows each UAV to obtain a larger bandwidth allocation, thereby reducing the transmission time during inference.

VII. Conclusion

In this paper, we studied UAV deployment and model architecture design in a multi-view edge inference system to enhance inference performance. Specifically, we analyzed the impact of UAV deployment and view-pooling layer placement on inference accuracy, and derived a closed-form boundary that explicitly connects them to the classification margin. This analysis reveals that jointly optimizing of these factors enables the extraction of more discriminative features and supports a more effective model architecture, thereby improving inference accuracy. We further quantified the quantization error introduced by feature quantization during wireless transmission. Building on this analysis, we formulated an inference latency minimization problem subject to an inference accuracy constraint. Then, we employed an alternating optimization approach that decomposes it into three subproblems. Moreover, we proposed the SVQP scheme to efficiently computes suboptimal solutions. Simulation results demonstrated that the

[9] We assume that all UAVs possess the same maximum transmission power $P_{k,\max}$ and computation capacity $V_{k,\max}$. Thus, the computation time $T_k^{cp}(l_s, v_k)$ is identical for all UAVs. In Fig. 11, the transmission time corresponds to $\max_{k\in\mathcal{K}} T_k^{cm}(l_s, q, \boldsymbol{u}_k, p_k)$.

proposed SVQP scheme achieves superior performance in terms of the inference accuracy-latency tradeoff compared with existing baseline schemes.

References

[1] M. Song, Y. Lin, J. Wang, G. Sun, C. Dong, N. Ma, D. Niyato, and P. Zhang, "Trustworthy intelligent networks for low-altitude economy," *IEEE Commun. Mag.*, vol. 63, no. 7, pp. 72–79, Jul. 2025.

[2] Y. Jiang, X. Li, G. Zhu, H. Li, J. Deng, K. Han, C. Shen, Q. Shi, and R. Zhang, "Integrated sensing and communication for low altitude economy: Opportunities and challenges," *IEEE Commun. Mag.*, pp. 1–7, Apr. 2025.

[3] D. He, W. Yuan, J. Wu, and R. Liu, "Ubiquitous UAV communication enabled low-altitude economy: Applications, techniques, and 3GPP's efforts," *IEEE Network*, pp. 1–1, May 2025.

[4] ITU-R, "Framework and overall objectives of the future development of IMT for 2030 and beyond," Nov. 2023. [Online]. Available: https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2030/Pages/default.aspx

[5] P. Zhang, W. Xu, Y. Liu, X. Qin, K. Niu, S. Cui, G. Shi, Z. Qin, X. Xu, F. Wang, Y. Meng, C. Dong, J. Dai, Q. Yang, Y. Sun, D. Gao, H. Gao, S. Han, and X. Song, "Intellicise wireless networks from semantic communications: A survey, research issues, and challenges," *IEEE Commun. Surv. Tutor.*, vol. 27, no. 3, pp. 2051–2084, Jun. 2025.

[6] M. Lee, G. Yu, and H. Dai, "Decentralized inference with graph neural networks in wireless communication systems," *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 2582–2598, May 2023.

[7] J. Shao and J. Zhang, "Communication-computation trade-off in resource-constrained edge inference," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 20–26, Dec. 2020.

[8] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[9] J. Shao, Y. Mao, and J. Zhang, "Task-oriented communication for multi-device cooperative edge inference," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 73–87, Jan. 2023.

[10] A. Das, S. K. Ghosh, A. Raha, and V. Raghunathan, "Toward energy-efficient collaborative inference using multisystem approximations," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 17 989–18 004, May 2024.

[11] M. Singhal, V. Raghunathan, and A. Raghunathan, "Communication-efficient view-pooling for distributed multi-view neural networks," in *Proc. Des. Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 1390–1395.

[12] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[13] D. Wen, X. Jiao, P. Liu, G. Zhu, Y. Shi, and K. Huang, "Task-oriented over-the-air computation for multi-device edge AI," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 2039–2053, Mar. 2024.

[14] Y. Fu, P. Qin, Y. Wang, L. Chen, M. Li, and X. Zhao, "Over-the-air edge inference for low-altitude airspace: Generative AI-aided multi-task batching and beamforming design," *IEEE Trans. Commun.*, pp. 1–1, Apr. 2025.

[15] D. Wen, P. Liu, G. Zhu, Y. Shi, J. Xu, Y. C. Eldar, and S. Cui, "Task-oriented sensing, computation, and communication integration for multi-device edge AI," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 2486–2502, Mar. 2024.

[16] J. Yao, W. Xu, G. Zhu, K. Huang, and S. Cui, "Energy-efficient edge inference in integrated sensing, communication, and computation networks," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 10, pp. 3580–3595, Oct. 2025.

[17] Z. Liu, Q. Lan, A. E. Kalør, P. Popovski, and K. Huang, "Over-the-air multi-view pooling for distributed sensing," *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 7652–7667, Jul. 2024.

[18] X. Chen, K. B. Letaief, and K. Huang, "On the view-and-channel aggregation gain in integrated sensing and edge AI," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 9, pp. 2292–2305, Sept. 2024.

[19] Q. Lan, Q. Zeng, P. Popovski, D. Gündüz, and K. Huang, "Progressive feature transmission for split inference at the wireless edge," 2021. [Online]. Available: https://arxiv.org/abs/2112.07244

[20] J. Shao and J. Zhang, "BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems," 2020. [Online]. Available: https://arxiv.org/abs/1910.14315

[21] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015.

[22] R. Chen, S. Han, J. Xu, and H. Su, "Visibility-aware point-based multi-view stereo network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3695–3708, Oct. 2021.

[23] W. Xu, T. Zhang, X. Mu, Y. Liu, and Y. Wang, "Trajectory planning and resource allocation for multi-UAV cooperative computation," *IEEE Trans. Wirel. Commun.*, vol. 72, no. 7, pp. 4305–4318, Jul. 2024.

[24] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 323–341, Jan. 2022.

[25] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Federated learning with quantized global model updates," 2020. [Online]. Available: https://arxiv.org/abs/2006.10672

[26] W. Khawaja, I. Guvenc, and D. W. Matolak, "A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2361–2391, thirdquarter 2019.

[27] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[28] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wirel. Commun*, vol. 15, no. 6, pp. 3949–3963, Jun. 2016.

[29] F. Dong, H. Wang, D. Shen, Z. Huang, Q. He, J. Zhang, L. Wen, and T. Zhang, "Multi-exit DNN inference acceleration based on multi-dimensional optimization for edge intelligence," *IEEE Trans. Mob. Comput.*, vol. 22, no. 9, pp. 5389–5405, Sept. 2023.

[30] X. Yuan, N. Li, T. Zhang, M. Li, Y. Chen, J. F. M. Ortega, and S. Guo, "High efficiency inference accelerating algorithm for NOMA-based edge intelligence," *IEEE Trans. Wireless Commun.*, vol. 23, no. 11, pp. 17 539–17 556, Nov. 2024.

[31] J. Sokolić, R. Giryes, G. Sapiro, and M. R. D. Rodrigues, "Robust large margin deep neural networks," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4265–4280, Aug. 2017.

[32] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[33] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[34] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research: the 14th international symposium ISRR*. Springer, 2011, pp. 3–19.

[35] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[36] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6721–6726, Jul. 2018.

[37] P. Liu, G. Zhu, S. Wang, W. Jiang, W. Luo, H. V. Poor, and S. Cui, "Toward ambient intelligence: Federated edge learning with task-oriented sensing, computation, and communication integration," *IEEE J. Sel. Top. Signal Process.*, vol. 17, no. 1, pp. 158–172, Jan. 2023.

[38] 3GPP Technical Report 36.777, "Study on enhanced LTE support for aerial vehicles (release 15)," Dec. 2017.

Yao Tang (Member, IEEE) received the B.Eng. degree in Communication Engineering from the University of Electronic Science and Technology of China in 2018, and the Ph.D. degree in Information Engineering from The Chinese University of Hong Kong in 2023. She worked at the Huawei Hong Kong Research Center in 2024 and was a Postdoctoral Research Fellow with the Department of Information Engineering, The Chinese University of Hong Kong, in 2025. She is currently a Research Assistant Professor with the AI for Science and Engineering Center, Shenzhen Loop Area Institution. Her research interests include optimization in UAV-enabled wireless communications, federated edge learning, and semantic communications.

Guangxu Zhu (Member, IEEE) received the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong in 2019. Currently he is a senior research scientist and deputy director of network and machine intelligence center at the Shenzhen research institute of big data, and an adjunct associate professor with the Chinese University of Hong Kong, Shenzhen. His recent research interests include edge intelligence, large foundation model, and integrated sensing and communication. He is a recipient of the 2023 IEEE ComSoc Asia-Pacific Best Young Researcher Award and Outstanding Paper Award, the World's Top 2% Scientists by Stanford University, the "AI 2000 Most Influential Scholar Award Honorable Mention", the Young Scientist Award from UCOM 2023. He serves as associate editors at top-tier journals in IEEE, including IEEE TMC, TWC and WCL. He is the vice co-chair of the IEEE ComSoc Asia-Pacific Board Young Professionals Committee.

Chao Shen (Member, IEEE) received the B.S. degree in communication engineering and the Ph.D. degree in signal and information processing from Beijing Jiaotong University (BJTU), Beijing, China, in 2003 and 2012, respectively. From 2014 to 2015, he was a Visiting Scholar with the University of Maryland, College Park, MD, USA, and The Chinese University of Hong Kong, Shenzhen, China, from 2017 to 2018. Since 2012, he has been an Associate Professor with the State Key Laboratory of Rail Traffic Control and Safety, BJTU. In 2022, he joined Shenzhen Research Institute of Big Data, as a Senior Research Scientist. His research interests focus on large-scale network optimization, ultra-reliable and low-latency communication (URLLC), and integrated sensing and communication (ISAC) for B5G/6G networks.

Tse-Tin Chan (Member, IEEE) received the B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, China, in 2014 and 2020, respectively. From 2020 to 2022, he was an Assistant Professor with the Department of Computer Science, The Hang Seng University of Hong Kong (HSUHK), Hong Kong. He is currently an Assistant Professor with the Department of Mathematics and Information Technology, The Education University of Hong Kong (EdUHK), Hong Kong. His research interests include wireless communications and networking, age of information (AoI), AI-native wireless systems, and integrated sensing and communication (ISAC). He has served as a Guest Editor for the IEEE Journal of Selected Topics in Signal Processing and the IEEE Open Journal of the Communications Society. He has also served on the organizing committees and technical program committees of various flagship conferences and workshops, including IEEE ICC and IEEE GLOBECOM.

Tat-Ming Lok (SM'03) received the B.Sc. degree from the Chinese University of Hong Kong, and the M.S.E.E. and the Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA. He is currently an Associate Professor in the Department of Information Engineering at the Chinese University of Hong Kong. His research interests include communication theory, communication networks, signal processing for communications, and wireless systems. He served on the technical program committees of many international conferences, including the IEEE International Conference on Communications, the IEEE Vehicular Technology Conference, the IEEE GLOBECOM, the IEEE Wireless Communications and Networking Conference, and the IEEE International Symposium on Information Theory. He also served on the editorial boards of several international journals, including the IEEE Transactions on Vehicular Technology and the IEEE transactions on Wireless Communications.

Multi-View Edge Inference in Low-Altitude Airspace: A Joint UAV Deployment, Model Architecture Design Framework

Yao Tang, Guangxu Zhu, Chao Shen, Tse-Tin Chan, and Tat-Ming Lok

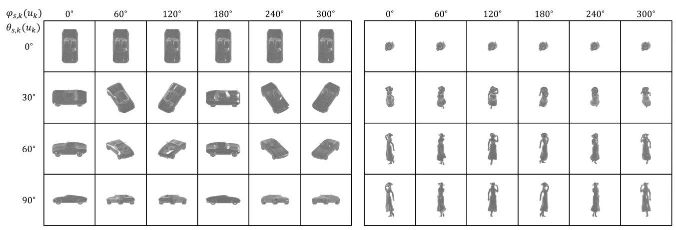

Fig. 1. Generated multi-view images under varying sensing elevation angle $\{\theta_{s,k}(\boldsymbol{u}_k)\}$ and azimuth angle $\{\varphi_{s,k}(\boldsymbol{u}_k)\}$.

Appendix A
Classification Margin Approximation

The classification margin $\Delta(l_v, \{\boldsymbol{u}_k\})$ comprises two components: the disturbance and the supremum of the Jacobian norm. To examine how these terms vary with l_v and $\{\boldsymbol{u}_k\}$, we conduct experiments using both average-pooling and max-pooling strategies. The experiments includes data generation, model training, and curve fitting, as detailed below.

- *Data generation:* We use the ModelNet40 dataset as the source of shapes. Multi-view images are generated in Python using the BlenderPhong renderer, as illustrated in Fig. 1. This process produces a total of $56,694$ images at a resolution of 112×112 pixels, with 45,534 images for training and $11,160$ for validation (see Fig. 1).
- *Model training:* Using datasets generated with varying elevation angle $\{\theta_{s,k}(\boldsymbol{u}_k)\}$, we train an MVCNN model with a VGG11 backbone, placing the view-pooling at different layers. Model checkpoints are saved for each training configuration.
- *The disturbance:* After loading the saved checkpoints, we fix the output features at the selected l_v. Using stochastic gradient descent, we compute the representative input $(x_j(\{\boldsymbol{u}_k\}), y_j)$ and evaluate the disturbance $\|\boldsymbol{x}_{kj}(\boldsymbol{u}_k) - \boldsymbol{x}_j(\{\boldsymbol{u}_k\})\|_2$ for different $\{\boldsymbol{u}_k\}$.
- *The supremum of the Jacobian norm:* Using the saved checkpoints and training data, we compute the maximum JM with respect to l_v and $\{\boldsymbol{u}_k\}$ via backpropagation.