Accelerating Real-Time Multi-View Edge Video Analytics with Spatial-Temporal Correlation

Jiachen Sun, Nan Wang, Yinglei Teng, *Senior Member, IEEE*
Beijing Key Laboratory of Space-ground Interconnection and Convergence
Beijing University of Posts and Telecommunications (BUPT), Xitucheng Road No.10, Beijing, China, 100876.
Email: sunjiachen@bupt.edu.cn, wangnan_26@bupt.edu.cn, lilytengtt@gmail.com

***Abstract*—Advances in edge computing and video analytics are driving the widespread use of real-time IoT-based sensing and monitoring systems at the network edge. However, as camera clusters expand and video resolutions increase, meeting stringent response time requirements for video analytics is increasingly challenging due to limited computational resources and variable network bandwidth. To address these challenges, we propose a spatial-temporal feature-aware node selection scheme designed for multi-view edge video analytics. Our approach leverages spatial redundancy across video feeds to dynamically select specific processing nodes, while exploiting temporal correlations from prior frame detection to guide subsequent frame analysis, thereby reducing the computational load and enhancing system efficiency. Furthermore, we formulate an optimization problem for joint node selection and resource allocation, balancing the trade-off between end-to-end latency and inference accuracy. To tackle the complex coupling and temporal correlation, we propose a two-stage optimization scheme (PPO-LAG), that resolves the action dimensionality of Proximal Policy Optimization (PPO) through iteratively applying the Lagrange multiplier for inter-resource allocation. Experimental results on Wildtrack dataset show that our proposed scheme comparatively achieves higher inference accuracy while adaptively reducing the total latency.**

***Index Terms*—spatial-temporal correlation, node selection, real-time video analytics, edge collaboration inference.**

I. Introduction

In the Artificial Intelligence of Things (AIoT) era, the decreasing costs of Internet of Things (IoT) devices such as high-definition (HD) cameras and the growing capabilities of deep artificial intelligence (AI) models, are boosting real-time sensing and surveillance [1]. However, as these video analytics tasks become increasingly time-sensitive, it poses growing demand for substantial computational resources. Supported by edge computing [2], video frames can be offloaded to nearby edge servers (ESs), which not only alleviates the computational burden on mobile devices (MDs) but also avoids the substantial latency caused by interactions with remote cloud centers [3]. Nevertheless, with the increase in MDs and the rising complexity of video analytics, traditional algorithms struggle to meet the real-time demand of video processing. This necessitates environment-adaptive edge video analytics to effectively balance accuracy and latency.

To reduce end-to-end latency, the spatial-temporal redundancy of video frames is exploited to decrease the computation load of video processing. Considering that detection outputs tend to overlap in successive frames, Arefeen *et al.* [4] proposed a reinforcement learning-based method that dynamically skips frames along the time dimension to reduce the number of frames for processing. Xiao *et al.* [5] introduced a method of applying lower compression for target regions to maintain detection accuracy and higher compression for background areas. Wang *et al.* [6] designed a framework that transmits each target object only when it reaches the position in the frame for optimal recognition and focusing on the target region spatially to avoid repeated processing. The above work greatly improved video processing performance for single-camera setups; however, scenarios involving multi-camera joint analytics often introduces correlations across different views, which increases the complexity of video analytics.

In the multi-camera collaborative scenario, the redundancy between multiple views can also be leveraged to further enhance performance. Guo *et al.* [7] introduced a method to remove irrelevant regions across different views and compress consecutive frames at varying compression rates. Liu *et al.* [8] enabled information sharing among multiple cameras, ensuring that only one camera tracks overlapping regions. Dai *et al.* [9] proposed prioritizing the uploading of frames with higher information and calculating the similarity between adjacent frames to reduce the number of frames transmitted and processed. These methods effectively reduce redundancy and enhance transmission efficiency. However, under limited bandwidth and varying channel conditions, the processing capabilities among MDs are heterogeneous. Inefficient resource allocation can significantly decrease system efficiency.

In this paper, we investigate the multi-view edge video analytics and propose a spatial-temporal correlation-based node selection scheme for real-time device-edge collaborative computation. Preliminary experiments reveal temporal correlations among neighboring frames and spatial redundancy across multi-streams. Based on this, we formulate a joint node selection and resource optimization problem to enhance online inference accuracy and reduce latency. We also propose a Lagrange-based two-stage Proximal Policy Optimization (PPO-LAG) algorithm, which reduces computational complexity dramatically. Experimental results demonstrate that our approach outperforms baselines, achieving higher accuracy and lower latency.

II. Preliminary Experiments

In this section, we explore an alternative metric to represent inference accuracy without real-time analytics ground truth,

and investigate the temporal correlation and spatial redundancy of multi-view frame sequences through experiments.

(1) An Alternative Metric to Represent Multi-View Video Inference Accuracy without Ground Truth: Multi-view video object detection integrates features from multiple cameras or viewpoints to improve object detection and tracking accuracy. It is typically assessed using metrics like Multi-Object Detection Accuracy (MODA) [10], which considers normalized missed detections and false positives across the video sequence. However, in real-time analytics tasks, MODA is difficult to obtain without ground truth [11]. To address this limitation, we propose an alternative method that uses the average output $\bar{g}_i$ across frames to approximate per-frame inference accuracy,

$$\bar{g}_i = \frac{1}{RC} \sum_{row=1}^{R} \sum_{col=1}^{C} g_{i,row,col}, \tag{1}$$

where i denotes the i-th frame in the video sequence, R and C are the number of rows and columns in the output grid, respectively. $g_{i,row,col}$ denotes the target presence probability at each grid location.

To demonstrate the usability of our proposed metric, we analyze the correlation between average $\bar{g}_i$ and MODA on the Wildtrack dataset. Notably, since MODA is typically calculated over the entire test set, we sum all $\bar{g}_i$ values across frames to obtain the average inference accuracy. In Fig. 1, the average inference accuracy[1] shows a strong correlation with MODA, with a Pearson coefficient of 0.8 and a p-value of 1.77×10^{-23}. This significant statistical correlation suggests that $\bar{g}_i$ can effectively represent inference accuracy in the absence of ground truth.

(2) Temporal Correlation in Multi-View Video Data Streams: In continuous video streams, there exists a temporal correlation between consecutive frames [11]. To quantify this correlation in multi-view video frame sequences, Fig. 2 plots the autocorrelation of inference accuracy on the dataset Wildtrack. We observe that the autocorrelation is larger than 0.5 and remains above the blue shaded areas[2] when *Lag* = 3, indicating a significant correlation in the inference accuracy of multi-view sequence frames over time. This insight motivates the use of prior inference accuracy so as to guide the process for subsequent frames.

(3) Spatial Redundancy of Multi-View Video Frames: Since multi-view video capture scenes from different viewpoints, each camera records different numbers of objects depending on their spatial distribution. In Fig. 3, we calculate the average number of objects captured per frame using different cameras across different datasets. We observe that the majority of objects are captured by five to six cameras in both Fig. 3(a) and Fig. 3(b). Hence, we conclude that adaptively selecting certain MDs or directly turning off some cameras can optimize total latency with minimal degradation in inference accuracy.

[1]Each data point represents a different configuration where a specific subset of cameras is deactivated, with missing feature maps replaced by zero-filled maps.

[2]Blue shaded areas indicate 95% confidence intervals, outside of which autocorrelation coefficients can be considered significant.

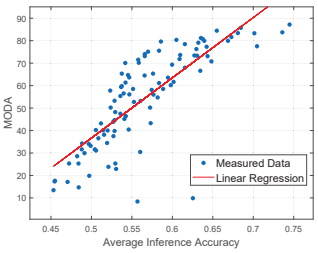


Fig. 1. Correlation between average inference accuracy and MODA.

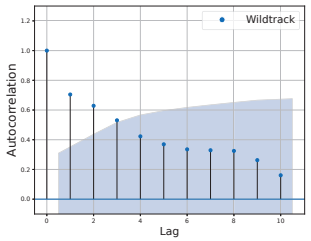
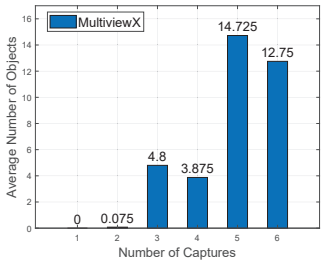
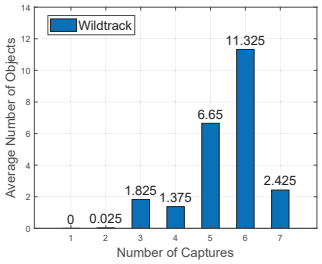


Fig. 2. Autocorrelation of the inference accuracy.



(a) On the dataset MultiviewX [12].



(b) On the dataset Wildtrack [13].

Fig. 3. Average number of objects captured per frame across different datasets.

III. System Model

As shown in Fig. 4, we consider a collaborative edge multi-view video analytics system, which comprises an ES and K MDs, i.e., $\mathcal{K} = \{1, 2, ..., K\}$. Each MD is equipped with a camera and N_t antennas, while the ES is equipped with N_r antennas. Assume a multi-view pedestrian detection scenario, such as video analytics for intelligent traffic monitoring. We implement a Q-layer DNN model and partition it into two segments at layer Q_{fe}, where Q_{fe} in $\mathcal{Q}$ represents the cumulative number of layers for feature extraction, with $\mathcal{Q} = \{1, 2, ..., Q\}$. Meanwhile, considering the limited computational resources and battery capacity of MDs, we deploy the feature extraction portion of the well-trained model on each MD, while placing the feature aggregation and inference segments on the ES.

We denote the consecutive frames as the set $\mathcal{I} = \{1, 2, ..., I\}$, each separated by an equal duration Δ. In each frame, in order to enhance the real-time processing capability of the system, we use a node selection strategy $\beta_{k,i}$ to decide whether MD k performs feature extraction as well as transmits the feature maps to the ES. Specifically, $\beta_{k,i} = 1$ indicates that MD k is selected to extract image features and transmit them to the ES, while $\beta_{k,i} = 0$ means the MD k is not selected. Once receiving the feature maps from all selected nodes, the ES concatenates them and uses DNN for inference, with the results also being available for visualization. Since inference results are crucial for task decision-making, the ES needs to send the results back to the MD after generating them.

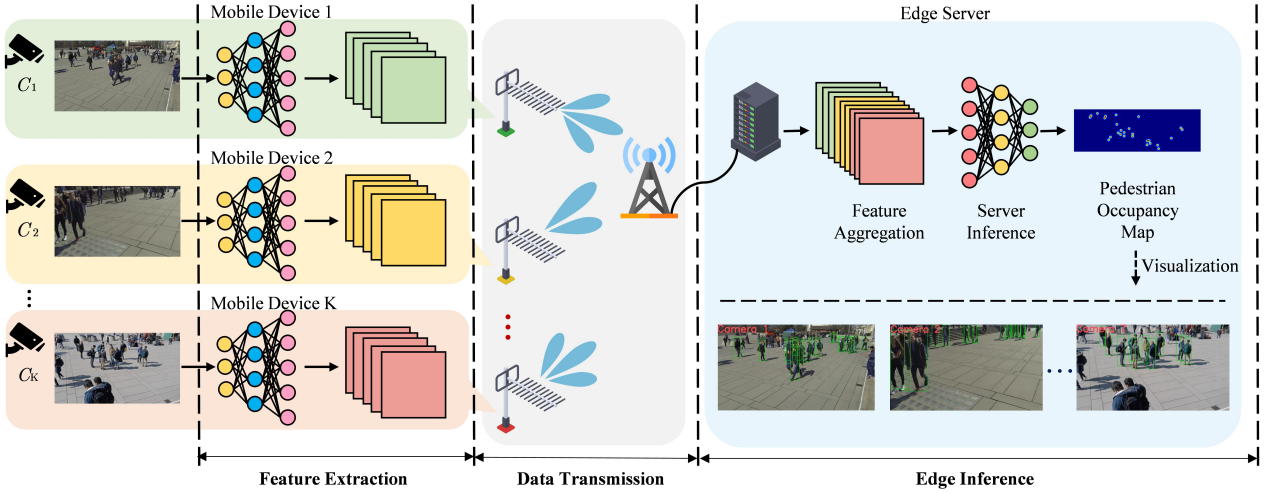


Fig. 4. Edge collaborative multi-view video analytics system.

However, we usually ignore the downlink transmission latency [14] due to the small data sizes.

A. End-to-End Latency Model

For the real-time collaborative edge video analytics, we take the complete end-to-end latency into account, which includes three components: execution latency in MDs, communication latency from MDs to ES, and execution latency in ES.

1) Execution latency in MDs: The execution latency for feature extraction at MD k during the i-th frame can be represented as

$$T_{k,i}^{loc} = \frac{\beta_{k,i}\rho_k \sum_{q=1}^{Q_{fe}} c_{k,q}}{f_{k,i}}, \tag{2}$$

where ρ_k represents the number of CPU cycles required to complete a floating-point operation on MD k, $c_{k,q}$ denotes the floating-point operations for the q-th layer of the DNN model, and $f_{k,i}$ is the CPU frequency of MD k.

2) Communication latency: Considering a multiuser mmWave massive MIMO beamforming system for uplink transmission, we apply a statistical channel model for mmWave outdoor transmission, using a standard uniform linear antenna array at 25 GHz. Due to channel sparsity and high free-space path loss, we use the extended Saleh-Valenzuela geometric model. For the i-th frame of the video stream, the channel matrix from MD k to the ES can be expressed as

$$H_{k,i} = \sqrt{\frac{N_t N_r}{L_p}} \sum_{l=1}^{L_p} \alpha_{l,k,i} a_{r,i}(\phi_{l,k,i}) a_{t,i}(\omega_{l,k,i})^H, \tag{3}$$

where $H_{k,i} \in \mathbb{C}^{N_r \times N_t}$ represents the mmWave channel matrix, N_t is the number of antennas at MD k, and N_r is the number of antennas at the ES. L_p is the number of distinguishable paths, and $\alpha_{l,k,i} \sim \mathcal{CN}(0,1)$ represents the complex gain of the l-th path from MD k to the ES. $a_{r,k,i}(\phi_{l,k,i})$ and $a_{t,k,i}(\omega_{l,k,i})$ are the receive and transmit antenna array response vectors, where $\phi_{l,k,i} \in [0, 2\pi)$ and $\omega_{l,k,i} \in [0, 2\pi)$ are the azimuth angles of arrival and departure (AoAs and AoDs). The array response vectors are given by

$$a_i(\theta) = \frac{1}{N}[1, e^{jkd_a \sin(\theta_i)}, \ldots, e^{jkd_a(N-1)\sin(\theta_i)}]^T, \tag{4}$$

where θ represents the azimuth angles of arrival and departure (AoAs and AoDs). N is the number of transmitting antennas N_t or receiving antennas N_r. $k = 2\pi/\mu$, where μ is the wavelength, and d_a is the antenna spacing. In this system, we adopt regularized zero-forcing (RZF) digital precoding to process the transmission signals of each MD. This approach reduces interference between different users and antennas, thereby enhancing the signal-to-interference-plus-noise ratio (SINR) for each user. The precoding matrix $W_i \in \mathbb{C}^{KN_t \times N_r}$ is defined as

$$W_i = H_i^H \left(H_i H_i^H + \alpha I\right)^{-1}, \tag{5}$$

where $H_i \in \mathbb{C}^{N_r \times KN_t}$ is the channel matrix. The signal-to-interference-plus-noise ratio (SINR) of MD k can be expressed as

$$SINR_{k,i} = \frac{p_{k,i}^{trans} \|H_{k,i} W_{k,i}\|^2}{\sum_{l \neq k} p_{l,i}^{trans} \|H_{k,i} W_{l,i}\|^2 + \sigma^2}, \tag{6}$$

where $W_{k,i} \in \mathbb{C}^{N_t \times N_r}$ is the precoding matrix of MD k. $p_{k,i}^{trans}$ denotes the transmission power of MD k. Let $p_i = [p_{1,i}^{trans}, \ldots, p_{K,i}^{trans}]^T$ represent the transmission power vector for each MD. σ^2 is the variance of the zero-mean additive white Gaussian noise. After extracting features from the i-th frame, MD k transmits the generated results to the ES over the mmWave massive MIMO channel at a transmission rate given by

$$r_{k,i} = B \log_2 \left(1 + SINR_{k,i}\right), \tag{7}$$

where B represents the system bandwidth. The transmission latency for MD k can be represented as

$$T_{k,i}^{trans} = \frac{\beta_{k,i} b_{k,i}^{trans}}{r_{k,i}}, \tag{8}$$

where $b_{k,i}^{trans}$ is the size of data transmitted by MD k. Since digital precoding does not involve compression or other operations, the data size matches the original feature maps. The processing latency for each MD can be expressed as

$$T_{k,i} = T_{k,i}^{loc} + T_{k,i}^{trans}. \tag{9}$$

The processing latency for all selected MDs is given by the maximum of the execution latency and transmission latency,

$$T_i^{\psi} = \max_{k \in \mathcal{K}} T_{k,i}. \tag{10}$$

3) Execution latency in the ES: Similarly, the execution latency for the ES during the i-th frame can be represented as

$$T_i^{es} = \frac{\rho_{es} \sum_{q=Q_{fe}+1}^{Q} c_q}{f_i^{es}}, \tag{11}$$

where ρ_{es} denotes the number of CPU cycles required by the ES to complete one floating-point operation, c_q represents the floating-point operations for the q-th layer of the DNN model, and f_i^{es} represents the CPU frequency of the ES. Due to the sequential computation by MDs and the ES, the total latency T_i^{total} can be expressed as

$$T_i^{total} = T_i^{\psi} + T_i^{es}. \tag{12}$$

B. Power Consumption Model

To evaluate the power requirements of the collaborative edge video analytics system, we consider the primary sources of power consumption: execution power at the MDs, data transmission by the MDs, and execution power at the ES. The power consumption for each MD can be represented as

$$p_{k,i}^{loc} = \beta_{k,i} \varepsilon_k (f_{k,i})^3, \tag{13}$$

where ε_k is the computation energy efficiency coefficient related to the processor's chip equipped at MD k. The power consumption p_i^{es} is similar to $p_{k,i}^{loc}$, expressed as

$$p_i^{es} = \varepsilon_{es} (f_i^{es})^3, \tag{14}$$

where ε_{es} is related to the processor's chip equipped at the ES. The power consumption $p_{k,i}^{trans}$ for MD k to transmit feature maps to the ES is determined by $\beta_{k,i}$. If $\beta_{k,i} = 1$, then $p_{k,i}^{trans}$ represents the power required for MD k. Conversely, if $\beta_{k,i} = 0$, MD k does not need to transmit any data, and $p_{k,i}^{trans} = 0$.

IV. Problem Formulation

A. Original Problem Formulation

Based on the above system model, we adopt a long-term weighted sum approach to simultaneously maximize the inference accuracy and minimize the total latency. To represent the trade-off between these objectives, we introduce a positive weight parameter λ, which reflects the bias between inference accuracy and total latency. The following multi-objective optimization problem ($\mathcal{P}1$) is formulated as follows:

$$\begin{aligned} \mathcal{P}1: & \max_{\boldsymbol{\beta},\mathbf{f},\mathbf{p}} \lim_{I \to \infty} \frac{1}{I} \sum_{i \in \mathcal{I}} \left(\bar{g}_i - \lambda T_i^{total} \right), \\ \text{s.t.} \quad & C_1: \beta_{k,i} \in \{0,1\}, \forall k, \forall i, \\ & C_2: 0 \le f_{k,i} \le f_{\max}^{loc}, \forall k, \forall i, \\ & C_3: 0 \le f_i^{es} \le f_{\max}^{es}, \forall i, \\ & C_4: 0 \le p_{k,i}^{trans} \le p_{\max}^{trans}, \forall k, \forall i, \end{aligned} \tag{15}$$

where $\boldsymbol{\beta} \triangleq \{\beta_{k,i}\}$, $\mathbf{f} \triangleq \{f_{k,i}, f_i^{es}\}$, $\mathbf{p} \triangleq \{p_{k,i}\}$. Constraint C_1 implies node selection is a binary variable and ensures the feasibility of the node selection decision. Constraints C_2 and C_3 enforce the range of the computational frequency on each MD and the ES, respectively. Constraint C_4 guarantees that the transmission power is adjusted within the tolerable range for each MD.

Obviously, $\mathcal{P}1$ is a highly complex stochastic optimization problem, covering both discrete and continuous variables. This problem is highly coupled in many aspects and non-convex. Furthermore, the objective function relies on the node selection decision $\boldsymbol{\beta}$ as a prerequisite for optimizing computational capacity $\mathbf{f}$ and transmission power $\mathbf{p}$. Thus, we separate the original problem and propose a two-stage method. Specifically, for the node selection strategy, we use a deep reinforcement learning method due to the time-dependency and no closed-form expression. Meanwhile, given the convex nature of the inter-resource allocation problem, we apply the Lagrange multipliers method to allocate computational capacity and transmission power. The details are shown in **Algorithm 1**.

B. The First Stage Algorithm

Considering the complexity of the problem and the sparsity of the action variables, we use the PPO algorithm to solve it for better rewards. We define the necessary states, actions, and reward as follows:

1) State Space: After receiving the reward r_{i-1} from the previous time step, the action a_{i-1} from the previous time step is combined with it to form an action-observation pair [11], i.e.,

$$z_i = (r_{i-1}, a_{i-1}). \tag{16}$$

Thus, the state of the i-th frame can be described as

$$s_i = (z_{i-\upsilon+1}, \ldots, z_i). \tag{17}$$

where υ represents the history length of the action-observation pairs.

2) Action Space: For the i-th frame, the collaborative edge inference system determines the node selection of each MD based on the PPO algorithm. The action is defined as

$$a_i = \boldsymbol{\beta}_i \tag{18}$$

3) Reward Function: For the real-time video analytics, the reward function is defined as the difference between inference accuracy and the total end-to-end latency:

$$r_i = \bar{g}_i - \lambda T_i^{total}. \tag{19}$$

Algorithm 1: Lagrange-Based Multi-Objective Two-Stage PPO Algorithm for Adaptive Node Selection and Resource Allocation (PPO-LAG)

Output: $\boldsymbol{\beta}$, $\mathbf{f}$, $\mathbf{p}$;

Randomly initialize the actor network with θ;
Randomly initialize the critic network with ϕ;
Initialize the old actor parameter with $\theta_{old} \leftarrow \theta$;
Initialize the buffers $\mathcal{B} \leftarrow \varnothing$;
Set parameters Z, I,$|\mathcal{B}|_{\max}$, N , ξ_a, ξ_c;
for episode = 1:Z **do**
  for step = 1:I **do**
    Observe the environment $z_i = (r_{i-1}, a_{i-1})$ and compute the state s_i by (17);
    Select an action using actor $\alpha \sim \pi_{\theta_{old}}(\cdot|s_i)$;
    Compute $\mathbf{f}$ and $\mathbf{p}$, according to the **Lagrange multipliers method** (described in Section IV.C);
    Observe the current reward r_i and the next state s_{i+1};
    Collect the buffer $\mathcal{B} \leftarrow \mathcal{B} \cup (s_i, a_i, r_i, s_{i+1})$;
    if $|\mathcal{B}| = |\mathcal{B}_{\max}|$ **then**
      for epoch = 1:N **do**
        Train actor network:
        Compute $\mathcal{L}(\theta)$ by (20);
        Update the actor parameter: $\theta \leftarrow \theta - \xi_a \nabla_\theta \mathcal{L}(\theta)$;
        Train critic network:
        Compute $\mathcal{L}(\phi)$ by (21);
        Update the critic parameter: $\phi \leftarrow \phi - \xi_c \nabla_\phi \mathcal{L}(\phi)$;
      end for
      Update $\theta_{old} \leftarrow \theta$;
      Clear the replay buffer $\mathcal{B} \leftarrow \varnothing$;
    end if
  end for
end for

Based on the advantage actor-critic (A2C) framework, the objective function for the actor network is given by

$$\mathcal{L}(\theta) = \hat{\mathbb{E}}_i \left[\min \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{old}}(a_i|s_i)} \hat{A}_{\theta_{old}}(s_i, a_i), \right. \right. \\ \left. \left. \text{clip}\left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{old}}(a_i|s_i)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{\theta_{old}}(s_i, a_i) \right) \right], \tag{20}$$

where $\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{old}}(a_i|s_i)}$ represents the probability ratio between the new and old policies, $\hat{A}_{\theta_{old}}(s_i, a_i)$ is the advantage function estimated using the generalized advantage estimation (GAE), and ϵ denotes the clipping rate. The clip function $\text{clip}(\cdot)$ prevents excessively large policy updates, reducing the risk of catastrophic performance degradation. The critic network is updated by minimizing the following loss function

$$\mathcal{L}(\phi) = \hat{\mathbb{E}}_i \left[\left(V_\phi(s_i) - \hat{V}_i \right)^2 \right], \tag{21}$$

where $V_\phi(s_i)$ indicates the value function approximated with the critic network, and $\hat{V}_i$ is the target value adopting GAE, combining discounted rewards and advantages from state s_i.

C. The Second Stage Algorithm

Given the node selection decision $\boldsymbol{\beta}$, the accuracy of the objective function becomes fixed values. The original problem can be transformed as

$$\begin{aligned} \mathcal{P}2: \min_{\mathbf{f},\mathbf{p}} \ & \lambda T_i^{\text{total}}, \\ \text{s.t.} \quad & C_2, C_3, C_4, \end{aligned} \tag{22}$$

Since T_i^{total} contains a max term, we define $\eta_{k,i}$ as a binary indicator parameter to simplify the solution process. When $\eta_{k',i} = 1$, $T_{k',i}$ reaches its maximum value, satisfying the supplementary condition $C_5 : T_{k',i} \geq T_{k,i}$ for $k, k' \in \mathcal{K}$. Conversely, when $\eta_{k,i} = 0$, $T_{k,i}$ is not maximized but still satisfies condition C_5. The equation (12) can be rewritten as follows:

$$T_i^{total} = \sum_{k \in \mathcal{K}} \eta_{k,i} T_{k,i} + T_i^{es}. \tag{23}$$

Given the convex nature of problem $\mathcal{P}2$ and the non-coupling between $\mathbf{f}$ and $\mathbf{p}$, we use the **Lagrange multiplier method** to address this issue, constructing the Lagrange function with non-negative multipliers $\boldsymbol{\chi} = \{\chi_{k,i} | k \in \mathcal{K}, i \in \mathcal{I}\}$, $\boldsymbol{\varpi} = \{\varpi_i^{es} | i \in \mathcal{I}\}$, and $\boldsymbol{\gamma} = \{\gamma_{k,i} | k \in \mathcal{K}, i \in \mathcal{I}\}$:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\chi}, \boldsymbol{\varpi}, \boldsymbol{\gamma}, \mathbf{f}, \mathbf{p}) = \lambda T_i^{total} + \sum_{k \in \mathcal{K}} \chi_{k,i} \left(f_{k,i} - f_{\max}^{loc} \right) \\ + \varpi_i^{es} \left(f_i^{es} - f_{\max}^{es} \right) + \sum_{k \in \mathcal{K}} \gamma_{k,i} \left(p_{k,i}^{trans} - p_{\max}^{trans} \right). \end{aligned} \tag{24}$$

Specifically, we calculate the partial derivatives for ϖ_i^{es} of equation (24). The optimal computational capability allocation for the ES can be expressed as

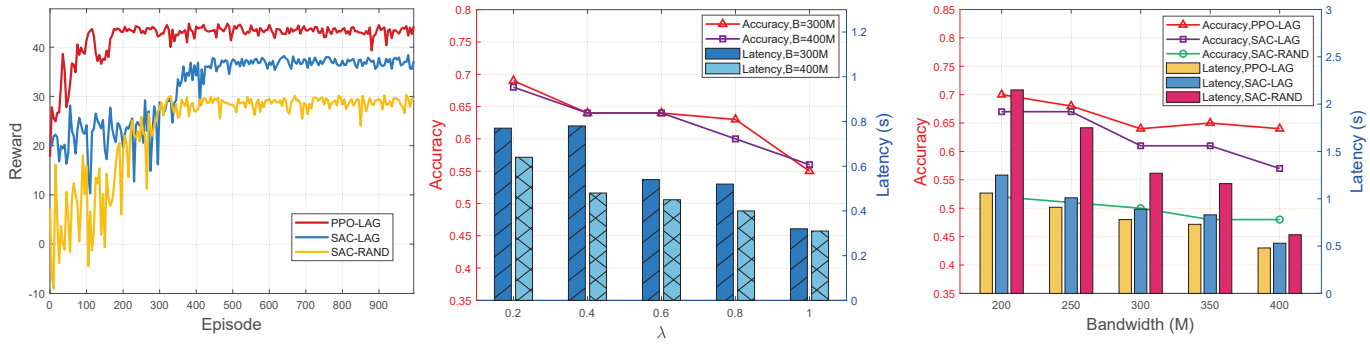
$$f_i^{es*} = \sqrt{\frac{\lambda \rho_{es} \sum_{q=Q_{fe}+1}^{Q} c_q}{\varpi_i^{es}}}. \tag{25}$$

With varying $\eta_{k,i}$ across selected nodes, the allocation of computational capacity and transmission power will follow different approaches:

i) The Optimal Computational Capacity Allocation Strategy: When $\eta_{k,i} = 0$, $\frac{\partial \mathcal{L}(\boldsymbol{\chi}, \boldsymbol{\varpi}, \boldsymbol{\gamma}, \mathbf{f}, \mathbf{p})}{\partial f_{k,i}}$ is a constant. In cases where $f_{k,i}^*$ satisfies C_2 and C_5 but the solution is not unique, we minimize the computational capability of MD k to ensure $f_{k,i}^*$ takes the lowest possible value. When $\eta_{k',i} = 1$, the optimal computational capability for MD k' is given by

$$f_{k',i}^* = \sqrt{\frac{\lambda \eta_{k',i} \beta_{k',i} \rho_{k'} \sum_{q=1}^{Q_{fe}} c_{k',q}}{\chi_{k',i}}}. \tag{26}$$

ii) The Optimal Transmission Power Allocation Strategy: Similarly, when $\eta_{k,i} = 0$, $\frac{\partial \mathcal{L}(\boldsymbol{\chi}, \boldsymbol{\varpi}, \boldsymbol{\gamma}, \mathbf{f}, \mathbf{p})}{\partial p_{k,i}^{trans}}$ remains constant. To minimize the transmission power of MD k, $p_{k,i}^{trans*}$ should be set to the lowest possible value while satisfying the constraints C_4 and C_5. However, when $\eta_{k',i} = 1$, finding an optimal solution for $p_{k',i,t}^{trans}$ becomes challenging. To address this, we transform $\frac{\partial \mathcal{L}(\boldsymbol{\chi}, \boldsymbol{\varpi}, \boldsymbol{\gamma}, \mathbf{f}, \mathbf{p})}{\partial p_{k',i}^{trans}}$ into a convex surrogate function and apply an iterative method for optimization. Specifically, the



(a) Convergence performance of different schemes.

(b) Impact of the weight parameter λ.

(c) Impact of the bandwidth B.

Fig. 5. Performance comparison.

logarithmic term containing $p_{k',i}^{trans}$ is replaced with $p_{k',i,t-1}^{trans}$, while the constant term uses $p_{k',i,t}^{trans}$. Thus, the optimal transmission power allocation can be expressed as

$$p_{k',i,t}^{trans*} = \frac{\lambda \eta_{k',i} \ln(2) \beta_{k',i} b_{k',i}^{trans} B}{\gamma_{k',i} \left[r_{k,i}(p_{k',i,t-1}^{trans}) \right]^2} - \frac{\sum_{l=1, l \neq k}^{K} p_{l,i}^{trans} \Theta + \sigma^2}{||H_{k',i} W_{k',i}||^2}, \tag{27}$$

where $\Theta = ||H_{k',i} W_{l,i}||^2$ and $p_{l,i}^{trans}$ is equivalent to $p_{k,i}^{trans*}$. Considering that $p_{k',i,t}^{trans*}$ depends on $p_{l,i}^{trans}$, while both $f_{k,i}^*$ and $p_{k,i}^{trans*}$ together need to satisfy C_5, we alternately optimize the these variables. Given that different k' correspond to different T_i^{total}, we select the set of variable values associated with the smallest T_i^{total} as the final optimal solution. Moreover, the Lagrange multiplier factors are updated as follows:

$$\chi_{k,i}(m) = \chi_{k,i}(m) + \delta_\chi \left(f_{k,i} - f_{\max}^{loc} \right), \tag{28}$$

$$\varpi_i^{es}(m) = \varpi_i^{es}(m) + \delta_\varpi \left(f_i^{es} - f_{\max}^{es} \right), \tag{29}$$

$$\gamma_{k,i}(m) = \gamma_{k,i}(m) + \delta_\gamma \left(p_{k,i}^{trans} - p_{\max}^{trans} \right), \tag{30}$$

where m is the iteration index, δ_χ, δ_ϖ, and δ_γ respectively correspond to the step sizes.

V. Simulation and Discussion

A. Experiment Setup

In this section, we evaluate the performance of our proposed scheme by implementing MVDet on the Wildtrack dataset for multi-view pedestrian detection [12]. All codes are implemented in Python 3.8 and run on a Linux server equipped with one NVIDIA GeForce RTX 4090 GPU. This multi-view edge video analytics system consists of one ES and seven MDs. Additionally, MDs are divided into two groups, each sharing a 400MHz channel. The transmission signals from MDs use 256QAM modulation, which increases the system's frequency efficiency by approximately eight times. For the PPO-LAG method, we set the episode number Z to 1000 and the number of steps per episode I to 40. The mini-batch size $|\mathcal{B}_{\max}|$ is set to 20, the number of epochs N is set to 10. Moreover, we set the discount factor is 0.98, the GAE parameter is 0.95, and the clipping parameter ϵ is set to 0.1. Other simulation parameters are summarized in Table I.

TABLE I: Simulation Parameters

Parameter	Values
Noise power σ^2	-80 dBm
Transmission Power $p_{k,i}^{trans}, k \in \mathcal{K}$	[0, 1] W
CPU frequency of ES f_i^{es}	[200, 300] GHz
CPU frequency of MDs $f_{k,i}, k \in \mathcal{K}$	[30, 80] GHz
Step size δ_χ, δ_ϖ, δ_γ	10^{-15}, 10^{-15}, 10^{-3}

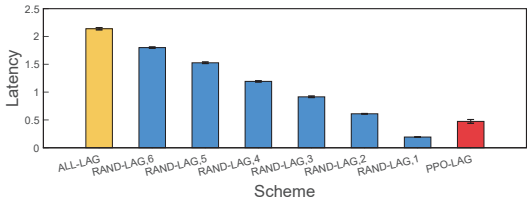
To demonstrate the effectiveness of our proposed scheme, we compare it with four baseline schemes: 1) Use the Soft Actor and Critic (SAC) algorithm for node selection (SAC-LAG). 2) Use SAC algorithm and implement random resource allocation (SAC-RAND). 3) Select node randomly (RAND-LAG). 4) Select all nodes simultaneously (ALL-LAG). To align with our proposed scheme, Baselines 1, 3, and 4 also use the Lagrange multiplier method for resource allocation.

B. Experiment Result

Fig. 5(a) shows the convergence performance of our proposed scheme compared to the SAC-LAG and SAC-RAND schemes. Our scheme converges faster and achieves higher rewards by simultaneously considering learning efficiency and stability. Additionally, the Lagrange multiplier method enables efficient allocation of computational and transmission resources, further boosting rewards.

In Fig. 6, we analyze the performance of different schemes, where the RAND-LAG scheme generates multiple results by fixing a randomly selected number of nodes. Compared to the RAND-LAG and ALL-LAG schemes, our scheme significantly reduces the total end-to-end latency while maintaining high accuracy. This is because our scheme effectively selects nodes with higher information, thereby significantly saving computational and transmission resources. Additionally, we can refer prior frame detection results to guide the processing of subsequent frames, further enhancing overall system efficiency.

(a) Accuracy Comparison.



(b) Latency Comparison.

Fig. 6. Comparison of node selection across different schemes.

Then, we evaluate the weight parameter λ on the performance of our proposed scheme. In Fig. 5(b), we fix all other parameters and vary λ from 0.2 to 1. It is evident that λ plays a trade-off role between accuracy and latency. As λ increases, both inference accuracy and total latency decrease for bandwidths of 300M and 400M. This is because the increase in λ places greater emphasis on latency in the optimization of the objective function, leading the agent to prioritize actions that reduce latency. Conversely, when λ is small, the agent is more likely to select more MDs to achieve higher inference accuracy.

Fig. 5(c) further demonstrates the impact of bandwidth on inference accuracy and total latency. Our proposed scheme outperforms both the SAC-LAG and SAC-RAND schemes, achieving lower total latency and higher inference accuracy. Moreover, across all three schemes,the total latency decreases significantly with increasing bandwidth, while the inference accuracy shows a slight decline. The reason is that higher bandwidth makes the agent focus more on reducing the total latency, which in turn reduces the number of selected MDs.

VI. Conclusion

In this paper, we present a spatial-temporal correlation based node selection scheme for multi-view edge video analytics designed to improve computational efficiency and system performance. By exploiting the spatial redundancy in video frames, our scheme adaptively selects certain MDs and utilizes the detection results from prior frames to guide the processing of subsequent frames. Furthermore, we formulate an optimization problem of joint node selection and resource allocation, which is addressed with the proposed two-stage PPO-LAG algorithm. Experimental results demonstrate the superiority of our scheme.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (No. 2021YFB3300100), and the National Natural Science Foundation of China (No. 62171062) and (U24A20234).

References

[1] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13849–13875, 2021.

[2] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE communications surveys & tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[3] W. Zhang, B. Han, and P. Hui, "On the networking challenges of mobile augmented reality," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, pp. 24–29, 2017.

[4] M. A. Arefeen, S. T. Nimi, and M. Y. S. Uddin, "Framehopper: Selective processing of video frames in detection-driven real-time video analytics," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 125–132, IEEE, 2022.

[5] W. Xiao, Y. Hao, J. Liang, L. Hu, S. A. Alqahtani, and M. Chen, "Adaptive compression offloading and resource allocation for edge vision computing," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[6] Z. Wang, X. He, Z. Zhang, Y. Zhang, Z. Cao, W. Cheng, W. Wang, and Y. Cui, "Edge-assisted real-time video analytics with spatial–temporal redundancy suppression," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6324–6335, 2022.

[7] H. Guo, B. Tian, Z. Yang, B. Chen, Q. Zhou, S. Liu, K. Nahrstedt, and C. Danilov, "Deepstream: bandwidth efficient multi-camera video streaming for deep learning analytics," *arXiv preprint arXiv:2306.15129*, 2023.

[8] Z. Liu, M. Wang, F. Chen, and Q. Lu, "Edge-assisted intelligent video compression for live aerial streaming," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1613–1623, 2022.

[9] X. Dai, P. Yang, X. Zhang, Z. Dai, and L. Yu, "Respire: Reducing spatial–temporal redundancy for efficient edge-based industrial video analytics," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9324–9334, 2022.

[10] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 319–336, 2008.

[11] S. Wang, S. Bi, and Y.-J. A. Zhang, "Edge video analytics with adaptive information gathering: a deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 22, no. 9, pp. 5800–5813, 2023.

[12] Y. Hou, L. Zheng, and S. Gould, "Multiview detection with feature perspective transformation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pp. 1–18, Springer, 2020.

[13] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret, "Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5030–5039, 2018.

[14] J. Heydari, V. Ganapathy, and M. Shah, "Dynamic task offloading in multi-agent mobile edge computing networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.