# OAVS: Efficient Online Learning of Streaming Policies for Drone-sourced Live Video Analytics

Zekai Li*, Miao Zhang†, and Yifei Zhu*‡
*UM-SJTU Joint Institute, Shanghai Jiao Tong University
†School of Computing Science, Simon Fraser University
‡Cooperative Medianet Innovation Center (CMIC), Shanghai Jiao Tong University
Email: lzk666@sjtu.edu.cn, mza94@sfu.ca, yifei.zhu@sjtu.edu.cn

***Abstract*—Drone-sourced live video analytics has extensive applications across diverse domains. Adaptive video streaming is a pivotal technique in these applications that targets at effectively delivering video content to servers under varying network conditions, enabling complex analytics afterward. However, our thorough data analysis reveals that conventional offline video streaming policies cannot effectively adapt to highly fluctuating drone network environments and dynamic changes in aerial view scenes. This results in suboptimal analytic performance and necessitates online adaptation for policy models. Yet, obtaining ground-truth analytics results directly from drones is infeasible due to their limited capacity. Furthermore, naively streaming original videos to the server for online adaption is greatly challenged by the scarce and dynamic networks, leading to decreased accuracy performance and escalated transmission cost if not properly designed. In this paper, we present `OAVS`, a novel online learning-enabled adaptive streaming framework for drone-sourced video analytics. To facilitate cost-effective online retraining, we design a hierarchical reinforcement learning approach in which the upper-level module intelligently determines the timing for online retraining, balancing machine-perceived quality of experience (QoE) improvement and transmission cost. Meanwhile, the lower-level module dynamically allocates bitrate to maximize machine-perceived QoE. Extensive experiments based on real-world drone video and aerial network datasets demonstrate that our proposed framework achieves a 17.7% mean accuracy increase, a 37.5% decrease in the mean failure rate of video uploading, and a 5.2% mean latency decrease compared to state-of-the-art solutions.**

## I. Introduction

Drones have witnessed widespread adoption in diverse domains, such as disaster recovery, logistics and transportation, precision agriculture, and forestry [1], [2]. The global unmanned aerial vehicle market size is projected to reach 25 billion USD by 2027 [3], fostering the development of the low-altitude airspace economy. Drone-sourced live video analytics that involves real-time processing and interpretation of video data captured by drones is the key to unleash the full potential of drones. Powered by advanced 5G communication techniques and beyond, these cellular network-assisted drones benefit from extra ultra-low latency and high reliability, enabling operation in GPS-disabled environments and beyond visual line-of-sight scenarios. For example, in scenarios such as large-scale events or sensitive locations, it enables a remote mobile surveillance system that can proactively identify and address potential security threats.

This work is supported by the National Natural Science Foundation of China (Grant No. 62302292) and the Fundamental Research Funds for the Central Universities. Corresponding author: Yifei Zhu.

Constrained by the limited onboard computing capability, drone-sourced videos are usually streamed back to ground servers over volatile aerial networks so that complex video analytics tasks can be accomplished. To ensure real-time video transmission, adaptive video streaming is a widely adopted technique that dynamically adjusts the video bitrate based on varying conditions, such as real-time bandwidth and buffer size [4], [5]. Recent years have seen the rise of Deep Reinforcement Learning (DRL)-based adaptive streaming solutions [6], [7], which leverage sophisticated algorithms to optimize streaming quality by learning from vast datasets of network conditions and system interactions. However, the majority of these solutions rely heavily on models trained offline using pre-collected datasets, which may not fully represent the complexities and variabilities of real-world networks and system interactions. This reliance on offline training leads to suboptimal performance, particularly for high-mobility devices like drones, since offline-trained models are often challenged by data patterns unseen during the training phase.

Our thorough analysis of the drone-sourced videos and network traces confirms the limitations of existing adaptive streaming solutions and reveals the following challenges: First, *drone-to-terrestrial cellular networks are highly fluctuating.* Current cellular networks are essentially ground-centric. Highly flexible drones operating at diverse speeds and altitudes experience more dynamic network conditions and unseen network patterns such as frequent handovers [8]. Most of the existing network predictors suffer from performance drift over time during the flight. Second, *aerial-view scenes captured by drones are exceptionally dynamic.* The high mobility of drones and the relative motion of both drones and the observed objects further make the content highly dynamic. We observe that the mean object size ratio can change $7.6\times$ on average across consecutive chunks. An offline-trained video streaming system, including inaccurate network throughput predictions, suffers a severe loss of video completeness and real-time performance, hurting the analytics results in underexplored and unseen data patterns.

Fundamentally, addressing the challenges posed by drone environments demands an innovative approach capable of adapting to constant changes and unseen data patterns. While offline-trained streaming strategies struggle to meet this requirement, online learning [9] emerges as a promising solution due to its ability to continuously update models with new data, enhancing adaptability to changing environments. However, ground-truth analytics result is required for applying online learning to update drone video streaming policies. Constrained by real-time requirements and limited resources on drones, such analytics cannot be conducted on drones at all. Consequently, original drone-captured videos have to be transmitted back to servers for analytics and retraining. This leads to the third challenge, *online adaption of drone-end streaming policy is extremely costly.* Naively periodically or uniformly transmitting original videos leads to either insignificant or delayed online adaption. The transmission of original videos also significantly reduces streaming performance in limited bandwidth which leads to unbearably performance in latency and video completeness. Therefore, achieving a balance among communication resource usage, analytic performance, and real-time model updates is crucial.

In this paper, we present `OAVS`, a novel online learning-enabled adaptive video streaming framework for drone-sourced video analytics. `OAVS` enables cost-effective online learning of bitrate allocation policy and network prediction model to handle drones' dynamic environments. Specifically, we first build a machine-perceived Quality of Experience (QoE) model to capture the intrinsic requirements in drone-sourced video analytics. We then design a novel lightweight AdaLSTM model on drones for accurate bandwidth prediction to handle the model drifting situation for network prediction. A novel hierarchical reinforcement learning algorithm is further designed for our core streaming module. The upper-level reinforcement learning module in this process balances the improvement in machine-perceived QoE and the potential transmission cost, and intelligently decides when to transmit original video data and other supporting data to the server for model retraining. The lower-level module then refines the bitrate allocation based on the specific video content and network conditions to maximize the machine-perceived QoE. To the best of our knowledge, this is *the first work that enables online adaption of adaptive video streaming policy using hierarchical deep reinforcement learning*, with greatly improved machine-centric QoE for drone-sourced video analytics and significantly reduced model update overheads. In summary, the main contributions of this paper are as follows:

- We conduct a comprehensive measurement that uncovers the unique characteristics of drone-sourced video analytics. Our findings highlight the necessity of continuously updating the streaming strategies and the significant challenges in doing so in drone-sourced video analytics.
- We design a novel online learning-enabled adaptive video streaming framework for drone-sourced video analytics utilizing hierarchical reinforcement learning. The upper-level policy network intelligently determines when to perform online retraining, while the lower-level policy network makes specific bitrate selection decisions that maximize machine-perceived QoE.
- Extensive experiments on real-world drone network traces and drone video datasets demonstrate that our approach outperforms the state-of-the-art data-driven approaches in bandwidth prediction accuracy by 70.0%, improves mean analytic accuracy by 17.7% with 5.2% reduction in latency and 37.5% reduction in the mean uploading failure rate of video transmission.

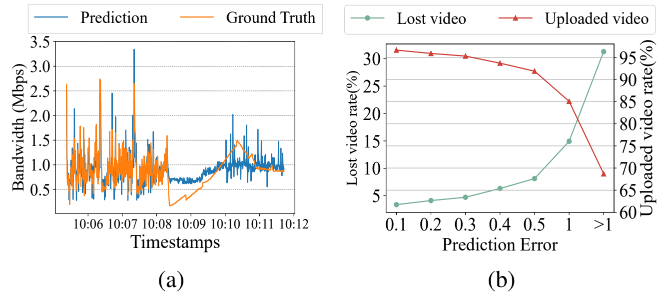



Fig. 1: (a) Prediction results generated by an offline LSTM model [12]. (b) Inaccurate network prediction has a severe influence on video delivery.

The remainder of this paper is organized as follows. We first introduce our measurement studies in Section II to identify the limitations and challenges of drone-sourced video analytics. Then we present our problem formulation in Section III. We illustrate `OAVS`'s design in Section IV and evaluate its performance in Section V. Related work is discussed in Section VI, followed by the conclusion in Section VII.

## II. MOTIVATION

Drones are constrained by onboard computational resources, necessitating the offloading of video analytics tasks to more powerful servers. A prevalent solution is to employ adaptive bitrate allocation for transmitting video from drones to servers. Adaptive streaming typically consists of bandwidth prediction and adaptive bitrate selection. Captured videos are split into several fixed-length downloadable video chunks and each chunk will be encoded and transmitted with a specific bitrate according to the bitrate selection model. This section explores two key questions: (1) why offline-trained policies fail (II-A), and (2) what challenges emerge when updating streaming policies online (II-B)? We choose the commonly used VisDrone2019 [10] and SeaDroneSee datasets [11] as our drone video datasets and drone-based cellular network dataset [8] to reveal the characteristics of drone-sourced video analytics in dynamic cellular networks. Detailed introductions to the dataset can be found in the evaluation section. The datasets cover a wide range of environments, objects, and densities in drone scenarios.

### *A. Why do offline-trained policies fail?*

**Cellular-connected drones experience highly fluctuating network conditions.** Drones, operating at diverse locations and altitudes, experience more dynamic network conditions

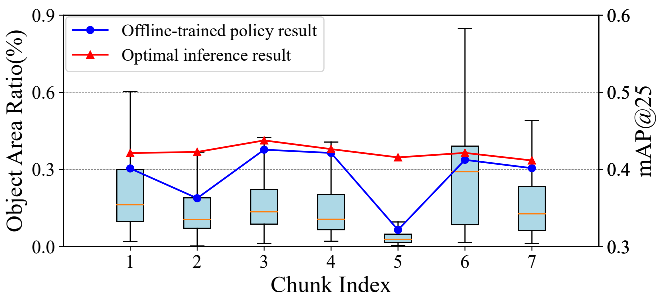



Fig. 2: Varying object sizes in box plot and the corresponding accuracy performance over a series of video chunks using an offline-trained streaming policy. Optimal accuracy is also plotted according to inference results on original video chunks.

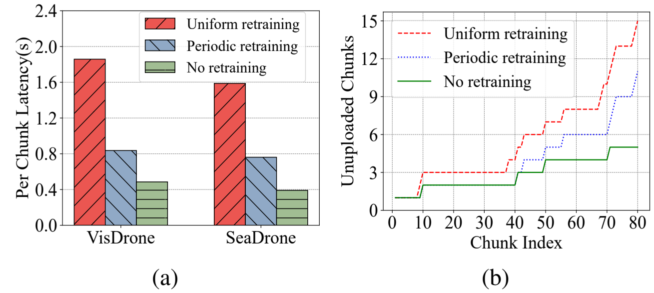



Fig. 3: (a) Per chunk transmission latency using three streaming modes. (b) Count of unuploaded chunks over continuous video chunks using three streaming modes.

due to frequent handovers. Previous studies [8], [13] also identify the significant changing patterns of aerial networks, which challenge bandwidth prediction on drones. We train a standard LSTM using the training set selected from drone network traces data [8], then we test the performance of this pre-trained model on unseen network traces and present the results in Fig. 1a. It is obvious that when encountered with a new network pattern after 10:08, the prediction performance drifts drastically. Correspondingly, the mean absolute error (MAE) of the prediction result raises from 26.5% to over 100.0% after encountering these new network patterns.

Inaccurate predictions and the resulting inappropriate bitrate allocations can have a severe influence on streaming performance. Underestimated bandwidth results may lead to low bitrate allocation and degrade analytics accuracy. Overestimated bandwidth may lead to higher bitrate allocation and subsequent chunks may be lost due to a highly occupied buffer. We group video chunks according to their prediction error range and plot the value of lost video ratio and received video ratio against six prediction error intervals in Fig. 1b. The ratio represents the ratio of unsuccessfully or successfully uploaded video chunks in a typical aerial streaming system. As can be seen, higher prediction errors lead to more lost video chunks and fewer uploaded video chunks. Specifically, more than 10% video chunks are lost when the error is over 50%.

**Offline-trained video streaming policies are not robust in drone's dynamic environments.** Fig. 2 demonstrates the varying video content characteristics and their impacts on the video analytics accuracy of an offline-trained policy. We randomly select seven continuous video chunks from VisDrone [10] and plot the box plot of their object size ratio. Then we apply a state-of-the-art reinforcement learning-based offline-pretrained bitrate selection policy and compute their video analytics accuracy (mean average precision [14]). We can observe that the mean object size ratio could change 7.6× on average across consecutive chunks. The resulting accuracy fluctuates widely when encountered with dynamic video content, e.g., accuracy drops over 25% from 0.43 to 0.31 from chunk 4 to chunk 6. Offline-trained policy thus cannot maintain consistent performance in the dynamic drone environment.

*The observations identify the impact of dynamic aerial network conditions and aerial scenes on offline-trained network predictions and streaming policies. Freshness and completeness of the content, and eventually the analytics results are all greatly jeopardized.*

### *B. Challenges of updating streaming policy online*

**Transmitting original videos is a must but it severely damages streaming performance.** In order to evaluate the performance of the current streaming policy and for further training, it is necessary to have the originally captured videos as ground truth for obtaining accurate analytics results. Therefore, the original videos captured on drones need to be transmitted to servers for further analytics and evaluation, which significantly increases communication pressure. This process is further slowed by the limited computing resources on drones, which also detracts from their primary computational tasks.

Fig. 3 demonstrates the performances of latency and unuploaded chunks when naively transmitting original videos for online retraining. Uniform retraining strategy requires continuous transmission of original videos all the time; no retraining strategy does not transmit original videos and adaptively allocates bitrates for each chunk based on network conditions as previously mentioned; periodic retraining only transmits original videos every ten chunks. As shown in Fig. 3a, we randomly select 80 continuous video chunks and evaluate the per-chunk transmission latency. Uniform retraining and periodic retraining cause 3.47× and 1.74× higher latency compared to offline streaming strategy, respectively. We also count unuploaded chunks when streaming 80 continuous video chunks selected from VisDrone [10]. Uniform retraining and periodic retraining cause 3× and 2.2× higher uploading failure compared to non-retraining streaming strategy as illustrated in Fig. 3b. Naively transmitting original videos for online retraining is challenging and impractical in this scenario.

*The observations reveal the necessity for a cost-efficient online training framework capable of delivering robust performance across all drone environments while minimizing retraining costs.*

## III. System Model and Problem Formulation

### A. Drone-sourced Live Video Analytics

We study a drone-sourced video analytics system that uses tile-based video streaming to transmit captured videos from drones to the server for backend live video analytics. To be specific, once $a$ frames are accumulated, they are compressed into a video chunk and subsequently transmitted to the server. Each video chunk is divided into $m \times n$ tiles, where $m$ denotes the number of rows and $n$ represents the number of columns. We define $i = \{1, 2, ..., m\}$ as the row index candidates and $j = \{1, 2, ..., n\}$ as the column index candidates. We name the tiles that include detected objects as Tiles of Interest (ToI).

Upon capturing a frame, a ToI selection model on the drone detects significant regions based on a lightweight detection model. Using the location on the ToI, the adaptive bitrate selection model determines the bitrate for tiles inside and outside the ToI, in accordance with the video analytics' QoE preferences. For the $c$-th chunk, $b^c_{i,j} \in \mathcal{R}$ is the bitrate for the tile at location $(i, j)$, where $\mathcal{R}$ represents the candidate bitrate set. The processed chunk enters a fixed-size video buffer waiting for transmission. If the sending buffer is full, the upcoming chunk will be dropped. The drone continuously transmits video chunks in the buffer to the server with the selected bitrate for further video analytics.

### B. Machine-Perceived QoE Model

Adapting to the scarce and variable network bandwidth between camera clients and servers remains a significant challenge in video analytics streaming. Traditional adaptive video streaming protocols are not well-suited for live video analytics since they are designed to optimize human-perceived QoE, which focuses on providing high-quality videos without interruptions for human viewers. However, live video analytics aims at maximizing machine-perceived QoE by optimizing server-side DNN inference accuracy without causing analysis lags. Unlike human viewers who may be sensitive to dropped frames or reduced video quality, video analytics algorithms can tolerate such issues as long as the analytics results are not affected.

Motivated by this distinction, we develop a machine-perceived QoE model in this study by incorporating three critical factors in drone-sourced video analytics: backend vision task accuracy, transmission latency, and video chunk loss rate. They reflect the analytics accuracy, timeliness, and completeness of the analytic results. For the purpose of evaluating our approach, we take the object detection task as a case study, wherein the accuracy is represented by the mean average precision (mAP) [14]. Specifically, these three factors are defined as follows for a video.

**Average accuracy** is defined as the average accuracy of the video analytics task result for a video $v$. Suppose $v$ is divided into $C$ chunks, $b_c$ is the bitrate selection result for each video chunk $c$. $A_k(\cdot)$ is the definition of accuracy for task $k$:

$$QoE_{1,v} = \sum_{c=0}^{C} \frac{A_k(b_c)}{C}. \tag{1}$$

**Average Uploading latency.** Assume $B_c$ is the average bandwidth when transmitting the $c$-th chunk and $b^c_{i,j}$ is the bitrate for each tile. The transmission latency of the chunk can be calculated as

$$QoE_{2,v} = \sum_{c=0}^{C} \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} b^c_{i,j}}{B_c}. \tag{2}$$

Other sources of latency such as encoding latency, inference latency, RTT latency, and ToI detection latency are usually fixed values or independent from adaptive streaming, with the chunk latency being the major latency factor that directly reflects the performance of adaptive bitrate selection decisions, so we only consider transmission latency here. The complete end-to-end latency, including other latency sources, will be presented in the evaluation part.

**Average Loss rate.** Assume $V_c(\cdot)$ is a boolean function as an indication of whether a video chunk is uploaded successfully or not. Inaccurate bitrate selection results always lead to loss occurrence which fails upcoming chunks to stack in the occupied video buffer. In our system, a new upcoming video chunk is discarded when this chunk is generated but the video buffer is full at the edge side. This loss is then detected and counted into the loss rate. The loss rate can be calculated by the ratio of the uploaded video chunks to all test video chunks, namely

$$QoE_{3,v} = \frac{\sum_{c=0}^{C} V_c(b_c)}{C}. \tag{3}$$

In summary, our machine-perceived QoE at the $c$-th chunk can be modeled as:

$$QoE_v = \alpha_1 QoE_{1,v} - \alpha_2 QoE_{2,v} - \alpha_3 QoE_{3,v}, \tag{4}$$

where $\alpha_1, \alpha_2, \alpha_3$ are non-negative weights for each component of the QoE metrics.

### C. Problem Formulation for Drone-sourced Video Analytics

For a given QoE weight vector $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, we target maximizing the overall machine-perceived QoE over all video chunks by deciding the bitrate for each tile. Formally, our bitrate allocation problem for drone-sourced video analytics is formulated as follows:

$$\max_{r^c_{ij}} \sum_{c=1}^{C} QoE_c \tag{5}$$

$$B_c = max\{(B_{c-1} - D_{c-1}), 0\} + T_c \tag{6}$$

$$b_{i,j} \in \mathcal{R} \tag{7}$$

where (6) denotes the buffer occupancy and $T_c$ is the duration of the $c$-th chunk; (7) indicates that the bitrate of each tile is chosen from the candidate bitrate set R.

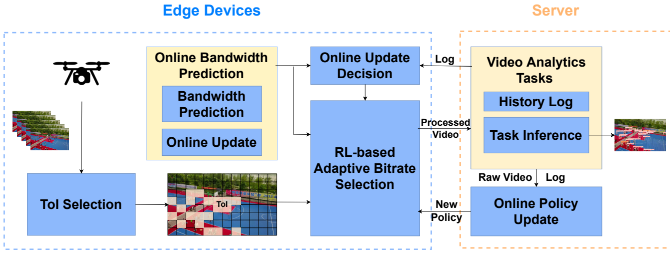



Fig. 4: The online-learning-based adaptive video streaming framework for drone-sourced video analytics.

## IV. System Design of OAVS

### A. Design Overview

Our drone-based real-time video streaming framework for video analytics is depicted in Fig. 4. For each video chunk captured by the drone camera, our ToI selection module detects the ToI tiles within the video and returns their coordinates. Meanwhile, we have carefully designed an online-LSTM-based bandwidth prediction module that predicts future bandwidth based on recently learned data patterns through an online approach, suited for the extremely dynamic network conditions of drones. The ToI locations and predicted bandwidth are then fed into our hierarchical online adaptive bitrate selection model to generate the optimal video transmission policy.

Considering the dynamic nature of video content characteristics, the hierarchical online adaptive bitrate selection model first employs the upper-level online update decision model to determine whether to send original videos back to the server for retraining. If the upper decision result does not involve sending raw video data, the downstream adaptive bitrate selection model will use another RL model to allocate optimal bitrates for ToI and background tiles, respectively. Otherwise, the system will directly send the raw video data at the highest quality and update the bitrate selection policy to stay aligned with the latest video and network data patterns while minimizing the retraining cost. We next illustrate the detailed design of each module in the following parts.

### B. ToI Selection

Existing saliency or gradient-based approaches [15], [16] cannot capture the feature of small and gathering groups well while the savings from reducing the redundancy space are offset by the fixed area and location of each tile. We use a simple but useful approach to detect ToI and adapt to drone situations. We deploy a detection model with a lightweight backbone such as Mobilenetv2 [17] on drones and use the bounding box generated from detection results for ToI selection. Then overlapped bounding boxes will be integrated into one box to avoid redundant processes. After each video chunk is divided into $m \times n$ tiles, each tile is determined to be a ToI or not, depending on the overlap area between the bounding boxes and the tile. To be specific, we regard tiles with over 5% overlap area with bounding boxes as ToIs. Finally, the coordinates of ToI are fed forward as a state dimension into our bitrate selection model.

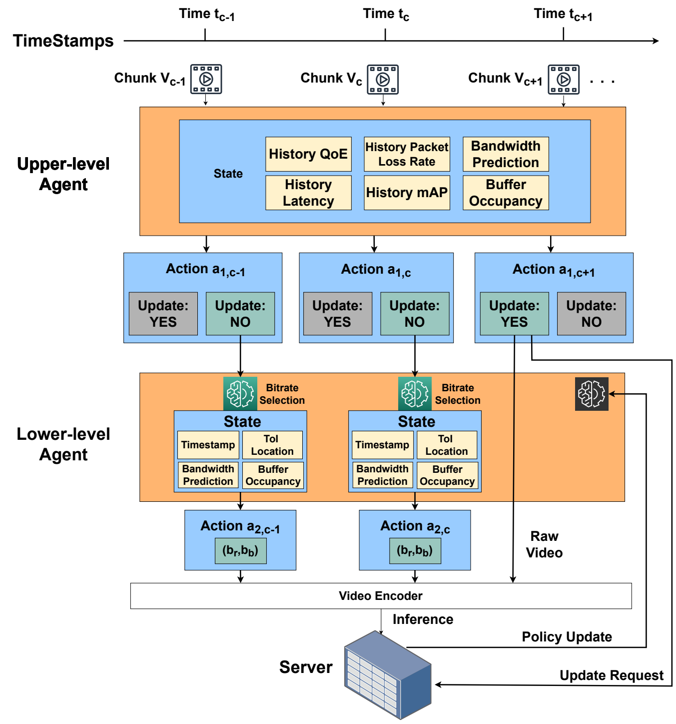



Fig. 5: The architecture of our hierarchical online adaptive bitrate selection model

### C. Online Bandwidth Prediction

As mentioned in Sec. II, existing offline prediction models often struggle to adapt to highly dynamic network conditions. To address this challenge, we propose an online bandwidth prediction network AdaLSTM designed to adapt to real-time changes in drone networks.

We assign an AdaLSTM cell for prediction, capable of updating its hidden state and weights when the prediction loss exceeds a predefined threshold. Specifically, at each time step $t$, given the input bandwidth data $X_t$, and the current hidden state of the AdaLSTM cell $H_t$, the model generates the next hidden state $H_{t+1}$ and an updated output $Y_{t+1}$. The LSTMCell follows the standard LSTM structure. The output of the AdaLSTM cell $H_{t+1}$ is further fed into a fully connected (FC) layer to match the dimension of the input. The updated hidden state and output are used for prediction at the next time step. Additionally, the prediction results are evaluated based on a given window size. If the average loss exceeds the predefined threshold, denoted as $\xi$, the LSTM model is retrained using the ground truth data within the window to minimize the prediction loss. This approach allows the model to continually adapt to the changing patterns in bandwidth usage, enabling accurate predictions while maintaining computational efficiency. This process can further be summarized as follows:

$$H_{t+1} = \text{LSTMCell}\,(X_t, H_t; \Theta), \tag{8}$$

$$Y_{t+1} = FC(H_{t+1}), \tag{9}$$

$$\text{LSTMCell} = \text{Update}\,(\text{LSTMCell}, \xi, Y_{t+1}), \tag{10}$$

where $\Theta$ represents the parameters for the AdaLSTM model, $Y_{t+1}$ is used to compute the average prediction loss fluctuation. Our bandwidth prediction module can learn and adapt to real-time changes in bandwidth usage while maintaining computational efficiency.

### *D. Online Adaptive Bitrate Selection*

In our proposed system, the predicted bandwidth and ToI location results are utilized by the adaptive bitrate selection model. DRL has emerged as a popular approach for determining the optimal bitrate for both the ToI and background in order to maximize QoE. By defining the state space, action space, and QoE objectives, the agent learns the optimization policy through environmental exploration, enabling it to select the most appropriate bitrate for a given state. However, as discussed in our motivation, classical offline trained DRL policy is not robust in drones' dynamic environments. Online training models require original video on a server for getting data labels and retraining. The cost of transmitting raw video without selective processing is prohibitive, potentially leading to high packet loss rates and latency. Naive online adaptive DRL policies lead to unacceptable communication costs and degraded machine-perceived QoE. Consequently, there is an urgent need to update the policy network weight with low communication costs. However, developing a separate decision model to control the update of the policy network is not straightforward. The update decision directly influences whether the adaptive bitrate decision is made, indicating a significant interdependence between the two policies. Additionally, the decision granularity differs between these policies, necessitating a collaborative design of both the adaptive bitrate decision policy and the online update decision policy.

To address this issue, we meticulously design a hierarchical online bitrate adaptation model to intelligently determine the retraining time. The hierarchical framework consists of two decision models as shown in Fig. 5. The upper-level decision model, which determines when raw video data should be sent for retraining, and the lower-level bitrate selection model, which chooses the optimal bitrate for the ToI and background when sending raw video data is deemed unnecessary. In this manner, our model can update the policy network to accommodate dynamic environments, making optimal decisions while maintaining low and acceptable transmission costs. We formulate hierarchical online adaptive bitrate selection as a Markov Decision Process which calls for the design of state, action, and reward. Suppose the upper-level online update decision model is $agent_1$ and the downstream adaptive bitrate selection model is $agent_2$. We present our design for each of them as follows.

**Higher-level agent design.** The upper-level decision model, which determines when raw video data should be sent for retraining, which aims at maximizing adaptive bitrate selection policy improvement while minimizing retraining communication cost. This model is fixed during the inference stage.

- State space $s_{1,c}$: current timestamp $t_c$ and buffer occupancy $B_c$, predicted bandwidth in the coming $n$ timesteps $Bw_c$, history QoE $Q_c$, analysis accuracy $A_c$, latency $L_c$ and packet loss rate $L_c$.
- Action space $a_{1,c}$: the upper-level agent determines whether to send raw video chunk which can be represented as a boolean value $a_{1,c} \in \{1, 0\}$, if $a_1 = 1$ then the video encoder will transmit the video chunk at the highest quality for update bitrate selection policy.
- Reward $r_{1,c}$: the upper-level aims at maximizing adaptive bitrate selection policy improvement while minimizing retraining communication cost. It can be present as $\omega_1(A_{c,update} - A_{c,base}) - \omega_2 C_c$, where $A_{c,update}, A_{c,base}$ is the backend accuracy of video chunk $c$ generated by updated policy and original policy, $C_c$ is the transmission cost introduced by sending raw video chunk at the highest quality. The cost is calculate from $\beta_1 L_c + \beta_2 P_c$. The reward is calculated as zero when it decides not to retrain.

**Lower-level agent design.** The lower-level agent is for adaptive bitrate selection, which chooses the optimal bitrate for the ToI and background tiles when sending raw video data is deemed unnecessary. This model dynamically updates to accommodate dynamic environments.

- State space $s_{2,c}$: current timestamp $t_c$ and buffer occupancy $B_c$, predicted bandwidth in the coming $n$ timesteps $Bw_c$, the location of ToI $R_c$.
- Action space $a_{2,c}$: the lower-level agent selects different bitrate for ToI tiles and background tiles which can be represented as $a_{2,c} = \{b_r, b_b\}$ where $b_r, b_b \in \mathcal{R}$.
- Reward $r_{2,c}$: it can be calculated as $r_{2,c} = QoE_c$ as previously defined.

**Network architecture.** Two DRL-based models have similar architecture. The agent take actions based on the policy $\pi_{\theta}$:$p(s_c, a_c) \longrightarrow [0, 1]$, representing the probability of taking an action $a_c$ at state $s_c$. $\theta$ are the parameters of the policy network. We use the A3C method [18] as the actor-critic training algorithm. Given $\theta$, the gradient of the accumulated discounted reward can be calculated as:

$$\Theta_{\theta} E_{\pi_\theta}[\sum_{c=0}^{n} \gamma^c r^c] = E_{\pi_\theta}[\Theta_\theta log_{\pi_\theta}(s, a) A^{\pi_\theta}(s, a)], \tag{11}$$

$$A^{\pi_\theta}(s, a) = r_c + \gamma V_{\theta_v}^{\pi_\theta}(s_{c+1}) - V_{\theta_v}^{\pi_\theta}(s_c), \tag{12}$$

where $\gamma$ is the discount factor on future rewards, $A^{\pi_\theta}(s, a)$ is the advantage function over action $a$ and $V_{\theta_v}^{\pi_\theta}(s_c)$ is the output of the critic network. The actor and critic network parameters can be updated as below:

$$\theta = \theta + \eta_a \sum_c \Theta_\theta log_{\pi_\theta}(s, a) A^{\pi_\theta}(s, a) + \delta \Theta_{\theta_\alpha} H^{\pi_{\theta_\alpha}}(s_c), \tag{13}$$

$$\theta_v = \theta_v - \eta_v \sum_c \Theta_\theta [A^{\pi_\theta}(s_c, a_c)]^2, \tag{14}$$

where $\eta_a, \eta_v$ is the learning rate for the actor and critic network, relatively. $H(\cdot)$ is the entropy of policy to discourage converging to a sub-optimal policy. Two decision models are trained separately, the downstream bitrate selection model is first trained to develop the optimal policy for bitrate allocation for ToI and background. Then the upper-level online update decision model is trained to maximize the reward of online update current bitrate selection policy while minimizing the cost of transmitting raw video data.

## V. EVALUATION

### A. Experiment Setup

**Drone network trace dataset:** We use a real-world drone network dataset [8] to evaluate the performance of OAVS. This dataset is collected in Munich, Germany, involving drone flights in both urban and rural environments, where drones continuously transfer high-quality RTP-based video over an LTE network to a remote server hosted within the AWS cloud. The bandwidth fluctuates between 0 to 35 Mbps. We selected over 100 traces collected from rural and urban areas, each spanning 100 seconds.

**Drone video datasets:** We choose the VisDrone2019 [10] and SeaDroneSee datasets [11] as our video datasets as we did in the motivation section. Both datasets consist of over 200 video clips, each lasting approximately 10 to 20 seconds, captured by various drone-mounted cameras with various resolutions. These datasets were collected using different drone platforms in various scenarios and under diverse weather and lighting conditions, representing the typical dynamic characteristics of drone video content.

**Baselines:** We compare OAVS with the following baslines.

1) **Rate:** This approach does not utilize tiling and selects the same bitrate for the entire chunk. It assigns each chunk with the maximum bitrate under the predicted bandwidth, supported by a standard LSTM model.
2) **VA_Pensieve:** The RL-based adaptive bitrate algorithm in Pensieve [6] is tailored for video analytics tasks. VA_Pensieve uses the same reward as we do.
3) **ED:** EarlyDiscard [19] is specially designed for drone-sourced video analytics which utilizes a drone-side neural network for frame filtering. It only uploads mission-oriented useful frames. There is no network-aware adaption block.
4) **ACC:** ACC [20] utilizes adaptive resolution for video frames, motion compensation for video encoding and adaptive QP value for tiles which is also specially designed for drone-sourced video analytics. There is no network-aware adaption block.
5) **Offline:** This approach includes a ToI selection part to identify the ToI for each chunk, an RL-based model that uses DQN [21] to assign different bitrates, and a standard LSTM model to predict bandwidth. Unlike our approach, this policy is offline-trained without online adaption.

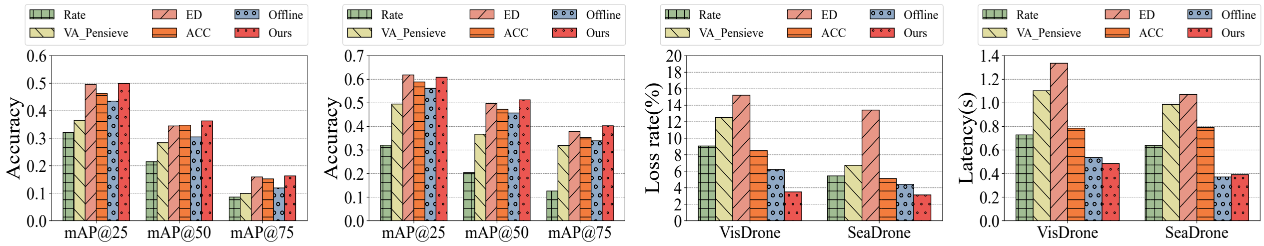
**Parameter setting:** For the online bandwidth prediction model, the hidden size for the basic LSTM cell is 64; The number of LSTM layers is 2; The prediction duration of bandwidth is 1.5s; All the FC layers have a size of 128; The learning rate during the training process and the online-update process is 0.01. For the ToI selection model, we use the small version of yolov4 [22] with Mobilenetv2 [17] as the backbone; For the online DRL bitrate adaptation model, The learning rate for the actor network $\eta_\alpha$ in both upper-level online decision model and bitrate selection model training process is 0.0001. The learning rate for the critic network of both models $\eta_c$ is set 0.0001; The discount factor $\gamma$ is 0.9; The weights in reward modeling are [4, 2, 2] through random search; For the video transmission, we simulate under different drone network traces as introduced in the previous dataset; The candidates bitrates are [5, 10, 15, 20, 25] Mbps; The chunk size of VisDrone2019 and SeaDroneSee are 2 seconds and 1 second per chunk; The high-quality video data used for online policy update is compressed using H264 at QP value of 18; For the backend video analytics, we choose video object detection as evaluation task; The object detection is accomplished by YOLOV5-large version; The training data and validation data is split by 0.8 and 0.2. Our system uses a Jetson TX2 platform with a 256-core NVIDIA Pascal GPU, dual and quad-core CPUs, and 8GB of memory which is compatible with various drone types like DJI M600. The backend server has four 3090Ti GPUs and a 24-core Intel Xeon Gold 5119T CPU.

**Metrics** We use the following metrics to examine OAVS's performance in analytic accuracy, the freshness and the completeness of the analytic result.

- **Mean accuracy.** This metric represents the mean of the server-side DNN inference accuracies for all test video chunks, considering only the received chunks. Since we select object detection as our evaluation task, we use the mean average precision (mAP) for various Intersection over Union (IoU) thresholds (0.25, 0.5, 0.75) as our accuracy metric. This allows us to assess the system's overall performance under different IoU thresholds.
- **Mean upload latency.** This metric calculates the mean upload latency for all test video chunks which reflects the performance of adaptive bitrate selection decisions. We also examine the latency of each component in the end-to-end latency of our system.
- **Loss rate.** This metric is the ratio of unuploaded test video chunks to all test video chunks, representing the proportion of video data that fails to reach the server.

### B. Evaluation Results

**Overall performance.** Fig. 6 compares the performance of streaming various videos using different methods for video analytics tasks. Compared to existing adaptive video streaming methods like Rate, VA_Pensieve, and Offline, our approach reduces the mean loss rate by up to 43.8% and improves the mean accuracy by up to 18.9% (mAP25) in the VisDrone2019 dataset. It also reduces the mean loss rate by up to 31.2% and improves the mean accuracy by up to 16.5% (mAP25) in the SeaDroneSee dataset. Furthermore, OAVS outperforms the existing mean latency by 5.1% in the VisDrone dataset




(a) VisDrone Mean Accuracy (b) SeaDrone Mean Accuracy (c) Mean Loss Rate (d) Mean Upload Latency

Fig. 6: Performance comparison of different methods on two drone datasets

and achieves competitive latency in the SeaDroneSee dataset. Compared to drone-specific video analytics approaches like ED and ACC, our approach reduces the mean loss rate by up to 58.9% and 39.1% on two datasets respectively. It also reduces the mean latency by up to 38.5% and 50.6% on two datasets respectively while achieving similar or even higher mean accuracy. The performance of Rate and VA_Pensieve methods is mainly affected by their non-tile-based design and the inaccuracy of the bandwidth prediction model. The performance of ED and ACC methods is mainly affected by a lack of network-aware bitrate adaption which leads to unacceptable loss rate and upload latency when they come across network fluctuation. Compared to the Offline method, which also applies tile-based bitrate selection, its offline policy struggles to handle complex and fluctuating scenarios. The standard offline LSTM model also is less accurate, leading to worse mean latency and mean loss rates. Although our approach requires continuous raw video transmission and policy updates, our hierarchical decision model compensates for its overhead by choosing the most suitable condition to retrain.

Meanwhile, our approach is proven to meet the requirements of real-time transmission and analysis with low end-to-end latency. As shown in Table I we omit other sources of latency like encoding latency, inference latency, and ToI detection latency. They are usually fixed values or negligible for each approach, with the chunk upload latency being the major latency factor that directly reflects the performance of the video analytics pipeline. All only account for less than 5% latency variation to the overall end-to-end latency. As a result, our method achieves the highest mean accuracy across different drone-sourced video datasets, with the lowest mean latency and mean loss rates.

**Robustness in challenging cases.** We also specifically evaluated the performance of our approach for small objects and fluctuating network traces. As for the performance for small object sizes, we select the objects that take no more than 0.1% area of the frame size and evaluate analytics accuracy of them. Compared to Rate, VA Pensieve, and Offline baselines, our approach increases their accuracy by 97.3%, 68.4%, and 28.1%, respectively, achieving accuracy comparable to that of ED and ACC approaches. As for the performance under fluctuating network conditions, we select the most challenging

TABLE I: End-to-end latency decomposition for OAVS.

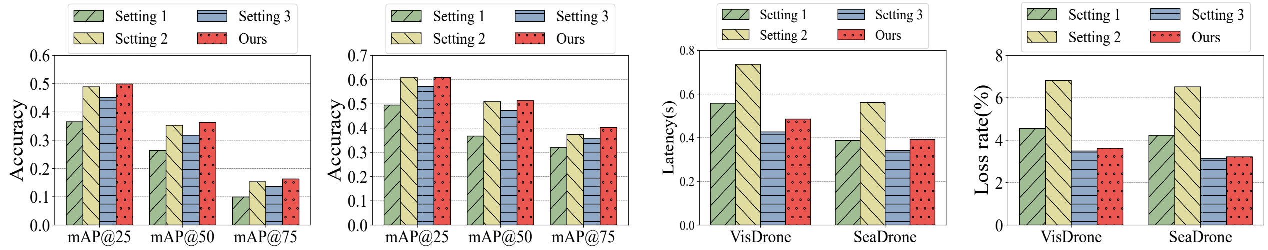
| Stage | Frame(FPS) | Time(ms) |
|---|---|---|
| ToI Selection | 108.7 | 9.2 |
| Policy Network Inference | 840.3 | 1.2 |
| Frames Encoding | 412.6 | 2.4 |
| Video Uploading | - | 427.3 |
| RTT | - | 20.0 |
| Server-side Inference | 98.0 | 10.2 |

network traces in the dataset with handover frequency over 0.1 times per second and bandwidth variation over 10Mbps and evaluate their loss rate and latency. For Rate, VA_Pensieve, and Offline approach(we did not consider ED and ACC approaches because they have no bitrate adaption block), the mean loss rates are 10.1%, 16.0%, and 8.8% respectively while ours is 4.1%. The mean transmission latency is 0.78s, 1.52s, and 0.66s, respectively, while ours is 0.48s. The mean end-to-end latency is 0.81s, 1.54s, and 0.70s, respectively, while ours is 0.51s.

**Ablation study.** We propose three decomposed settings of `OAVS` to examine the contribution of each block in our system. Based on `OAVS`, *Setting 1* deletes ToI selection with tile-based bitrate allocation. *Setting 2* replaces AdaLSTM with standard LSTM for bandwidth prediction and *Setting 3* replaces online adaptive streaming with fixed adaptive streaming policy.

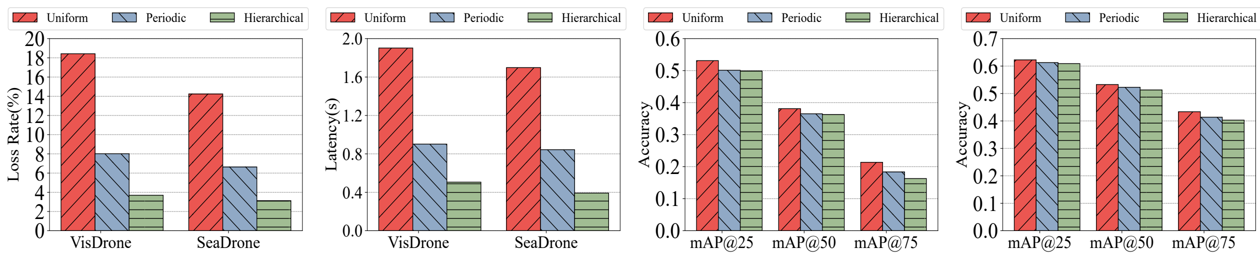
The performance of each configuration is presented in Fig. 7. Tile-based bitrate allocation results in an approximately 23.6% mean accuracy increase, and the online-update policy leads to around an 11.2% mean accuracy increase on VisDrone. For mean latency, the online-update bandwidth prediction leads to a 50.1% reduction and tile-based bitrate allocation leads to a 21.5% reduction. Benefiting from the hierarchical online adaptive bitrate selection model, the latency still remains competitive with the offline system due to the dynamic decision model, and the improvements in accuracy and loss rate are much more significant. The loss rate decreases by 45.4% due to our online-updated prediction and upper-level decision model, nearly eliminating the influence of sending high-quality video data on the loss rate.

We also examine the efficiency of the hierarchical decision structure in Fig. 8. We compare the system performance under our learning-based retraining model with the uniform strategy




(a) VidsDrone Mean Accuracy (b) SeaDrone Mean Accuracy (c) Mean Upload Latency (d) Mean Loss Rate

Fig. 7: Performance comparison of different system blocks for ablation experiments




(a) Mean Loss Rate (b) Mean upload Latency (c) VisDrone Mean Accuracy (d) SeaDrone Mean Accuracy

Fig. 8: Performance comparison of adopting different online update strategies

and the one under the periodic retraining strategy which is used in [23]. We find that compared with sending new data for retraining periodically, the learning-based retraining decision model results in an 81.3% reduction in loss rate and a 69.5% reduction in latency, with only a 5.38% loss in accuracy.

**Overhead analysis.** We examine the retraining overhead for AdaLSTM and reveal that this module requires few resources for training and can be easily supported by existing drone-compatible computing devices, like NVIDIA Jetson TX2. Specifically, the training process of the LSTM model only consumed around 832MB GPU memory, which took around 10.4% of the total available GPU resources (8GB) and around 5.7% of the maximum operating power which is far less than the overall resources consumption of the drone system. Meanwhile, the decision of retraining is made according to the average performance of a period of past time so that it can prevent retraining decisions caused by temporarily drastic network condition changes. After further considering the stringent real-time requirement for prediction, we design this module to be fully run on drones without offloading to servers and not include it in the hierarchical decision model which we will introduce next. Meanwhile, we also examine the inference overhead of the adaptive bitrate selection model and reveal that it consumes around 456MB GPU memory, around 5.7% of the GPU capacity, and around 4.5% of the maximum operating power. The retraining process can be accomplished within the processing time of around three chunks which enables real-time updates for the streaming policy. This analysis proves our system is real-time and lightweight enough for practical application.

## VI. RELATED WORK

**Adaptive streaming for video analytics.** Recent work has begun to explore adaptive video streaming tailored specifically for video analytics [24]–[27], in contrast to the traditional human-centric streaming frameworks [4], [28]–[30]. Chameleon [26] is another system that focuses on efficient computational resource management for large-scale streaming analytics in a single data center or cloud. AWStream [31] is a wide-area streaming analytics system that adjusts the application data rate to match available bandwidth while maximizing accuracy. AdaDSR [32] proposes to dynamically select the optimal downsampling and upscaling ratios at the client and server side, respectively to balance the tradeoff among analytics accuracy, transmission cost, and computation cost. However, these works do not directly address the unique challenges of drone video analytics in dynamic and resource-constrained environments. They either rely on a simple method for network throughput estimation and profiling or offline trained models, making it difficult to drone's dynamic environments. On the other hand, DDS [16] provides a server-driven video streaming method for DNN inference. Unlike DDS, our proposed `OAVS` system is specifically designed for drone-sourced video analytics scenarios using a client-driven design framework to avoid the delay of waiting for the server's feedback. Efficient online learning approaches are proposed to further improve analytics performance.

**Drone-sourced video analytics.** Processing high-resolution video on drones is extremely challenging due to limited resources [33]. Several works have proposed techniques to overcome these challenges. For instance, systems like EarlyDiscard [19] and ACC [20] focus on optimizing on-device processing and communication for drone-sourced video analytics. These approaches employ techniques such as frame filtering, and compression to reduce the computational and communication overhead associated with drone-sourced video analytics. While these works provide valuable insights for improving drone-sourced video analytics, they do not specifically address the problem of adaptive streaming in this context. Our proposed `OAVS` system fills this gap by introducing a novel online learning-based framework that addresses the unique challenges of adaptive streaming for drone video analytics, achieving significant performance improvements over existing techniques.

## VII. CONCLUSION

The complex drone network conditions and the intrinsic dynamics of video content necessitate the development of a robust and efficient adaptive video streaming system for drone-sourced video analytics. Existing offline-trained network prediction models and streaming policy models suffer from significant accuracy degradation and latency increase. In this paper, we proposed `OAVS`, the first online adaptive video streaming framework for drone-sourced video analytics. We judiciously designed a new hierarchical deep reinforcement learning approach to effectively determine when to conduct online retraining and determine the appropriate bitrate for tiles of a video chunk. Our design balances the tradeoff between analytics accuracy, freshness, and completeness of analytics results. Extensive experiments using real-world drone network traces and video datasets demonstrate the superior performance of `OAVS` compared to state-of-the-art streaming methods.

## REFERENCES


[1] J. Sagar, L. Youjie, N. Shadi, R. Vaishnavi, K. Peeyush, N. Andrew, T. Michael, S. Sudipta, C. Ranveer, and B. Anirudh, "Visage: Enabling timely analytics for drone imagery," in *Proc. ACM MobiCom*, 2021, pp. 789–803.

[2] J. Cauchard, J. E, K. Zhai, and J. Landay, "Drone & me: an exploration into natural human-drone interaction," in *Proc. ACM UbiComp*, 2015, pp. 361–365.

[3] F. B. Insights, "Unmanned aerial vehicle (uav) market size and growth projection," Market Research Report, p. 160, 2020.

[4] T. Stockhammer, "Dynamic adaptive streaming over http– standards and design principles," in *Proc. ACM MMSys*, 2011, pp. 133–144.

[5] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, 2011.

[6] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. ACM SIGCOMM*, 2017, pp. 197–210.

[7] Y. Wang, W. Wang, J. Zhang, J. Jiang, and K. Chen, "Bridging the {Edge-Cloud} barrier for real-time advanced vision analytics," in *USENIX HotCloud*, 2019.

[8] A. Baltaci, H. Cech, N. Mohan, F. Geyer, V. Bajpai, J. Ott, and D. Schupke, "Analyzing real-time video delivery over cellular networks for remote piloting aerial vehicles," in *Proc. ACM IMC*, 2022, pp. 98–112.

[9] G. Van de Ven and A. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[10] P. Zhu, D. Du, L. Wen, X. Bian, H. Ling, Q. Hu, T. Peng, J. Zheng, X. Wang, and Y. e. a. Zhang, "Visdrone-vid2019: The vision meets drone object detection in video challenge results," in *Proc. IEEE/CVF ICCV*, 2019.

[11] L. Varga, B. Kiefer, M. Messmer, and A. Zell, "Seadronessee: A maritime benchmark for detecting humans in open water," in *Proc. IEEE/CVF WACV*, 2022, pp. 2260–2270.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] A. Fakhreddine, C. Bettstetter, S. Hayat, R. Muzaffar, and D. Emini, "Handover challenges for cellular-connected drones," in *Proc. ACM DroNet*, 2019, pp. 9–14.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, pp. 303–338, 2010.

[15] K. Fischer, F. Fleckenstein, C. Herglotz, and A. Kaup, "Saliency-driven versatile video coding for neural object detection," in *Proc. IEEE ICASSP*, 2021, pp. 1505–1509.

[16] K. Du, A. Pervaiz, X. Yuan, A. Chowdhery, Q. Zhang, H. Hoffmann, and J. Jiang, "Server-driven video streaming for deep learning inference," in *Proc. ACM SIGCOMM*, 2020, pp. 557–570.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE CVPR*, 2018, pp. 4510–4520.

[18] V. Mnih, A. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. PMLR ICML*, 2016, pp. 1928–1937.

[19] J. Wang, Z. Feng, Z. Chen, S. George, M. Bala, P. Pillai, S.-W. Yang, and M. Satyanarayanan, "Bandwidth-efficient live video analytics for drones via edge computing," in *Proc. IEEE/ACM SEC*, 2018, pp. 159–173.

[20] J. Guo and C. Peng, "Towards drone-sourced live video analytics via adaptive-yet-compatible compression," in *Proc. ACM HotMobile*, 2021, pp. 172–173.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[22] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[23] R. Bhardwaj, Z. Xia, G. Ananthanarayanan, J. Jiang, Y. Shu, N. Karianakis, K. Hsieh, P. Bahl, and I. Stoica, "Ekya: Continuous learning of video analytics models on edge compute servers," in *Proc. USENIX NSDI*, 2022, pp. 119–135.

[24] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. Freedman, "Live video analytics at scale with approximation and delay-tolerance," in *Proc. USENIX NSDI*, 2017.

[25] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, "Deepdecision: A mobile deep learning framework for edge video analytics," in *Proc. IEEE INFOCOM*, 2018, pp. 1421–1429.

[26] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *Proc. ACM SIGCOMM*, 2018, pp. 253–266.

[27] Y. Li, A. Padmanabhan, P. Zhao, Y. Wang, G. Xu, and R. Netravali, "Reducto: On-camera filtering for resource-efficient real-time video analytics," in *Proc. ACM SIGCOMM*, 2020, pp. 359–376.

[28] Y. Lu, Y. Zhu, and Z. Wang, "Personalized 360-degree video streaming: A meta-learning approach," in *Proc. ACM MM*, 2022, p. 3143–3151.

[29] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of quality of experience of video-on-demand services: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 401–418, 2015.

[30] H. Yeo, Y. Jung, J. Kim, J. Shin, and D. Han, "Neural adaptive content-aware internet video delivery," in *Proc. USENIX OSDI*, 2018, pp. 645–661.

[31] B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E. Lee, "Awstream: Adaptive wide-area streaming analytics," in *Proc. ACM SIGCOMM*, 2018, pp. 236–252.

[32] S. Cen, M. Zhang, Y. Zhu, and J. Liu, "Adadsr: Adaptive configuration optimization for neural enhanced video analytics streaming," *IEEE Internet of Things Journal*, 2023.

[33] T. Huang, C. Zhou, R. Zhang, C. Wu, X. Yao, and L. Sun, "Comyco: Quality-aware adaptive video streaming via imitation learning," in *Proc. ACM MM*, 2019, pp. 429–437.