Prioritized Information Bottleneck Theoretic Framework with Distributed Online Learning for Edge Video Analytics

Zhengru Fang, Senkang Hu, *Student Member, IEEE,* Jingjing Wang, *Senior Member, IEEE,*
Yiqin Deng, Xianhao Chen, *Member, IEEE* and Yuguang Fang, *Fellow, IEEE, ACM*

Abstract—Collaborative perception systems leverage multiple edge devices, such surveillance cameras or autonomous cars, to enhance sensing quality and eliminate blind spots. Despite their advantages, challenges such as limited channel capacity and data redundancy impede their effectiveness. To address these issues, we introduce the Prioritized Information Bottleneck (PIB) framework for edge video analytics. This framework prioritizes the shared data based on the signal-to-noise ratio (SNR) and camera coverage of the region of interest (RoI), reducing spatial-temporal data redundancy to transmit only essential information. This strategy avoids the need for video reconstruction at edge servers and maintains low latency. It leverages a deterministic information bottleneck method to extract compact, relevant features, balancing informativeness and communication costs. For high-dimensional data, we apply variational approximations for practical optimization. To reduce communication costs in fluctuating connections, we propose a gate mechanism based on distributed online learning (DOL) to filter out less informative messages and efficiently select edge servers. Moreover, we establish the asymptotic optimality of DOL by proving the sublinearity of their regrets. To validate the effectiveness of the PIB framework, we conduct real-world experiments on three types of edge devices with varied computing capabilities. Compared to five coding methods for image and video compression, PIB improves mean object detection accuracy (MODA) while reducing 17.8% and reduces communication costs by 82.65% under poor channel conditions.

Index Terms—Collaborative edge inference, information bottleneck, distributed online learning, variational approximations.

I. INTRODUCTION

A. Background

VIDEO analytics is rapidly transforming various sectors such as urban planning, retail analysis, and autonomous navigation by converting visual data streams into useful insights [2]. A large number of video cameras produce vast amounts of video data continuously and often require real-time video stream [3]. Numerous developing applications such as remote patient care [4], video games [5], UAV sensing [6], [7] and virtual and augmented reality depend on the efficient analysis of video data with minimal delay [8].

The increasing number of smart devices requires a computational paradigm shift towards edge computing. This approach involves processing data closer to its source, resulting in several benefits compared to traditional cloud-based paradigms, particularly reduced latency and bandwidth costs. Even a delay as short as second can lead to disastrous consequences. For example, interactive applications such as online gaming and video conferencing require latencies below 100 ms to ensure real-time feedback and seamless user experience [9]. Similarly, VR/AR applications demand extremely low latencies, often less than 20 ms, to prevent motion sickness and maintain a high-quality immersive experience [10]. Utilizing remote cloud services for data processing can result in significant latency increases, often exceeding 100 ms [11]. Moreover, the importance of privacy, particularly in regions with strict data protection laws such as the General Data Protection Regulation (GDPR), makes edge computing even more attractive [12]. According to the Ponemon Institute, 60% of companies express apprehension toward cloud security and decide to manage their own data onsites in order to mitigate potential risks [13].

A key aspect of video analytics, particularly for Bird's Eye View (BEV) applications, is accurately capturing pedestrian occupancy across multiple camera views [14], [15]. BEV representations rely on precise spatial context to depict ground-level scenes in order to minimize occlusions, blind spots, and viewpoint discrepancies. Pedestrian occupancy data enables reliable identification and localization of individuals within a shared view, refining collaborative perception and enhancing tasks such as detection and prediction in complex, dynamic environments. However, the integration of edge devices into video analytics also brings in many significant challenges [16]. The computational demands of deep neural network (DNN) models, such as GoogLeNet [17], which requires about 1.5 billion operations per image classification, place a substantial burden on the limited processing capacities of edge devices [18]. Additionally, the outputs from high-resolution cameras increase the communication load. For example, a 4K video stream requires up to 18 Gbps of bandwidth to transmit

raw video data, potentially overwhelming wireless networks [19]. Therefore, we need to explore efficient video coding for compressing streamed videos. As shown in Fig. 1(a), the traditional compression is to reconstruct streaming frame through efficient entropy coding and motion prediction. However, there are still extensive less informative data being processed, wasting communication bandwidth. For instance, if the tasks involve human recognition or positioning, the reconstructed background of each frame might not be useful for the application.
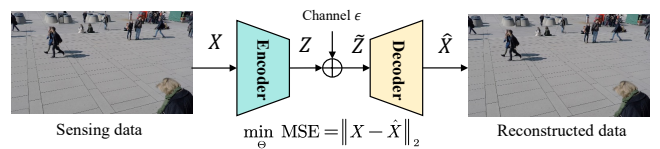
As shown in Fig. 1(b), the information bottleneck (IB) framework is a feasible choice for task-oriented video compression, enabling a trade-off between communication cost and prediction accuracy for specific tasks. However, the current communication strategies for integrating edge devices into video analytics are not effective enough. One major issue is how to handle the computational complexity and transmission of redundant data generated from the overlapping fields of view (FOVs) from multiple cameras [20]. In scenarios with dense camera deployments, up to 60% of data can be redundant due to overlapping FOVs, which unnecessarily overburdens the network [21]. In addition, these strategies often lack adaptability in transmitting tailored data features based on Region of Interest (RoI) and signal-to-noise ratio (SNR), resulting in poor video fusion or alignment. These limitations can negatively impact collaborative perception, even making it less effective than single-camera setups [14].

In this paper, we aim to refine multi-camera video analytics by developing a strategy to prioritize wireless video transmissions. Our proposed Prioritized Information Bottleneck (PIB) strategy attempts to effectively leverage SNR and RoI to selectively transmit data features, significantly reducing computational load and data transmissions. Our method can decrease data transmissions by up to 82.65%, while simultaneously enhancing the mean object detection accuracy (MODA) compared to current state-of-the-art techniques. This approach not only compresses data but also intelligently selects data for processing to ensure only relevant information is transmitted, thus mitigating noise-induced inaccuracies in collaborative sensing scenarios. This innovation sets a new benchmark for efficient and accurate video analytics at the edge.
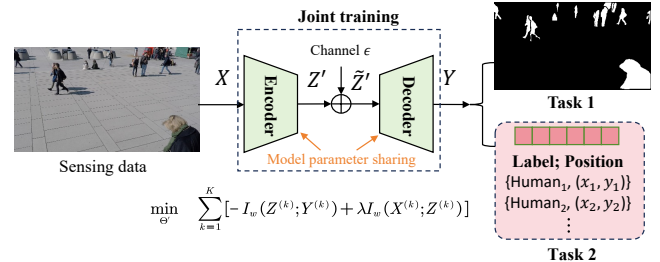
B. State-of-the-Art

This subsection reviews advancements in edge video analytics, with an emphasis on the designs on communication-computing latency reduction. We explore the information bottleneck method to enhance task-oriented performance by minimizing data redundancy. Additionally, we investigate online learning for dynamic ROI management and perceptual quality.

1) Edge Video Analytics: Live video analytics is crucial in various domains such as autonomous driving [14], [15], [22]–[25], mixed reality [26], [27], 3D point cloud analytics [28], and traffic control [29]. These applications, including object recognition [30], are typically equipped with sophisticated machine learning models like Convolutional Neural Networks (CNNs)and Graph Neural Networks (GNNs) [31],

(a) Traditional compression method with redundant data

(b) Information bottleneck method for task-specific compression

Fig. 1: Comparison of compression methods: (a) Traditional compression method with redundant data, (b) Information bottleneck method for task-specific compression.

[32]. However, offloading these applications to central clouds can result in unpredictable transmission delays in wide area networks, particularly when streaming high-quality videos [33]. Therefore, researchers utilize edge computing to serve as a promising alternative to reduce service latency and energy consumption. Li *et al.* propose a novel approach called *ESMO* to optimize frame scheduling and model caching for edge video analytics [34]. Khani *et al.* introduce *RECL*, a new framework for video analytics that integrates model reuse and online retraining to quickly adapt expert models to specific video scenes, optimizing resource allocation and achieving substantial performance gains over prior methods [35]. Wang *et al.* design an MEC-enabled multi-device video analytics system using a Markov decision process to address real-time ground truth absence and content-varying degradation-accuracy issues that significantly enhances the accuracy-latency tradeoff through adaptive information gathering and efficient bandwidth allocation. However, few works consider how to strike a dynamic balance between channel resources and inference performance in a multi-camera sensing system.

In typical edge video analytics scenarios, the lack of infrastructure and limited bandwidth makes real-time object detection challenging, especially for the multi-view camera sensing for wild animals or criminals in remote areas [36]. To achieve real-time object sensing, it is crucial to reduce redundant information and the bandwidth resource demand. Semantic communication can address this challenge by transmitting only the essential semantic information, thereby compressing data streams and reducing transmission overhead. Zhang *et al.* propose a comprehensive framework to highlight the importance of semantic communication in optimizing information transmission [37]. Shao *et al.* introduce a new conceptualization of semantic communication that characterizes it within joint source-channel coding theory, aiming to minimize the semantic distortion-cost region [38]. Xie *et al.* explore a deep

learning-based semantic communication system with memory, showing how dynamic transmission techniques can enhance transmission reliability and efficiency by masking unessential elements [39]. Zhou *et al.* design and implement a deep learning-based image processing pipeline on the ESP32-CAM, proposing a DRL-based approach for efficient camera configuration adaptation in multi-camera systems [40]. Existing research primarily focuses on rate-distortion (R-D) optimization, adapting the bitstream rate based on channel state information (CSI) to reconstruct raw videos [41]. However, these methods rarely consider the performance of specific downstream tasks, such as mean object detection accuracy (MODA), as a system evaluation metric. Consequently, the transmitted information often contains redundant data.

To address this issue, researchers incorporate the information bottleneck (IB) framework to optimize edge video analytics by focusing on task-specific performance, thereby reducing redundancy [42]. The IB framework helps the cause in selectively transmitting only the most relevant features needed for specific tasks, enhancing efficiency. Pensia *et al.* propose a novel feature extraction strategy in supervised learning that enhances classifier robustness to small input perturbations by incorporating a Fisher information penalty into the information bottleneck framework [43]. Wang *et al.* present a deep multi-view subspace clustering framework to extend the information bottleneck principle to a self-supervised setting, leading to superior performance in multi-view subspace clustering on real-world datasets [44]. IB tradeoff is well-suited for bandwidth-limited edge inference and is a key design principle in our study for efficient communication. Wang *et al.* introduce the Informative Multi-Agent Communication (IMAC) method, which uses the information bottleneck principle to develop efficient communication protocols and scheduling for multi-agent reinforcement learning under limited bandwidth [45]. Shao *et al.* propose a task-oriented communication scheme for multi-device cooperative edge inference, optimizing local feature extraction and distributed feature encoding to minimize data redundancy and focus on task-relevant information, leveraging the information bottleneck principle and extending it to a distributed deterministic information bottleneck framework. However, these existing studies often neglect the need to prioritize different data from multiple cameras for various downstream tasks, such as considering ROIs. Moreover, most existing studies overlook the correlation between multiple cameras in multi-view scenarios. Shao *et al.* extract compact task-oriented representations based on the IB principle, but they overlook the fact that different tasks require varying levels of priority [16]. By leveraging these correlations and levels of priority, it is possible to further reduce data rates by minimizing duplicate information across different camera feeds and enhance inference performance at the same time.

2) Learning-Based Transmission Scheduling: Online learning in multi-agent deep reinforcement learning (MADRL) enhances multi-camera sensing under dynamic channels and overlapped ROIs. Effective transmission scheduling determines when and which agents communicate through binary vectors indicating allowed communications at specific time steps, forming a communication graph. Central transmission scheduling schemes use a globally shared policy to control communication. Kim *et al.* propose SchedNet, which uses a global scheduler to limit broadcasting agents and reduce communication overhead [46]. Du *et al.* introduce FlowComm, forming a directed graph for communication [47]. Liu *et al.* develop GA-Comm, using a two-stage attention network (G2ANet) to manage agent interactions [48]. Niu *et al.* present MAGIC, a framework using a directed communication graph for enhanced coordination [49]. In distributed transmission scheduling schemes, each agent individually determines whether to communicate, forming a graph structure. Liu *et al.* propose a framework for multi-agent collaborative perception, addressing communication group construction and decision-making for efficient bandwidth use, significantly reducing communication while maintaining performance [50]. Deep learning optimizes these systems by refining communication actions and schedules, transmitting only relevant information, and minimizing redundancy. However, existing multi-camera cooperative sensing algorithms do not effectively address the transmission scheduling problem, particularly under dynamic wireless channels and overlapped ROIs.

C. Our Contributions

Edge computing plays a crucial role in collaborative perception systems, improving tracking precision and minimizing blind spots through multi-view sensing. However, challenges such as limited channel capacity and data redundancy impede their effectiveness. To address these issues, we propose the Prioritized Information Bottleneck (**PIB**) framework for edge video analytics. Compared with the conference version [1], this paper improves the MODA by up to 17.88% and reduces the communication cost by 23.94%. Our contributions are summarized as follows:

- We propose the PIB framework that prioritizes the share data based on the signal-to-noise ratio (SNR) and camera coverage of the region of interest (RoI), reducing redundancy both spatially and temporally. This approach avoids the need for video reconstruction at edge servers and maintains low latency.
- Our framework leverages a deterministic information bottleneck method to extract compact, relevant features, balancing informativeness and communication costs. For high-dimensional data, we apply variational approximations for practical optimization.
- To reduce communication costs in fluctuating links, we introduce a gate mechanism based on distributed online learning (DOL) to filter out unprofitable messages and efficiently select edge servers. We establish the asymptotic optimality of DOL by showing the sublinearity of their regrets.
- Our extensive experimental evaluations across different real-world hardware platforms demonstrate that PIB significantly enhances mean object detection accuracy (MODA) and reduces communication costs. Compared to TOCOM-TEM, JPEG, H.264, H.265, and AV1, PIB improves MODA by 17.8% while reducing communication costs by 82.65% under poor channel conditions.
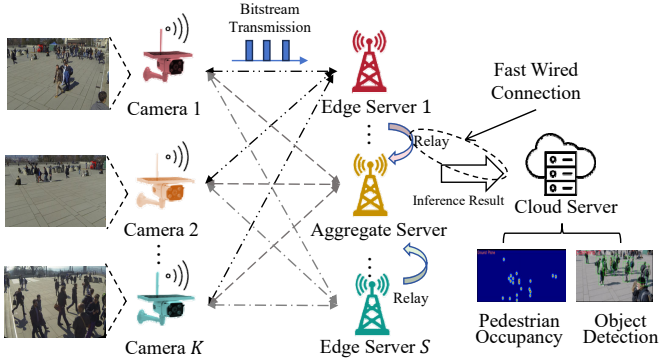
Fig. 2: System model.

TABLE I: Key Notations

Notation	Description
$C_{k,s}$	Capacity of the link between camera k and edge server s
$d_{k,s}^T$	Transmission delay from camera k to edge server s
d_{k,s_0}^I	Inference delay at the aggregate edge server s_0
d_{r,s_0}^R	Relay delay from edge server r to aggregate server s_0
p_k, w_k	Priority score and weight for camera k
$X^{(k)}$	Input data from camera k
$Z^{(k)}$	Feature extracted from camera k
Y	Random variable of the output
$\mathcal{E}_{k,s}^{c \to e}$	Connection between camera k and edge server s
$\mathcal{E}_s^{e \to e_0}$	Connection between edge server s and aggregate server s_0
$\mathcal{K}$	Set of cameras selected at time t
Ψ_s^F	Computational cost at edge server s
Ψ_s^T	Computational cost of forwarding features
Ψ_k^R	Remaining computing capacity of edge server k
$\mathcal{T}_{k,s}^{c \to e}$	Latency from camera k to edge server s
$\mathcal{T}_s^{e \to e}$	Latency between edge server s and aggregate server s_0

Additionally, our method can reduce the standard deviation of streaming packet sizes by up to 9.43%, while simultaneously maintaining higher MODA, ensuring better transmission robustness under poor channel conditions.

The remainder of this paper is organized as follows: Sec. II introduces the system model. Sec. III covers the problem formulation, including prioritized information bottleneck analysis and the CMAB problem. Sec. IV describes the methodology, focusing on the derivation of the IB problem's upper bound, loss function design, and the distributed gate mechanism. Sec. V evaluates the performance of the PIB framework through simulations that forecast pedestrian occupancy in urban settings, considering communication bottlenecks, camera delays, and edge server connectivity.

II. SYSTEM MODEL

As illustrated in Fig. 2, our system comprises a set of edge cameras, denoted as $\mathcal{K} = \{1, 2, \ldots, K\}$, and a set of edge servers, denoted as $\mathcal{S} = \{1, 2, \ldots, S\}$. Each camera has a specific Field of View (FoV), FoV_k, covering a subset of the monitored area. Our goal is to enable collaborative perception for pedestrian occupancy prediction under constrained channel capacity. The backbone of our system employs intermediate collaboration, where only encoded feature representations Z are transmitted to a central aggregate server for decoding and inference. Other edge servers serve solely as relay nodes, as they do not perform decoding or final inference due to the network structure while does not support task offloading for these operations. The aggregate server is chosen to minimize overall system delay, while the relay-only role of other edge servers ensures streamlined data flow. The key notations are given in Table I for the ease of reading.

A. Communication Model

We use Frequency Division Multiple Access (FDMA) to manage communication among cameras, defining the capacity $C_{k,s}$ for each camera k and edge server s combination using the SNR-based Shannon capacity:

$$C_{k,s} = B_{k,s} \log_2 \left(1 + \text{SNR}_{k,s} \right), \tag{1}$$

where $B_{k,s}$ is the bandwidth allocated to the link between camera k and edge server s, and $\text{SNR}_{k,s}$ is the signal-to-noise ratio of this link. The transmission delay $d_{k,s}^T$ is given by:

$$d_{k,s}^T = \frac{D}{C_{k,s}}, \tag{2}$$

where D is the data packet size. Each camera k decides whether to transmit data directly to its aggregate server or via a relay edge server based on channel quality: 1) If the channel quality is good, the camera transmits directly to the aggregate server s_0. The total delay is $d_{k,s}^{\text{total}} = d_{k,s_0}^T + d_{k,s_0}^I$, where d_{k,s_0}^I is the inference delay at the aggregate server. 2) If the channel quality is poor, the camera first transmits to a relay edge server r, which then forwards the data to the aggregate server. The total delay in this case is $d_{k,s}^{\text{total}} = d_{k,r}^T + d_{r,s_0}^R + d_{k,s_0}^I$, where d_{r,s_0}^R is the relay delay. By dynamically choosing between relay and direct transmission, the system adapts to varying channel conditions, ensuring minimal delays and efficient use of network resources.

B. Priority Weight Formulation

Dynamic priority weighting is essential for optimizing network resource allocation, as various data sources require different levels of attention. Inspired by our previous work [14], we employ a dual-layer Multilayer Perceptron (MLP)[1]

---

[1] The MLP is trained in a supervised learning manner, where the input features are the normalized delay $d_{\text{norm},k}$ and the normalized number of perceived moving objects $\chi_{\text{norm},k}$. The target output is the optimal priority weight W_{target}. The loss function used for training is designed to minimize the discrepancy between the computed weights w_k and the target weights W_{target}, as described in Sec. IV-D.

to compute priority weights based on normalized delay and the number of perceived objects (χ_k).

$$p_k = \text{MLP}(d_{\text{norm},k}, \chi_{\text{norm},k}; \Theta_M), \qquad (3)$$

where p_k denotes the computed priority score for camera k, and Θ_M represents the trainable parameters of MLP. The architecture of this MLP, featuring two layers, allows it to effectively model the interactions between delay and the number of perceived moving objects. Specifically, $d_{\text{norm},k} = \frac{d_k}{d_{\text{max}}}$ and $\chi_{\text{norm},k} = \frac{\chi_k - \chi_L}{\chi_U - \chi_L}$. To account for the dynamic nature of the system, we periodically update d_{max}. Specifically, we define d_{max} as $d_{\text{max}} = \max_k(d_k) + \Delta$, where d_k is the delay for camera k and Δ is a predefined threshold. This dynamic computation ensures that the normalized delay $d_{\text{norm},k} = \frac{d_k}{d_{\text{max}}}$ remains within a reasonable range, preventing resource overcommitment. χ_k represents the number of moving objects perceived by camera k, while χ_U and χ_L denote the upper and lower bounds of the number of moving objects that any edge camera should perceive, respectively.

To transform the raw priority scores into a usable format within the system, we apply a softmax function, which normalizes these scores into a set of weights summed to one:

$$w_k = \frac{e^{p_k}}{\sum_{j=1}^{K} e^{p_j}}, \qquad (4)$$

where w_k signifies the priority weight for camera k. This method ensures that cameras which are more critical, either due to high coverage or due to lower delays, are given priority, thereby enhancing the decision-making capabilities and responsiveness of the edge analytics system.

III. PROBLEM FORMULATION

In this section, we establish the theoretical foundation for our PIB framework. We begin by detailing the IB analysis to determine the optimal balance between data compression and relevant information retention. Following this, we formulate the combinatorial multi-armed band (CMAB) problem to model the decision-making process of cameras in a distributed environment.

A. Prioritized Information Bottleneck Analysis

In the context of information theory, the IB method seeks an optimal trade-off between the compression of an input variable X and the preservation of relevant information about an output variable Y [51]. Throughout this paper, upper-case letters (e.g., X, Y, and Z) represent random variables, while lower-case letters (e.g., x, y, and z) denote their realizations. We formalize the input data from camera k as $X^{(k)}$, and the target prediction as Y, corresponding to the population in the dataset $\mathcal{D}$. The goal is to encode $X^{(k)}$ into a meaningful and concise representation $Z^{(k)}$, which aligns with the hidden representation $z^{(k)}$ that captures task-relevant features of multi-view content for prediction tasks. The classical IB problem can be formulated as a constrained optimization task:

$$\max_{\Theta} \quad \sum_{k=1}^{K} I\left(Z^{(k)}; Y\right)$$
$$\text{s.t.} \quad I\left(X^{(k)}; Z^{(k)}\right) \leq I_c, \quad (k = 1, 2, \cdots, K), \qquad (5)$$

where $I(Z^{(k)}, Y)$ denotes the mutual information between two random variables $Z^{(k)}$ and Y. Θ represents the set of all learnable parameters in the PIB framework, including Θ_M and the variational approximation in the following section. The mutual information is essentially a measure of the amount of information obtained about one random variable through the other random variable. I_c is the maximum permissible mutual information that $Z^{(k)}$ can contain about $X^{(k)}$. The objective is to ensure that $Z^{(k)}$ captures the most relevant information about $X^{(k)}$ for predicting Y while remaining as concise as possible. By introducing a Lagrange multiplier[2] λ, the problem is equivalently expressed as:

$$\max_{\Theta} \quad R_{IB} = \sum_{k=1}^{K} \left[I\left(Z^{(k)}; Y\right) - \lambda I\left(X^{(k)}; Z^{(k)}\right) \right], \qquad (6)$$

where R_{IB} represents the IB functional, balancing the compression of $X^{(k)}$ against the necessity of accurately predicting Y. Next, we extend the IB framework to a multi-camera setting by introducing priority weights to the mutual information terms, adapting the optimization problem as follows:

$$\min_{\Theta} \quad \sum_{k=1}^{K} \left[-I_w\left(Z^{(k)}; Y\right) + \lambda I_w\left(X^{(k)}; Z^{(k)}\right) \right], \qquad (7)$$

where the weighted mutual information terms are defined as follows:

$$\begin{cases} I_w\left(Z^{(k)}; Y\right) = w_k \cdot I\left(Z^{(k)}; Y\right), \\ I_w\left(X^{(k)}; Z^{(k)}\right) = e^{w^0 - w_k} \cdot I\left(X^{(k)}; Z^{(k)}\right), \end{cases} \qquad (8)$$

where the non-negative value w^0 represents the maximum allowable weight for w_k. The first term with linear weights $I_w\left(Z^{(k)}; Y\right) = w_k I\left(Z^{(k)}; Y\right)$ is the weighted mutual information between the compressed representation $Z^{(k)}$ from camera k and the target Y. This term can also be used to capture the semantic compression in raw data. The linear weighting with w_k ensures the influence of each camera is proportional to its priority weight. Higher w_k values increase the weight given to $I\left(Z^{(k)}; Y\right)$ in the objective function, emphasizing cameras that provide high-quality data for accurate target prediction.

The second term with negative exponential weights $I_w\left(X^{(k)}; Z^{(k)}\right) = e^{(w^0 - w_k)} \cdot I\left(X^{(k)}; Z^{(k)}\right)$ denotes the mutual information between the original data $X^{(k)}$ and its compressed form $Z^{(k)}$, scaled by a negative exponential function of w_k. This ensures an exponential decay in the influence of $I\left(X^{(k)}; Z^{(k)}\right)$ as w_k increases. Cameras with lower priority weights (lower w_k) undergo more aggressive data compression (as $e^{w^0 - w_k}$ is greater for smaller w_k values), optimizing bandwidth and storage usage without significantly affecting overall performance. In this paper, we use this type of weighting for the proof of concept study and will investigate more general weighting in the future.

---

[2]All Lagrange multipliers λ are the same, and we only use trainable weight parameters to dynamically balance between accuracy and communication bottleneck.

B. Combinatorial Multiarmed Bandit (CMAB) Problem

In dynamic environments with varying channel states and regions of interest (ROIs), ensuring high inference accuracy is challenging. The system must adaptively determine whether each camera should transmit its features and decide which edge server to use for transmission. Moreover, due to the edge server's limited bandwidth and computing capacity, each camera must decide if it should transmit directly to an edge server or use another edge server as a relay node before data fusion at the final edge server.

Accordingly, the problem can be formulated as a combinatorial multi-armed bandit (CMAB) problem. Each camera's connection establishment and edge server's connection establishment are base arms, and the collective actions of all agents constitute a super arm. Let $a_k(t) \in \left\{ \mathcal{E}_{k,s}^{c \to e}, \mathcal{E}_s^{e \to e_0} \right\}$ represent the action taken by camera k at time t, where $\mathcal{E}_{k,s}^{c \to e}$ denotes the connection between the k-th camera and the s-th edge server, and $\mathcal{E}_s^{e \to e_0}$ denotes the connection between the s-th edge server and the s_0-th edge server for data fusion. The super arm is a subset of arms selected for the decision to transmit (the s-th edge server) and data fusion (the s_0-th edge server)[3]. Dynamic channel state and ROI impact inference accuracy. This metric can be defined using the change in Multiple Object Detection Accuracy (MODA). MODA is calculated as $M = 1 - \frac{FN+FP}{TP+FN}$, where TP denotes the number of correctly detected objects, FN means the number of missed detections, and FP is the number of false detections. Specifically, the gain in MODA from adding the k-th camera's feature to the ego camera's[4] feature can be expressed as:

$$\Delta M_k = M_{C_a \cup \{k\}} - M_{C_a}, \tag{9}$$

where C_a represents the set of cameras already selected, and $M_{C_a \cup \{k\}}$ represents the MODA score when the k-th camera is added to the set $\mathcal{K}$. To incorporate submodularity[5], the reward function needs to reflect the diminishing returns property. Therefore, we define the reward function $r_{\mathcal{K}}(t)$ as:

$$r_a(t) = \sum_{k \in \mathcal{K}} \Delta M_k, \tag{10}$$

where $\Delta M_k = M_{\mathcal{K} \cup \{k\}} - M_{\mathcal{K}}$, and $\mathcal{K}$ is the set of cameras selected at time t. The computational cost of multi-camera fusion and inference at the edge server s is denoted as Ψ_s^F. The computational cost of simply forwarding features from one edge server to another is denoted as Ψ_s^T. The remaining computing capacity of the k-th edge server is Ψ_s^R. $\mathcal{T}_k^{c \to e}$ denotes the latency of the transmission between the k-th camera and the s-th edge server, and $\mathcal{T}_s^{e \to e}$ denotes the latency of the transmission between the k-th edge server and the s-th edge server. Therefore, the CMAB problem can be formulated as:

$$\max_{a_k(t)} \quad \sum_{t=1}^{T} \mathbb{E}[r_a(t)]$$

s.t. $(11a):$ $\quad K_{\min} \leq |\mathcal{K}| \leq K_{\max},$

$\quad\quad (11b):$ $\quad \Psi^F \leq \Psi_{s_0}^R,$

$\quad\quad (11c):$ $\quad \sum_{s \in \mathcal{S}} \mathcal{E}_{k,s}^{c \to e} \cdot \mathcal{E}_{s,s_0}^{e \to e} = 1, \forall k \in \mathcal{K},$

$\quad\quad (11d):$ $\quad 0 \leq \sum_{k \in \mathcal{K}} \mathcal{E}_{k,s}^{c \to e} \leq \mathcal{E}_{\max}^{c \to e}, \forall s \in \mathcal{S},$

$\quad\quad (11e):$ $\quad \mathcal{E}_{k,s}^{c \to e} \mathcal{E}_s^{e \to e_0} \left(\mathcal{T}_{k,s}^{c \to e} + \mathcal{T}_{s,s_0}^{e \to e} \right) \leq \mathcal{T}^U, \forall k \in \mathcal{K},$
$$\tag{11}$$

where (11a) ensures the number of selected cameras falls within the specified range and $|\mathcal{K}| = \sum_{k \in \mathcal{K}} \mathcal{E}_{i,s}^{c \to e}$, (11b) ensures that the remaining computing capacity of the aggregate server (s_0) chosen for inference is no less than the required capacity for fusion, (11c) ensures that each camera uses a unique transmission connection, (11d) ensures that the number of connections established by a single edge server does not exceed the maximum allowable connections, and (11e) ensures that the total latency for any transmission path is within the allowable time limit $\mathcal{T}^U$ for all edge servers $s \in \mathcal{S}$.

Solving the CMAB problem in multi-camera collaborative perception is challenging for traditional optimization methods due to: 1) **Dynamic environment**: Constantly changing channel states and ROIs make real-time adaptation difficult. 2) **Computational complexity**: The problem's combinatorial nature creates a massive solution space. 3) **Decentralized decision**: Independent yet collaborative decisions by multiple cameras and edge servers require a decentralized approach. Therefore, we employ distributed online learning techniques to address the CMAB problem in Sec. IV-E, allowing the system to learn and adapt dynamically, solve efficiently, and make decentralized decisions.

IV. METHODOLOGY

In this section, we first introduce the overview of the proposed encoder/decoder architecture. Then, we propose the variational approximation method to reduce the computational complexity of estimating the mutual information during the minimization of Eq. (7) in Sec. IV-B. In Sec. IV-C, we design a multi-frame correlation model that utilizes variational approximation to capture the temporal correlation in video sequences. In Sec. IV-D, we derive the loss functions for the PIB-based encoder and decoder. Sec. IV-E proposes a gate mechanism based on distributed online learning to address the CMAB problem.

A. Architecture Summary

In this subsection, we outline the workflow of our PIB framework, designed for collaborative edge video analytics. As depicted in Fig. 3, the process starts with each edge camera (denoted by k) capturing raw video data $X_t^{(k)}$ and extracting feature maps. These cameras utilize priority weights w_k to optimize the balance between communication costs and perception accuracy, adapting to varying channel conditions.

---

[3]We omit "(t)" for simplicity in the definition of connection establishment.

[4]The ego edge camera is the reference camera selected for data fusion. It typically has the highest number of detected moving objects (χ_k) to provide the most comprehensive feature set for accurate object detection.

[5]The submodularity of $r_a(t)$ can be proven in Appendix C of [14].
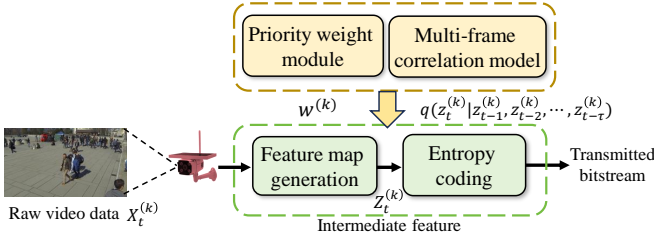
Fig. 3: The procedure of video encoding.

Fig. 4: The procedure of video decoding.

The extracted features are then compressed using entropy coding and sent as a bitstream to the edge server for further processing. On the server (see Fig. 4), the video features are reconstructed using the shared parameters such as weights w_k and the variational model parameters $q(Z_t^{(k)}|Z_{t-1}^{(k)}, ..., Z_{t-\tau}^{(k)})$. The server integrates these multi-view features to estimate Y_t, such as pedestrian occupancy and object detection. This approach leverages historical frame correlations through a multi-frame correlation model to enhance prediction accuracy. The DNN architecture of the PIB framework is detailed in Appendix A.

B. Variational Approximation Method

The objective function of information bottleneck in Eq. (7) can be divided into two parts. The first part is $-\sum_{k=1}^{K} w_k \cdot I\left(Z^{(k)}; Y\right)$, which denotes the quality of video reconstruction by decoding at an edge server. The second part is $\lambda \sum_{k=1}^{K} e^{w^0 - w_k} \cdot I\left(X^{(k)}; Z^{(k)}\right)$, which denotes the compression efficiency for feature extraction. As it has been shown in the way a decoder works, $p\left(Y|Z^{(k)}\right)$ can be any valid type of conditional distributions, but most often it is not feasible enough for straightforward calculation. Because of this complexity, it is highly challenging to directly compute the two mutual information components in Eq. (7).

As for the first part, we adopt the variational approach [52]. This approach suggests that the decoder is part of a simpler group of distributions called Q. We then search for a distribution $q\left(Y|Z^{(k)}; \Theta_d^{(k)}\right)$ within this group that is most similar to the best possible decoder distribution, using the KL-divergence to measure the closeness. $\Theta_d^{(k)}$ is a learnable parameter. Because computing the high-dimensional integrals in the posterior is infeasible, we substitute the optimal inference model with a variational approximation. Thus, we obtain the lower bound of $I_w\left(Z^{(k)}; Y^{(k)}\right) = w_k \cdot I\left(Z^{(k)}; Y^{(k)}\right) \geq w_k\left\{\mathbb{E}_{p(Y,Z)}\left[\log q\left(Y^{(k)}|Z^{(k)}; \Theta_d^{(k)}\right)\right] + H\left(Y^{(k)}\right)\right\}$, as established in Proposition 1.

Proposition 1: *The probabilistic model of decoder $p(Y|Z)$ maps a representation $Z \in \mathbb{Z}$ into task inference $Y \in \mathbb{Y}$. Let $q(Y|Z)$ denote the variational approximation of decoder $p(Y|Z)$. We can obtain*

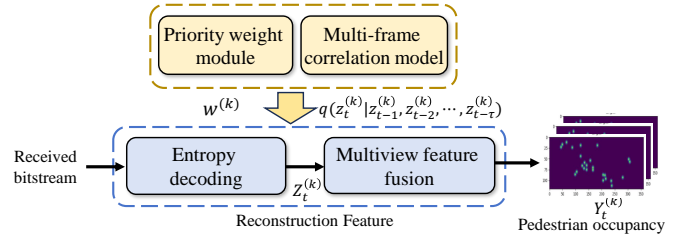$$I(Z;Y) \geq \mathbb{E}_{p(Y,Z)}[\log q(Y|Z)] + H(Y). \quad (12)$$

Proof: We start with the standard definition of mutual information:

$$I(Z;Y) = \mathbb{E}_{p(Y,Z)}\left[\log \frac{p(Y,Z)}{p(Y)p(Z)}\right] = \mathbb{E}_{p(Y,Z)}\left[\log \frac{p(Y|Z)}{p(Y)}\right], \quad (13)$$

which utilizes the relationship $p(Y,Z) = p(Y|Z)p(Z)$ to express the mutual information in terms of the ratio of the conditional probability to the marginal probability of Y.

Introducing the Kullback-Leibler (KL) divergence, which measures how the distribution $q(Y|Z)$ approximates the true distribution $p(Y|Z)$, we have:

$$D_{KL}\left[p(Y|Z) \parallel q(Y|Z)\right] = \mathbb{E}_{p(Y|Z)}\left[\log \frac{p(Y|Z)}{q(Y|Z)}\right] \geq 0, \quad (14)$$

where the KL divergence is always non-negative. This leads to:

$$\mathbb{E}_{p(Y|Z)}\left[\log p(Y|Z)\right] \geq \mathbb{E}_{p(Y|Z)}\left[\log q(Y|Z)\right], \quad (15)$$

which can be simplified to:

$$\mathbb{E}_{p(Y,Z)}\left[\log p(Y|Z)\right] \geq \mathbb{E}_{p(Y,Z)}\left[\log q(Y|Z)\right]. \quad (16)$$

Therefore, we can derive the lower bound for the mutual information as follows:

$$\begin{aligned} I(Z;Y) &= \mathbb{E}_{p(Y,Z)}\left[\log \frac{p(Y|Z)}{p(Y)}\right] \\ &\geq \mathbb{E}_{p(Y,Z)}[\log q(Y|Z)] - \mathbb{E}_{p(Y)}[\log p(Y)] \\ &= \mathbb{E}_{p(Y,Z)}[\log q(Y|Z)] + H(Y), \end{aligned} \quad (17)$$

where $H(Y)$ is the entropy of Y, a constant that reflects the inherent uncertainty in Y independent of Z. ∎

To establish an upper bound for the term $\lambda \sum_{k=1}^{K} e^{w_0 - w_k} \cdot I(X^{(k)}; Z^{(k)})$ in the context of the complexity in directly minimizing it, we proceed as follows. Recognizing that $H(Z^{(k)}|X^{(k)}) \geq 0$ from the properties of entropy, we obtain the inequality:

$$\begin{aligned} \lambda \sum_{k=1}^{K} I_w\left(X^{(k)}; Z^{(k)}\right) &= \lambda \sum_{k=1}^{K}\left[\frac{H\left(Z^{(k)}\right) - H\left(Z^{(k)}|X^{(k)}\right)}{e^{w_k - w_0}}\right] \\ &\leq \lambda \sum_{k=1}^{K} \frac{H\left(Z^{(k)}\right)}{e^{w_k - w_0}} \leq \lambda \sum_{k=1}^{K} \frac{H\left(Z^{(k)}, V^{(k)}\right)}{e^{w_k - w_0}}, \end{aligned} \quad (18)$$

where we use the latent variables $V^{(k)}$ as the side information to encode the quantized feature and we have used $H(Z^{(k)}, V^{(k)}) \geq H(Z^{(k)})$. The joint entropy $H(Z^{(k)}, V^{(k)})$ represents the communication cost, which is minimized when the joint entropy is minimized.

Proposition 2: *The upper bound for the mutual information term in Eq. 7 is given by:*

$$I_w\left(X^{(k)};Z^{(k)}\right) \leq \mathbb{E}_{p(Z^{(k)},V^{(k)})}\left[-\log q\left(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)}\right)\right.$$
$$\left.\times q(V^{(k)};\Theta_l^{(k)})\right] e^{w_0-w_k},$$
(19)

where $q\left(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)}\right)$ is the variational distribution conditioned on the latent variables $V^{(k)}$ with parameters $\Theta_{con}^{(k)}$, and $q(V^{(k)};\Theta_l^{(k)})$ is the marginal variational distribution with parameters $\Theta_l^{(k)}$.

Proof: The proof begins by recognizing that the joint entropy $H(Z^{(k)},V^{(k)})$ represents the communication cost, which is minimized when the joint entropy is minimized. The joint entropy can be expressed as the expectation over the logarithm of the ratio of the true joint distribution $p(Z^{(k)},V^{(k)})$ to the variational distribution $q(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)}) \cdot q(V^{(k)};\Theta_l^{(k)})$, where $\Theta_{con}^{(k)}$ represents the learnable parameter:

$$H(Z^{(k)},V^{(k)}) = \mathbb{E}_{p(Z^{(k)},V^{(k)})}\left[-\log q(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)})\right.$$
$$\left.-\log q(V^{(k)};\Theta_l^{(k)})\right] - D_{KL}(p||q),$$
(20)

where $D_{KL}(p||q)$ is the KL-divergence between the distribution of $p = p(Z^{(k)},V^{(k)})$ and $q = q(Z^{(k)},V^{(k)})$. The KL-divergence is non-negative, thus we have:

$$D_{KL}\left[p\left(Z^{(k)},V^{(k)}\right) \| q\left(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)}\right) q\left(V^{(k)};\Theta_l^{(k)}\right)\right] \geq 0,$$
(21)

Combining the joint entropy equation (20) with inequality (21), we get:

$$H(Z^{(k)},V^{(k)}) \leq \mathbb{E}_{p(Z^{(k)},V^{(k)})}\left[-\log q(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)})\right.$$
$$\left.\times q\left(V^{(k)};\Theta_l^{(k)}\right)\right].$$
(22)

Thus, we can substitute Ineq. (21) into Ineq. (18) to obtain the result in Ineq. (19). ∎

It should be noted that the lower bound in Ineq. (17) and upper bound in Ineq. (19) enables us to establish an upper limit on the objective function in minimization problem in (7). This makes it easier to minimize with the corresponding loss function during network training, as discussed in Sec. IV-D.

C. Multi-Frame Correlation Model

Inspired by the previous work [16], PIB framework utilizes a multi-frame correlation model to leverage variational approximation to capture the temporal dynamics in video sequences. This approach utilizes the temporal redundancy across contiguous frames to model the conditional probability distribution effectively. Our model approximates the next feature in the sequence by considering the variational distribution $q(Z_t^{(k)}|Z_{t-1}^{(k)},...,Z_{t-\tau}^{(k)};\Theta_\tau^{(k)})$, which can be modeled as a Gaussian distribution aimed at mimicking the true conditional distribution of the subsequent frame given the previous frames:

$$q\left(Z_t^{(k)}|Z_{t-1}^{(k)},...,Z_{t-\tau}^{(k)};\Theta_\tau^{(k)}\right) = \mathcal{N}\left(\mu\left(\Theta_\tau^{(k)}\right),\sigma^2\left(\Theta_\tau^{(k)}\right)\right),$$

where μ and σ^2 are parametric functions of the preceding frames, encapsulating the temporal dependencies. These functions are modeled using a deep neural network with parameters $\Theta_\tau^{(k)}$ learned from data. By optimizing the variational parameters, our model aims to closely match the true distribution, thus encoding the features more efficiently.

D. Network Loss Functions Derivation

In this subsection, we formulate our network loss functions to enhance the information transmission in a multi-camera scenario based on the priority-driven mechanism and the IB principle as discussed in Sec. II-B and Sec. III-A.

Given the variability in channel quality and the occurrence of delays, we introduce the first loss function, $\mathcal{L}_1^{(k)}$, designed to minimize the impact of unreliable data sources while maximizing inference accuracy. We also consider to improve the number of perceived moving objects (χ_{norm}). Thus, the loss function of the MLP network in Sec. II-B is:

$$\mathcal{L}_1 = \sum_{k=1}^{K}\left[1_{d_{\text{norm},k}<\epsilon}\frac{(w_k - W_{\text{target}})^2}{\chi_{\text{norm},k}} + 1_{d_{\text{norm},k}>\epsilon}\left(w_k^2\right)\right], \quad (23)$$

where ϵ denotes a permissible delay that cannot result in errors in multi-view fusion, and W_{target} represents the target weight for a camera without excessive delay. The second loss function $\mathcal{L}_2$ aims to minimize the upper bound of the mutual information, following the inequalities derived in (17) and (19). $\mathcal{L}_2$ ensures efficient encoding while preserving essential information for accurate prediction:

$$\mathcal{L}_2 = \sum_{k=1}^{K}\underbrace{\mathbb{E}[-w_k \log q(Y|Z^{(k)};\Theta_d^{(k)})]}_{\text{The upper bound of } -I_w(Z^{(k)};Y)} + \lambda \cdot \min\left\{R_{max},\right.$$
$$\left.\underbrace{\mathbb{E}\left[-\log q(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)}) \cdot q(V^{(k)};\Theta_l^{(k)})\right] e^{(w^0-w_k)}}_{\text{The upper bound of } I_w(X^{(k)};Z^{(k)})}\right\}.$$
(24)

The first term of $\mathcal{L}_2$ ignores $H(Y)$ in Ineq. (12) because it is a constant. R_{max} represents the penalty for the excessive communication cost of the variation approximation $q(Z^{(k)}|V^{(k)};\Theta_{con}^{(k)}) \cdot q(V^{(k)};\Theta_l^{(k)})$, which captures the degradation of training decoder $p(Y|Z^{(k)})$. In Sec. IV-C, the Multi-Frame Correlation Model leverages temporal dynamics, which is critical for sequential data processing in video analytics. The third loss function, $\mathcal{L}_2^{(k)}$, is needed to minimize the KL divergence between the true distribution of frame sequences and the modeled variational distribution:

$$\mathcal{L}_3 = \sum_{k=1}^{K} D_{KL}\left[p(Z_t^{(k)}|Z_{<t}^{(k)})||q(Z_t^{(k)}|Z_{<t}^{(k)};\Theta_\tau^{(k)})\right], \quad (25)$$

where $Z_{<t}^{(k)} = (Z_{t-1}^{(k)},...,Z_{t-\tau}^{(k)})$. These loss functions collectively aim to optimize the trade-off between data transmission costs and perceptual accuracy, crucial for enhancing the performance of edge analytics in multi-camera systems. 1 introduces the detailed procedure of feature extraction and variational approximation.

Algorithm 1: Training Procedures of the Feature Extraction and Variational Approximation

---

Input: Training dataset, initialized parameters $\Theta_d^{(k)}$, $\Theta_e^{(k)}$, $\Theta_{con}^{(k)}$, $\Theta_l^{(k)}$, $\Theta_\tau^{(k)}$ for $k \in 1 : K$, Θ_M, w^0.

Output: Optimized parameters $\Theta_e^{(k)}$, $\Theta_d^{(k)}$, $\Theta_{con}^{(k)}$, $\Theta_l^{(k)}$ for $k \in 1 : K$, and Θ_M.

1: **repeat**
2:     Calculate the priority weights based on latency and sensing coverage of all cameras with parameter Θ_M.
3:     **for** $k = 1$ to K **do**
4:         Extract the features by the feature extractor of camera k with parameter $\Theta_e^{(k)}$.
5:         Compress the features based on the PIB framework with parameters $\Theta_d^{(k)}$, $\Theta_e^{(k)}$, $\Theta_{con}^{(k)}$, $\Theta_l^{(k)}$.
6:     **end for**
7:     Compute the loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ in Eqs. (23)-(24), respectively.
8:     Update parameters $\Theta_d^{(k)}$, $\Theta_e^{(k)}$, $\Theta_{con}^{(k)}$, $\Theta_l^{(k)}$ for $k \in 1 : K$, and Θ_M through backpropagation.
9: **until** Convergence of parameters $\Theta_d^{(k)}$, $\Theta_e^{(k)}$, $\Theta_{con}^{(k)}$, $\Theta_l^{(k)}$ for $k \in 1 : K$, and Θ_M.
10: **repeat**
11:     **for** $k = 1$ to K **do**
12:         Extract the features by the feature extractor of device k with parameter $\Theta_e^{(k)}$.
13:         Compress the features based on the multi-frame correlation model with parameters $\Theta_\tau^{(k)}$.
14:         Compute the empirical estimation of the loss function $\mathcal{L}_3^{(k)}$ in Eq. (25).
15:         Update parameters $\Theta_\tau^{(k)}$ through backpropagation.
16:     **end for**
17: **until** Convergence of parameters $\Theta_\tau^{(k)}$ for $k \in 1 : K$.

E. Gate Mechanism Based on Distributed Online Learning

The gate mechanism based on distributed online learning is designed to solve the combinatorial multi-armed bandit (CMAB) problem (11) formulated in Sec. III-B. In this subsection, we first introduce the intuitive ideas of gate mechanism. Then, we provide the details and explanation of the pseudocode. Finally, the evaluation of regret performance and communication cost for distributed execution is analyzed mathematically.

1) Distributed Online Learning for CMAB Problem: Firstly, we propose a distributed Upper Confidence Bound (UCB) algorithm to address this problem, leveraging the independence of each camera agent to learn the optimal transmission strategy. This approach is particularly effective for managing the dynamic nature of the multi-camera network, where real-time channel quality and server load can significantly impact the overall system performance. Specifically, we assume that each arm represents the connection establishment between a camera and an edge server. The super arm $(\mathcal{E}_{k,s}^{c \to e}, \mathcal{E}_{s,s_0}^{e \to e_0})$ is the combination of these connections. The reward is defined based on the gain in MODA by adding the k-th camera's feature to the ego camera in Eq. (9).

The intuitive idea behind using distributed UCB is to manage dynamic CSI and ROI efficiently. Each camera agent independently explores and exploits available edge servers based on local observations, making the system robust to changing network conditions. The algorithm has two phases: **exploration** and **exploitation**. In the exploration phase, each agent gathers information on potential rewards. In the exploitation phase, the agent selects the best action based on the UCB value, balancing exploration and exploitation under uncertainty. The UCB value for edge server s at time t is $\text{UCB}_{k,s}(t) = \hat{\mu}_{k,s}(t) + \alpha\sqrt{\frac{2\ln t}{N_{k,s}(t)}}$, where $\hat{\mu}_{k,s}(t)$ is the estimated reward, driving exploitation. The second term $(\alpha\sqrt{\frac{2\ln t}{N_{k,s}(t)}})$ accounts for uncertainty, encouraging exploration [53]. The key intuition is that actions that have been selected fewer times carry more uncertainty, so they are given a higher bonus to encourage exploration. Conversely, as an action is selected more often and its reward estimate becomes more reliable, the bonus decreases, leading to more exploitation of that action. The parameter α balances how aggressively the algorithm explores uncertain actions versus exploiting known rewards. The square root component diminishes as more information is gathered, while the logarithmic term ensures the exploration bonus decreases slowly, encouraging exploration of less frequently chosen actions. Algorithm 2 provides the pseudocode for the proposed gate mechanism with distributed online learning.

In Algorithm 2, each camera agent independently updates its observed channel state and edge server load (Line 7). This enables decentralized learning of the optimal transmission strategy without centralized control. Specifically, in Lines 9-11, each agent computes the UCB value $\text{UCB}_{k,s}(t) = \hat{\mu}_{k,s}(t) + \alpha\sqrt{\frac{2\ln t}{N_{k,s}(t)}}$ for each edge server s, where $\hat{\mu}_{k,s}(t)$ is the estimated reward and α is used to adjust the balance between exploration and exploitation. By selecting the action with the highest UCB value, the agent optimally balances exploring the underutilized connections and exploiting the high-reward connections based on its local observations, guiding the system towards an optimal transmission strategy. Lines 15-17 handle updates to action counts, cumulative rewards, and reward estimates, refining the UCB values. Periodic communication rounds (Line 19) ensure consistency across agents by allowing each to share local reward estimates with a central server, which aggregates these data and updates global estimates for synchronization (Lines 19-24). This process maintains synchronization across the distributed network while supporting scalability and adaptability.

2) Regret Analysis: The regret analysis reflects the efficiency and adaptability of the algorithm in optimizing network performance. A lower regret bound indicates that the algorithm performs close to the optimal strategy, ensuring high inference accuracy and efficient resource utilization despite the dynamic environment and varying network conditions. In the context of a multi-camera sensing network, the choices between different cameras and edge servers are interdependent. Furthermore, the connections between different cameras and edge servers exhibit heterogeneity, with each super arm having different distribution parameters. Therefore, we consider a CMAB prob-

Algorithm 2: Gate Mechanism with Distributed Online Learning (DOL)

1: Initialize parameters: α, β, γ
2: **for** each Camera $k = 1$ to K **do**
3:    Initialize reward estimates $\hat{\mu}_{k,s}(0) = 0$, action counts $N_{k,s}(0) = 0$, cumulative rewards $R_{k,s}(0) = 0$
4: **end for**
5: **for** each time step $t = 1$ to T **do**
6:    **for** each Camera $k = 1$ to K **do**
7:       Update channel state $\text{CSI}_{k,s}(t)$ and edge server load $l_s(t)$
8:       Select edge server s for fusion based on current state
9:       Compute UCB value for each edge server s:

$$\text{UCB}_{k,s}(t) = \hat{\mu}_{k,s}(t) + \alpha\sqrt{\frac{2\ln t}{N_{k,s}(t)}}$$

10:       Select action $a_k(t) = \left\{\mathcal{E}_{k,s}^{c\to e}, \mathcal{E}_{s,s_0}^{e\to e_0}\right\}$ that maximizes UCB value
11:       Execute action $a_k(t)$ and observe reward $r_k(t)$
12:       **if** constraints (11a) or (11b) or (11c) or (11d) or (11e) are violated **then**
13:          Set $r_k(t) = 0$
14:       **end if**
15:       Update action counts $N_{k,s}(t) = N_{k,s}(t-1) + 1$
16:       Update cumulative rewards $R_{k,s}(t) = R_{k,s}(t-1) + r_k(t)$
17:       Update reward estimates $\hat{\mu}_{k,s}(t) = \frac{R_{k,s}(t)}{N_{k,s}(t)}$
18:    **end for**
19:    **if** Communication round is started **then**
20:       **for** each Camera $k = 1$ to K **do**
21:          Aggregate rewards and action counts across agents
22:          Update global reward estimates $\hat{\mu}_{k,s}(t)$ for all edge servers s
23:       **end for**
24:    **end if**
25: **end for**

---

lem with a non-identically distributed (non-i.i.d) assumption and derive its regret upper bound in the following part.

The regret $R_L(T)$ is an important metric to evaluate the performance of online learning algorithm. The regret over a time horizon T is defined as the difference between the maximum expected reward obtainable by an optimal strategy and the expected reward obtained by the algorithm:

$$R_L(T) = \sum_{t=1}^{T}\left(\max_a \mathbb{E}[r(a)] - \mathbb{E}[r(a_k(t))]\right), \quad (26)$$

where a represents an action, and $\mathbb{E}[r(a)]$ is the expected reward for action a. To derive the regret bounds, we first establish a lemma using Bernstein's inequality in Lemma 1.

Lemma 1: *(Bernstein's inequality) Let $X_1, X_2, \ldots, X_T$ be independent random variables such that $|X_t - \mathbb{E}[X_t]| \le b$ almost surely. Then, for any $\epsilon > 0$,*

$$P\left(\left|\sum_{t=1}^{T}(X_t - \mathbb{E}[X_t])\right| \ge \epsilon\right) \le 2\exp\left(-\frac{\epsilon^2}{2\sum_{t=1}^{T}Var(X_t) + \frac{2}{3}\epsilon}\right). \quad (27)$$

Proof: Please refer to Sec. 2.8 in [54]. ∎

Theorem 1: *In a dynamic environment with non-identically distributed (non-i.i.d) rewards due to network heterogeneity, where the variance of the reward for arm k is denoted as $\sigma_{r_k}^2$, the cumulative regret $R(T)$ over T rounds of the distributed UCB algorithm is bounded by:*

$$R(T) \le O\left(\left(\sqrt{2\sum_{k=1}^{\mathcal{K}^{arm}}\sigma_{r_k}^2} + 2\mathcal{K}^{arm}\sqrt{\frac{2}{a_N}}\right)\mathcal{K}^{arm}\sqrt{T\ln T}\right.$$
$$\left. + \frac{2\mathcal{K}^{arm}}{3}\ln T - 2\mathcal{K}^{arm}\sqrt{\frac{2\ln T}{a_N}}\right), \quad (28)$$

where $\mathcal{K}^{arm} = \sum_{k=K_{\min}}^{K_{\max}} C_K^k S^{k+1}$ represents the maximum number of arms in the optimal super arm, σ_{r_k} is the standard deviation of the reward for arm k, and a_N is an upper bound on the linear growth rate of the number of times UCB algorithm arms are selected over time[6]. The non-i.i.d nature of the rewards represents the heterogeneity in the network where different arms can have different reward distributions due to varying network conditions, processing capabilities, and data qualities.

Proof: We start by defining the reward for camera k transmitting to edge server s at time t as $r_{k,s}(t)$. The mean reward for this transmission is denoted by $\mu_{k,s}$, and the variance of the reward is $\sigma_{k,s}^2$. The total variance in rewards across all camera-edge server pairs is represented by σ_r^2, which accounts for variability due to both the dynamic channel state information (CSI) and the edge server load fluctuations. To derive the regret bound, we use Bernstein's inequality to bound the sum of rewards for each camera-edge server pair. For any $\epsilon > 0$, Bernstein's inequality gives the following probability bound:

$$P\left(\left|\sum_{t=1}^{T}(r_{k,s}(t) - \mu_{k,s})\right| \ge \epsilon\right) \le 2\exp\left(-\frac{\epsilon^2}{2\sum_{t=1}^{T}\sigma_{k,s}^2 + \frac{2}{3}\epsilon}\right). \quad (29)$$

Now, we set ϵ as $\epsilon = \sqrt{2T\sigma_{k,s}^2\ln T} + \frac{2}{3}\ln T$ to account for the cumulative uncertainty over time T. Substituting this into the right-hand side of Bernstein's inequality, we get:

$$2\exp\left[-\frac{2T\sigma_{k,s}^2\ln T + \frac{4}{3}\ln T \cdot \sqrt{2T\sigma_{k,s}^2\ln T} + \frac{4}{9}(\ln T)^2}{2T\sigma_{k,s}^2 + \frac{2}{3}\sqrt{2T\sigma_{k,s}^2\ln T} + \frac{4}{9}\ln T}\right].$$

As T increases, the dominant terms in the expression are $2T\sigma_{k,s}^2\ln T$ in both the numerator and the denominator. Therefore, we approximate the right-hand side as:

$$2\exp\left(-\frac{\epsilon^2}{2\sum_{t=1}^{T}\sigma_{k,s}^2 + \frac{2}{3}\epsilon}\right) \approx 2\exp(-\ln T) = \frac{2}{T}. \quad (30)$$

---

[6]It indicates that $N_k(t)$ follows a linear growth trend, i.e., $N_k(t) \approx a_N t$.

Thus, combining Ineq. (29) and Eq. (30), we obtain:

$$P\left(\left|\sum_{t=1}^{T}(r_{k,s}(t)-\mu_{k,s})\right| \geq \sqrt{2T\sigma_{k,s}^2 \ln T} + \frac{2}{3}\ln T\right) \leq \frac{2}{T}. \quad (31)$$

It implies that, with high probability, when T is sufficiently large, we have:

$$\left|\sum_{t=1}^{T}(r_{k,s}(t)-\mu_{k,s})\right| \leq \sqrt{2T\sigma_{k,s}^2 \ln T} + \frac{2}{3}\ln T. \quad (32)$$

Therefore, the regret for each camera-edge server pair, denoted by arm (k,s), can be bounded as:

$$R_{k,s}(T) \leq \sum_{t=1}^{T}(\mu_{k,s}^* - \mu_{k,s}) + \sqrt{2T\sigma_{k,s}^2 \ln T} + \frac{2}{3}\ln T, \quad (33)$$

where $\sum_{t=1}^{T}(\mu_{k,s}^* - \mu_{k,s})$ represents the regret due to not always selecting the optimal arm (k,s), while the term $\sqrt{2T\sigma_{k,s}^2 \ln T} + \frac{2}{3}\ln T$ captures the uncertainty in reward estimation. Since the UCB algorithm selects the arm (k,s) that maximizes the UCB value, we have:

$$\mu_{k,s}^* \leq \text{UCB}_{k,s}(t) = \hat{\mu}_{k,s}(t) + \sqrt{\frac{2\ln t}{N_{k,s}(t)}}, \quad (34)$$

where $N_{k,s}(t)$ is the number of times the camera-edge server pair (k,s) has been selected up to time t. This implies that:

$$\mu_{k,s}^* - \mu_{k,s} \leq \left(\hat{\mu}_{k,s}(t) - \mu_{k,s}\right) + \sqrt{\frac{2\ln t}{N_{k,s}(t)}}. \quad (35)$$

Therefore, the upper bound on the cumulative loss (i.e., regret) for arm (k,s) can be expressed as:

$$\sum_{t=1}^{T}(\mu_{k,s}^* - \mu_{k,s}) \leq \sum_{t=1}^{T}\left(\hat{\mu}_{k,s}(t) - \mu_{k,s} + \sqrt{\frac{2\ln t}{N_{k,s}(t)}}\right). \quad (36)$$

Since $\hat{\mu}_{k,s}(t)$ is an unbiased estimate of $\mu_{k,s}$, its expected value is zero. Thus, the regret is mainly determined by the term $\sqrt{\frac{2\ln t}{N_{k,s}(t)}}$. We approximate the cumulative sum of this term by using an integral, given that $N_{k,s}(t)$ is assumed to grow linearly with time, i.e., $N_{k,s}(t) \approx a_N t$, where a_N is a constant. Under this assumption, we have:

$$\int_1^T \frac{1}{\sqrt{N_{k,s}(t)}}dt \approx \int_1^T \frac{1}{\sqrt{a_N t}}dt = \frac{2\sqrt{T}-2}{\sqrt{a_N}}. \quad (37)$$

Thus, the regret for a single arm (k,s) can be bounded as:

$$R_{k,s}(T) \leq \sqrt{2T\sigma_{k,s}^2 \ln T} + \frac{2}{3}\ln T + \left(2\sqrt{T}-2\right)\sqrt{\frac{2\ln T}{a_N}}. \quad (38)$$

To obtain the total regret $R(T)$, we sum the regret over all camera-edge server pairs in the set of arms $\mathcal{K}$:

$$R(T) \leq \sum_{(k,s)\in\mathcal{K}}\left(\sqrt{2T\sigma_{k,s}^2 \ln T} + \frac{2}{3}\ln T + \left(2\sqrt{T}-2\right)\sqrt{\frac{2\ln T}{a_N}}\right). \quad (39)$$

Then, we sum the bias and variance terms across all arms. For the bias term, we have:

$$\sum_{(k,s)\in\mathcal{K}}\sqrt{2T\sigma_{k,s}^2 \ln T} \leq \mathcal{K}^{\text{arm}}\sqrt{2T\sigma_r^2 \ln T}, \quad (40)$$

where $\mathcal{K}^{\text{arm}}$ is the number of arms in the optimal super arm, and σ_r^2 is the maximum variance among all arms. Therefore, when T is sufficiently large, the overall regret bound can be expressed as $R(T) \leq O\left(\left(\sqrt{2\sigma_r^2} + 2\sqrt{\frac{2}{a_N}}\right)\mathcal{K}^{\text{arm}}\sqrt{T\ln T} + \frac{2\mathcal{K}^{\text{arm}}}{3}\ln T - 2\mathcal{K}^{\text{arm}}\sqrt{\frac{2\ln T}{a_N}}\right)$. Thus, the cumulative regret $R(T)$ for the distributed UCB algorithm is bounded by the sum of the regret from all camera-edge server pairs, ensuring that the regret grows sub-linearly with respect to T. ∎

Proposition 3: *It is assumed that there are N camera agents and T time steps. The computational complexity of the proposed DOL method in Algorithm 2 for each camera agent at each time step is $O(\mathcal{K}^{arm}\log\mathcal{K}^{arm})$. The overall time complexity is $O(TN\mathcal{K}^{arm}\log\mathcal{K}^{arm})$.*

Proof: Each camera agent k computes the UCB value for each edge server s at each time step t. This involves updating the channel state and edge server load, computing the UCB values, selecting the optimal action, and updating the reward estimates. As for each camera agent, the complexity of updating the channel state and edge server load for each camera agent is $O(\mathcal{K}^{\text{arm}})$. Moreover, the complexity of computing the UCB value for each edge server is $O(\mathcal{K}^{\text{arm}})$. The complexity of selecting the action that maximizes the UCB value is $O(\mathcal{K}^{\text{arm}}\log\mathcal{K}^{\text{arm}})$. Thus, the complexity for each camera agent at each time step is $O(\mathcal{K}^{\text{arm}}\log\mathcal{K}^{\text{arm}})$. Given that there are N camera agents and T time steps, the overall time complexity is $O(TN\mathcal{K}^{\text{arm}}\log\mathcal{K}^{\text{arm}})$. ∎

Proposition 4: *Assuming there are X communication rounds over T time steps, the total communication cost of Algorithm 2 is $O(XN\mathcal{K}^{arm})$.*

Proof: Each communication round involves local communication between each camera agent and the central edge server, as well as the global aggregation and update phases. 1) *Local Communication*: Each camera agent k communicates its local reward estimates $\hat{\mu}_{k,s}(t)$ and action counts $N_{k,s}(t)$ for each edge server s to the central edge server. The communication cost for each agent per round is $O(\mathcal{K}^{\text{arm}})$. Given N agents, the total local communication cost per round is $O(N\mathcal{K}^{\text{arm}})$. 2) *Global Aggregation*: The central edge server aggregates the information from all N camera agents. The complexity of aggregating the information is $O(N\mathcal{K}^{\text{arm}})$. 3) *Global Update*: The central server then broadcasts the updated global reward estimates to all N agents. The communication cost for broadcasting is $O(N\mathcal{K}^{\text{arm}})$. Assuming there are X communication rounds over T time steps, the total communication cost is $O(XN\mathcal{K}^{\text{arm}})$. ∎
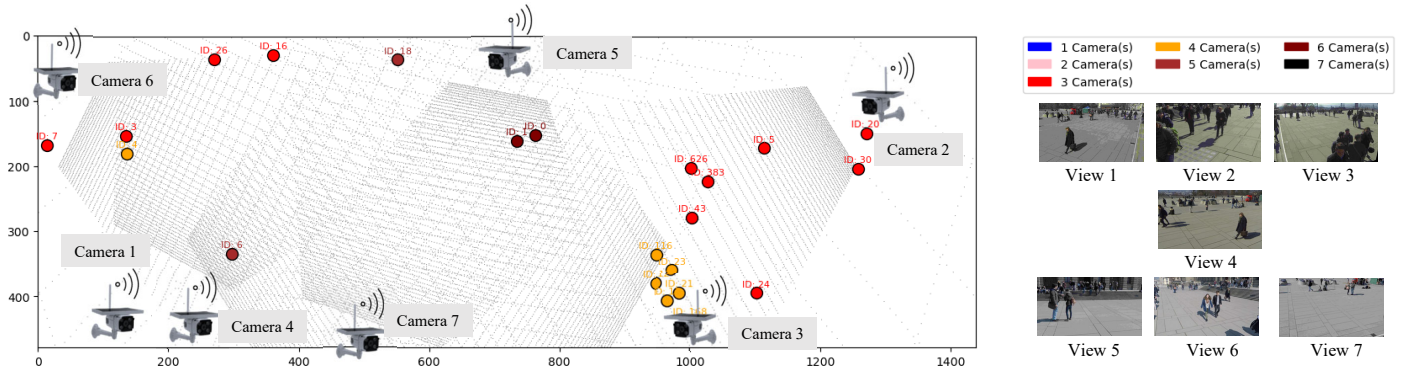
Fig. 5: The visualization of the edge video system. **Left**: We use contour lines to display the perception coverage of different cameras. The small dots in the grid represent pedestrians, with different colors of the dots indicating the number of cameras covering each pedestrian. It can be observed that areas closer to the perception center of cameras are covered by more cameras. **Right**: The visualization of the raw video data and the legend for different numbers of covered cameras.

(a) Pedestrian perception result using only **single camera**.

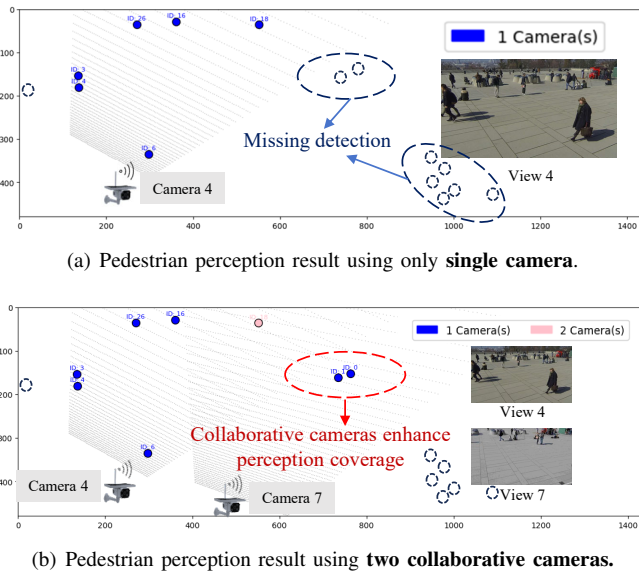(b) Pedestrian perception result using **two collaborative cameras.**

Fig. 6: Comparison of single and collaborative perception results. Fig. 6(a) shows the detection using only Camera 4. Fig. 6(b) demonstrates the enhanced detection capability achieved through the collaboration between Camera 4 and Camera 7.
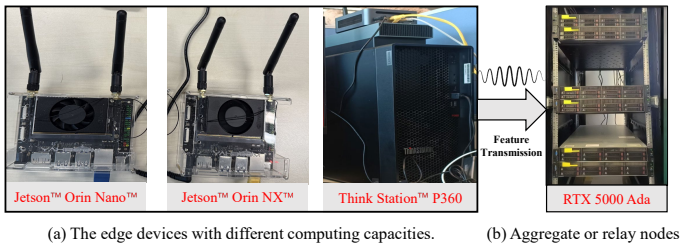

(a) The edge devices with different computing capacities.   (b) Aggregate or relay nodes.

Fig. 7: Real-world Hardware Testbed.

V. PERFORMANCE EVALUATION

A. Simulation Setup

We set up simulations to evaluate our PIB framework, aiming at predicting pedestrian occupancy in urban settings using multiple cameras. These simulations replicate a city environment, with variables like signal frequency and device density affecting the outcomes.

Our simulations use a 2.4 GHz operating frequency, a path loss exponent of 3.5, and a shadowing deviation of 8 dB. Devices emit an interference power of 0.1 Watts, with densities ranging from 10 to 100 devices per 100 square meters, allowing us to test different levels of congestion. The bandwidth is set to 2 MHz, with cameras located at about 200 meters from the edge server. We employ the *Wildtrack* dataset from EPFL, which features high-resolution images from seven cameras located in a public area, capturing unscripted pedestrian movements [55]. This dataset provides 400 frames per camera at 2 frames per second, documenting over 40,000 bounding boxes that highlight individual movements across more than 300 pedestrians. As shown in Fig. 7, our experimental setup features a practical hardware testbed that includes three distinct edge devices: NVIDIA Jetson™ Orin Nano™ 4GB, NVIDIA Jetson™ Orin NX™ 16GB, and ThinkStation™ P360. The edge devices collaboratively interact with edge servers equipped with RTX 5000 Ada GPUs for efficient video decoding. Our code will be made available at github.com/fangzr/PIB-Prioritized-Information-Bottleneck-Framework.

The primary measure we use is MODA, which assesses the system's ability to accurately detect pedestrians based on missed and false detections. We also look at the rate-performance tradeoff to understand how communication overhead affects system performance. For comparative analysis, we consider five baselines, including video coding and image coding:

- **TOCOM-TEM** [16]: A task-oriented communication framework utilizing a temporal entropy model for edge video analytics. It employs the deterministic Information Bottleneck principle to extract and transmit compact, task-relevant features, integrating spatial-temporal data on the server for improved inference accuracy.
- **JPEG** [56]: A widely used image compression standard employing lossy compression algorithms to reduce image data size, commonly used to decrease communication
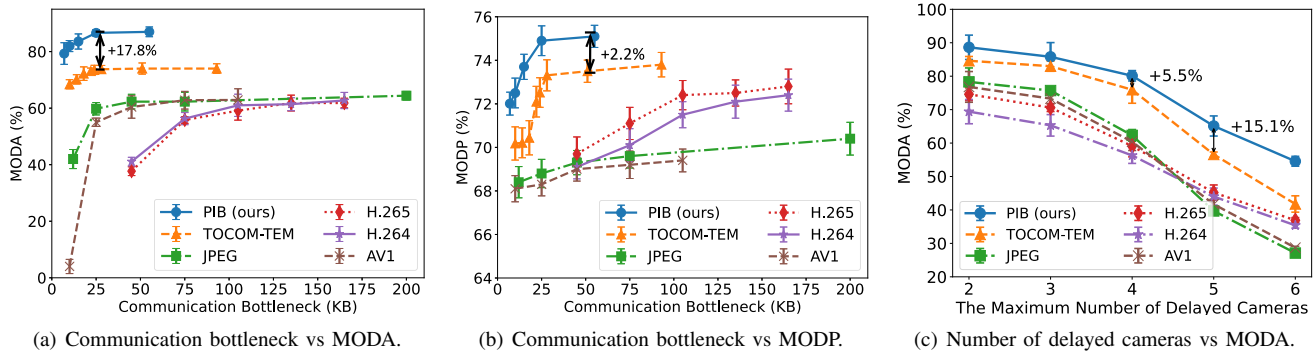
Fig. 8: Impact of communication bottlenecks and delayed cameras on perception accuracy.

load in networked camera systems.

- **H.265** [57]: Also known as High Efficiency Video Coding (HEVC) or MPEG-H Part 2, which offers up to 50% better data compression than its predecessor H.264 (MPEG-4 Part 10), while maintaining the same video quality, crucial for efficient data transmission in high-density camera networks.
- **H.264** [58]: Known as Advanced Video Coding (AVC) or MPEG-4 Part 10, which significantly enhances video compression efficiency, allowing high-quality video transmission at lower bit rates.
- **AV1** [59]: AOMedia Video 1 (AV1) is an open, royalty-free video coding format developed by the Alliance for Open Media (AOMedia), designed to succeed VP9 with improved compression efficiency. AV1 outperforms existing codecs like H.264 and H.265, making it ideal for online video applications.

In the simulation study, we examine the effectiveness of multiple camera systems in forecasting pedestrian presence. Unlike a single-camera configuration, this method minimizes obstructions commonly found in crowded locations by integrating perspectives from various angles. Fig. 5 demonstrates our experimental setup, where seven wireless edge cameras jointly perceive a 12m×36m area quantized into a 480×1440 grid using a resolution of 2.5 cm^2. We use contour lines to display the camera's perception range and the resolution of coordinates within that range. The denser the lines, the closer the perceived target is to the camera, and the higher the perception accuracy. Additionally, to clearly show the coverage of pedestrians at different positions by edge cameras, different colors represent the number of cameras covering each pedestrian. It can be observed that pedestrians in different locations have different probabilities of being detected, which will also affect the priority selection of cameras. Fig. 6(a) shows the perception results using a single camera (the 4th edge camera). The dashed circles represent pedestrians that are missing detection. It is evident that the perception range of a single camera is limited to its own angle and coverage area, resulting in numerous missing detections. In Fig. 6(b), we let the 4th and 7th edge cameras collaborate with each other. It can be observed that the collaboration enhances perception coverage, though there are still several pedestrians not detected compared to the results from seven edge cameras.

This highlights the improved but still limited capability of collaborative perception with only two cameras, indicating the necessity for a higher number of cameras to achieve comprehensive coverage and accurate pedestrian detection[7].

To evaluate the impact of communication bottlenecks and delayed cameras on perception accuracy, we present in Fig. 8(a)–8(c) the relationships between communication constraints and the perception accuracy. Nevertheless, the benefit of collaborative perception is accompanied by excessive communication overhead. The communication bottleneck refers to network capacity constraints that prevent real-time data transmission, causing frame latency. This issue is prevalent in UDP-based wireless streaming systems, where high throughput often results in out-of-order or delayed frames due to varying channel quality and jitter. Moreover, different coding schemes cause varying delays in dynamic channel conditions, misaligning data fusion due to channel quality and jitter. Therefore, in order to evaluate how latency differences affect perception accuracy, we set communication bottleneck constraints. In our experiments, we use MODA (Multiple Object Detection Accuracy) and MODP (Multiple Object Detection Precision) to assess coding efficiency and robustness.

In Fig. 8(a), PIB exhibits higher MODA across different communication bottlenecks compared to five baselines by more than 17.8%. This is due to PIB's strategic multi-view feature fusion, informed by channel quality and priority-based ROI selection. PIB prioritizes the shared information to mitigate delays that could degrade multi-camera perception accuracy. Interestingly, JPEG outperforms video coding schemes like H.265 and AV1 in our experiments, due to the low FPS of 2 used for video transmission, which does not leverage motion prediction advantages. AV1 performs well due to its high compression efficiency compared to H.264 and H.265. Fig. 8(b) shows that PIB achieves higher MODP performance compared to three other baselines. The results indicate that MODP is less affected by latency because it measures the precision of detection without considering missed detections, whereas MODA is more impacted as it accounts for both missed and false detections.

Fig. 8(c) depicts the performance rates of different compression techniques in a multi-view scenario in terms of the

[7]Our demo is available at the url: github.com/fangzr/PIB-Prioritized-Information-Bottleneck-Framework
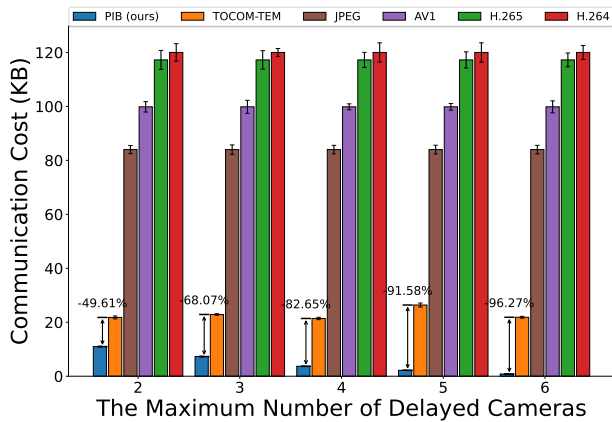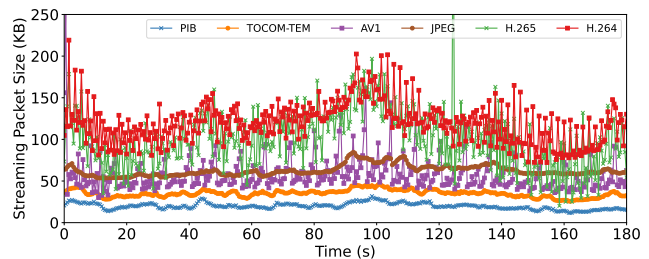
Fig. 9: Delayed cameras vs communication cost.

number of delayed cameras. Our proposed PIB method and TOCOM-TEM, both utilizing multi-frame correlation models, effectively reduce redundancy across multiple frames, achieving superior MODA at equivalent compression rates. PIB, in particular, employs a prioritized IB framework, enabling an adaptive balance between compression rate and collaborative sensing accuracy, optimizing MODA across various channel conditions. It is worth noting that the impact on collaborative perception MODA can be ignored in scenarios with fewer delayed cameras (<3). However, as channel conditions worsen and more cameras experience frame delays due to failing to meet communication bottleneck constraints, the performance significantly degrades.
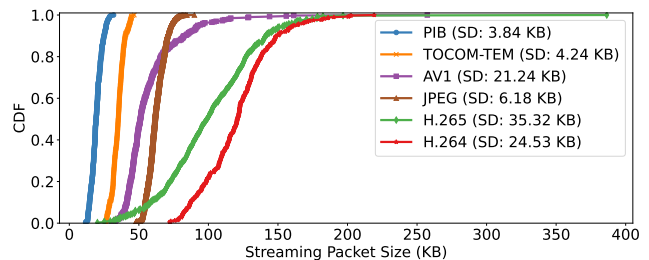
In Fig. 9, we analyze the impact of the number of delayed cameras on the communication cost[8] for various algorithms. The PIB algorithm demonstrates a significant reduction in communication costs as the number of delayed cameras increases. When the number of delayed cameras equals 4, PIB, utilizing a gate mechanism based on a distributed UCB algorithm, effectively filters out useless streaming data, greatly reducing communication costs. Compared to TOCOM-TEM, PIB achieves an impressive 82.8% decrease in communication costs. This efficiency is due to the algorithm's priority mechanism, which adeptly assigns weights and filters out adverse information caused by delays. Consequently, PIB prioritizes the transmission of high-quality features from cameras with more accurate occupancy predictions. For a fair comparison, baselines are selected at their highest MODA with the minimum communication cost data. Due to the use of an information bottleneck framework, PIB extracts only task-related features, resulting in a significantly reduced compression rate compared to five compression baselines.

Fig. 10 presents the streaming packet sizes and their cumulative distribution functions (CDF) for various compression algorithms. Fig. 10(a) illustrates the streaming packet sizes for PIB, TOCOM-TEM, AV1, JPEG, H.265, and H.264 over a duration of three minutes. All encoding methods were evaluated at their highest MODA with minimal communication

---

(a) Streaming packet size for PIB and five baselines over time.

(b) CDF for streaming packet sizes.

Fig. 10: (a) Streaming packet sizes for various compression algorithms over time slots. (b) Cumulative Distribution Functions (CDF) of the streaming packet sizes for different methods.

TABLE II: Impact of the Number of Fusion Cameras on Collaborative Perception Accuracy and Communication Cost.

Number	Comm. Cost	MODA (%)	MODP (%)
1	2.19 KB	65.11 (+17.99%)	71.53 (+2.59%)
2	3.68 KB	78.09 (+19.93%)	72.71 (+1.65%)
3	7.29 KB	84.99 (+8.85%)	72.92 (+0.29%)
4	10.98 KB	88.03 (+3.57%)	74.23 (+1.80%)
5	15.84 KB	88.64 (+0.69%)	75.15 (+1.24%)
6	17.68 KB	88.76 (+0.14%)	75.80 (+0.87%)
No Fusion	0.82 KB	55.17	69.72

costs. PIB consistently exhibits the smallest packet sizes, followed by TOCOM-TEM, indicating superior transmission efficiency. Additionally, PIB and TOCOM-TEM demonstrate less variability in packet sizes compared to AV1, enhancing transmission robustness under adverse channel conditions. JPEG compression yields smaller and more stable packet sizes than H.264 and H.265, likely due to the limited transmission rate of 2 fps restricting the efficiency of video codecs. Fig. 10(b) shows the CDF of streaming packet sizes for all algorithms. The standard deviation (SD) for each method is calculated as $SD = \sqrt{\frac{\sum_{i=1}^{n}(\text{Packet Size}_i - \text{Mean})^2}{n}}$. A lower SD indicates improved transmission robustness by reducing jitter and minimizing buffer requirements. PIB has the lowest SD (3.84 KB), followed by TOCOM-TEM (4.24 KB) and JPEG (6.18 KB). The other baseline methods exhibit higher SD values, underscoring PIB's advantage in minimizing both transmission requirements and packet size variability.

Our priority-based mechanism selects the camera with the most targets within RoI for the highest transmission priority. Collaboration priority is thus determined by the target count in
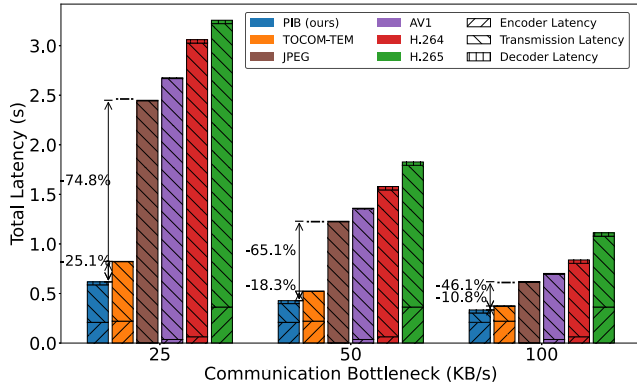
---

[8]The communication cost of a method is the average size of each frame. The instantaneous streaming rate is equal to the communication cost multiplied by the frames per second (fps).

Fig. 11: Communication bottleneck vs latency.

TABLE III: Encoder Latency Across Different Platforms.

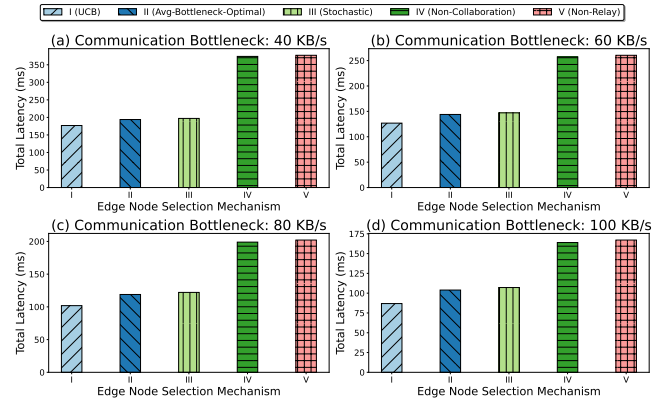| Phase \ Platform	Nano (ms)	Orin NX (ms)	P360 (ms)
Feature map generation	755.32±69.32	227.54±2.65	37.49±0.90
Entropy coding	10.83±3.51	1.79±0.75	0.40±0.11
Total encoder latency	**766.15±70.55**	**229.34±2.67**	**37.80±0.94**

Fig. 12: Impact of communication bottleneck on total latency for different edge server selection mechanisms.

Fig. 13: The number of edge servers vs various latencies under different communication bottlenecks.

each camera's perception area. As shown in Table II, just 0.82 KB of perception data achieves a MODA accuracy of 55.17% and MODP of 69.72%, highlighting significant redundancy among edge cameras. Adding more cameras initially improves perception significantly but offers diminishing returns as communication costs to the edge server increase. In Fig. 11, we show the relationship between communication bottleneck and total latency for different algorithms. By leveraging a priority information bottleneck framework and the UCB algorithm to reduce redundancy, our PIB, despite slightly higher encoding latency than the traditional video codecs, can achieve much lower transmission latency due to its efficient compression. Under a 25 KB/s bottleneck, our PIB reduces latency by 25.1% over TOCOM-TEM and 74.8% over JPEG. At 50 KB/s, our PIB outperforms TOCOM-TEM by 18.3% and JPEG by 65.1%, respectively. At 100 KB/s, PIB achieves 10.8% and 46.1% lower latency than TOCOM-TEM and JPEG, respectively. The encoding latency results of our PIB in different edge devices are presented in Table III. It can be observed that the feature map generation phase dominates the overall encoding latency, while the entropy coding phase contributes a negligible amount of time. Furthermore, edge devices with higher computing capacity exhibit significantly lower encoding latency.

Fig. 12 demonstrates the effectiveness of the proposed Gate Mechanism Based on Distributed Online Learning (Sec. IV-E). This figure evaluates total latency, defined as the sum of inference, relay, and transmission latency, excluding encoder latency, under different communication bottlenecks for various edge node selection mechanisms. Four baselines are used: *Avg-Bottleneck-Optimal* (exhaustive search for highest average bottleneck), *Stochastic* (random selection of relay and fusion nodes), *Non-Collaboration* (single edge server for fusion), and *Non-Relay* (lowest load edge server). The UCB method consistently achieves the lowest total latency, adapting efficiently to edge server load and channel conditions with

minimal overhead, thereby optimizing collaborative selection of edge servers. Fig. 13 illustrates the impact of different numbers of edge servers on the latency of multi-camera collaborative sensing data transmission and inference under varying communication bottlenecks. The results indicate that as the number of edge servers increases, the overall average latency of the cameras significantly decreases. This is because the communication bottleneck is comparable in magnitude to the size of the intermediate representations transmitted by the cameras. Therefore, increasing the number of edge servers markedly reduces latency, showcasing the effectiveness of adding more edge servers in enhancing system performance.

VI. Conclusion

In this paper, we have proposed the Prioritized Information Bottleneck (PIB) framework as a robust solution for collaborative edge video analytics. Our contributions are two-fold. First, we have developed a prioritized inference mechanism to intelligently determine the importance of different camera' FOVs, effectively addressing the constraints imposed by channel capacity and data redundancy. Second, the PIB framework showcases its effectiveness by notably decreasing communication overhead and improving tracking accuracy without requiring video reconstruction at the edge server. Extensive real-world experiments show that: PIB not only surpasses

the performance of conventional methods like TOCOM-TEM, JPEG, H.264, H.265, and AV1 with a marked improvement of up to 17.8% in MODA but also achieves a considerable reduction in communication costs by 82.65%, while retaining low latency and high-quality multi-view sensory data processing under less favorable channel conditions.

REFERENCES

[1] Z. Fang, S. Hu, L. Yang, Y. Deng, X. Chen, and Y. Fang, "PIB: Prioritized information bottleneck framework for collaborative edge video analytics," in *IEEE Global Communications Conference (GLOBECOM)*, Cape Town, South Africa, Dec. 2024, pp. 1–6.

[2] A. Padmanabhan, N. Agarwal, A. Iyer, G. Ananthanarayanan, Y. Shu, N. Karianakis, G. H. Xu, and R. Netravali, "Gemel: Model merging for memory-efficient, real-time video analytics at the edge," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, Boston, MA, 2023, pp. 973–994.

[3] 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, Mar. 2022.

[4] H. Wang, J. Huang, G. Wang, H. Lu, and W. Wang, "Contactless patient care using hospital IoT: CCTV camera based physiological monitoring in ICU," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 5781–5797, Aug. 2023.

[5] X. Yu, Z. Ying, N. Birkbeck, Y. Wang, B. Adsumilli, and A. C. Bovik, "Subjective and objective analysis of streamed gaming videos," *IEEE Transactions on Games*, pp. 1–14, 2023.

[6] J. Wang, L. Bai, Z. Fang, R. Han, J. Wang, and J. Choi, "Age of information based URLLC transmission for UAVs on pylon turn," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 8797–8809, Jan. 2024.

[7] M. Tang, C. Feng, and T. Q. S. Quek, "Decentralized semantic communication and cooperative tracking control for a UAV swarm over wireless MIMO fading channels," 2024. [Online]. Available: https://arxiv.org/abs/2411.06136

[8] G. Pan, H. Zhang, S. Xu, S. Zhang, and X. Chen, "Joint optimization of video-based AI inference tasks in MEC-assisted augmented reality systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 2, pp. 479–493, 2023.

[9] T. Kämäräinen, M. Siekkinen, A. Ylä-Jääski, W. Zhang, and P. Hui, "A measurement study on achieving imperceptible latency in mobile cloud gaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, Taipei, Taiwan, Jun. 2017, pp. 88–99.

[10] M. Xu, W. C. Ng, W. Y. B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. Shen, and C. Miao, "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 656–700, Nov. 2023.

[11] L. Corneo, N. Mohan, A. Zavodovski, W. Wong, C. Rohner, P. Gunningberg, and J. Kangasharju, "(How much) can edge computing change network latency?" in *IFIP Networking Conference (IFIP Networking)*. Espoo and Helsinki, Finland: IEEE, Jun. 2021, pp. 1–9.

[12] L. Marelli and G. Testa, "Scrutinizing the EU general data protection regulation," *Science*, vol. 360, no. 6388, pp. 496–498, May 2018.

[13] Ponemon Institute, "New ponemon institute study finds 60% of it and security leaders are not confident in their ability to secure access to cloud environments," 2021, online Accessed: 2022-07-20.

[14] Z. Fang, S. Hu, H. An, Y. Zhang, J. Wang, H. Cao, X. Chen, and Y. Fang, "PACP: Priority-aware collaborative perception for connected and autonomous vehicles," *IEEE Transaction of Mobile Computing (DOI: 10.1109/TMC.2024.3449371)*, Aug. 2024.

[15] S. Hu, Z. Fang, X. Chen, Y. Fang, and S. Kwong, "Towards full-scene domain generalization in multi-agent collaborative bird's eye view segmentation for connected and autonomous driving," 2024.

[16] J. Shao, X. Zhang, and J. Zhang, "Task-oriented communication for edge video analytics," *IEEE Transactions on Wireless Communications*, vol. 23, no. 5, pp. 4141–4154, May 2024.

[17] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, "Deep learning algorithm for autonomous driving using GoogleNet," in *IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, Jun. 2017, pp. 89–96.

[18] K. Gao, H. Wang, H. Lv, and W. Liu, "Localization-oriented digital twinning in 6G: A new indoor-positioning paradigm and proof-of-concept," *IEEE Transactions on Wireless Communications*, 2024.

[19] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360 video streaming: Solutions, challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[20] Y. Cui, G. Jiang, M. Yu, Y. Chen, and Y.-S. Ho, "Stitched wide field of view light field image quality assessment: Benchmark database and objective metric," *IEEE Transactions on Multimedia*, vol. 26, pp. 5092–5107, Nov. 2023.

[21] Z. Jiang, X. Zhang, Y. Xu, Z. Ma, J. Sun, and Y. Zhang, "Reinforcement learning based rate adaptation for 360-degree video streaming," *IEEE Transactions on Broadcasting*, vol. 67, no. 2, pp. 409–423, Oct. 2020.

[22] X. Chen, Y. Deng, H. Ding, G. Qu, H. Zhang, P. Li, and Y. Fang, "Vehicle as a service (VaaS): Leverage vehicles to build service networks and capabilities for smart cities," *IEEE Communications Surveys & Tutorials, (DOI: 10.1109/COMST.2024.3370169)*, 2024.

[23] S. Hu, Z. Fang, H. An, G. Xu, Y. Zhou, X. Chen, and Y. Fang, "Adaptive communications in collaborative perception with domain alignment for autonomous driving," *arXiv preprint arXiv:2310.00013*, 2023.

[24] S. Hu, Z. Fang, Z. Fang, Y. Deng, X. Chen, and Y. Fang, "Agentscodriver: Large language model empowered collaborative driving with lifelong learning," 2024.

[25] S. Hu, Z. Fang, Y. Deng, X. Chen, and Y. Fang, "Collaborative perception for connected and autonomous driving: Challenges, possible solutions and opportunities," *arXiv preprint arXiv:2401.01544*, 2024.

[26] X. Chi, H. Chen, G. Li, Z. Ni, N. Jiang, and F. Xia, "EDSP-Edge: Efficient dynamic edge service entity placement for mobile virtual reality systems," *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 2771–2783, Aug. 2024.

[27] Y. Jin, J. Liu, F. Wang, and S. Cui, "Ebublio: Edge-assisted multiuser 360° video streaming," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15 408–15 419, Apr. 2023.

[28] R. Tu, G. Jiang, M. Yu, Y. Zhang, T. Luo, and Z. Zhu, "Pseudo-reference point cloud quality measurement based on joint 2-D and 3-D distortion description," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, Jun. 2023.

[29] D. Wu, D. Zhang, M. Zhang, R. Zhang, F. Wang, and S. Cui, "ILCAS: Imitation learning-based configuration-adaptive streaming for live video analytics with cross-camera collaboration," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 6743–6757, Jun. 2024.

[30] Y.-F. Lu, J.-W. Gao, Q. Yu, Y. Li, Y.-S. Lv, and H. Qiao, "A cross-scale and illumination invariance-based model for robust object detection in traffic surveillance scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 6989–6999, Apr. 2023.

[31] Z. Bao, S. Yang, Z. Huang, M. Zhou, and Y. Chen, "A lightweight block with information flow enhancement for convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3570–3584, Aug. 2023.

[32] R. Lu, Z. Cheng, B. Chen, and X. Yuan, "Motion-aware dynamic graph neural network for video compressive sensing," *IEEE Transactions on Pattern Analysis and Machine Intelligence, (DOI: 10.1109/TPAMI.2024.3395804)*, pp. 1–17, May 2024.

[33] M. Xu, H. Du, D. Niyato, J. Kang, Z. Xiong, S. Mao, Z. Han, A. Jamalipour, D. I. Kim, X. Shen, V. C. M. Leung, and H. V. Poor, "Unleashing the power of edge-cloud generative ai in mobile networks: A survey of AIGC services," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 1127–1170, Jan. 2024.

[34] T. Li, J. Sun, Y. Liu, X. Zhang, D. Zhu, Z. Guo, and L. Geng, "ESMO: Joint frame scheduling and model caching for edge video analytics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 8, pp. 2295–2310, May 2023.

[35] M. Khani, G. Ananthanarayanan, K. Hsieh, J. Jiang, R. Netravali, Y. Shu, M. Alizadeh, and V. Bahl, "RECL: Responsive resource-efficient continuous learning for video analytics," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 917–932.

[36] 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, Jan. 2023.

[37] P. Zhang, W. Xu, Y. Liu, X. Qin, K. Niu, S. Cui, G. Shi, Z. Qin, X. Xu, F. Wang, Y. Meng, C. Dong, J. Dai, Q. Yang, Y. Sun, D. Gao, H. Gao, S. Han, and X. Song, "Intellicise wireless networks from semantic communications: A survey, research issues, and challenges," *IEEE Communications Surveys & Tutorials (DOI: 10.1109/COMST.2024.3443193)*, Aug. 2024.

[38] Y. Shao, Q. Cao, and D. Gündüz, "A theory of semantic communication," *IEEE Transactions on Mobile Computing (DOI: 10.1109/TMC.2024.3406375)*, pp. 1–18, May 2024.

[39] H. Xie, Z. Qin, and G. Y. Li, "Semantic communication with memory," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 8, pp. 2658–2669, Jun. 2023.

[40] S. Zhou, D. Van Le, R. Tan, J. Q. Yang, and D. Ho, "Configuration-adaptive wireless visual sensing system with deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 9, pp. 5078–5091, May 2023.

[41] Y. Chen, S. Zhang, Y. Jin, Z. Qian, M. Xiao, W. Li, Y. Liang, and S. Lu, "Crowdsourcing upon learning: Energy-aware dispatch with guarantee for video analytics," *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 3138–3155, 2024.

[42] S. Hu, Z. Lou, X. Yan, and Y. Ye, "A survey on information bottleneck," *IEEE Transactions on Pattern Analysis and Machine Intelligence, (DOI: 10.1109/TPAMI.2024.3366349)*, pp. 1–20, Feb. 2024.

[43] A. Pensia, V. Jog, and P.-L. Loh, "Extracting robust and accurate features via a robust information bottleneck," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 131–144, May 2020.

[44] S. Wang, C. Li, Y. Li, Y. Yuan, and G. Wang, "Self-supervised information bottleneck for deep multi-view subspace clustering," *IEEE Transactions on Image Processing*, vol. 32, pp. 1555–1567, Feb. 2023.

[45] R. Wang, X. He, R. Yu, W. Qiu, B. An, and Z. Rabinovich, "Learning efficient multi-agent communication: An information bottleneck approach," in *International Conference on Machine Learning (ICML)*, Virtual Event, 2020, pp. 9908–9918.

[46] D. Kim, S. Moon, D. Hostallero, W. Kang, T. Lee, K. Son, and Y. Yi, "Learning to schedule communication in multi-agent reinforcement learning," *arXiv preprint arXiv:1902.01554*, 2019.

[47] Y. Du, B. Liu, V. Moens, Z. Liu, Z. Ren, J. Wang, and H. Zhang, "Learning correlated communication topology in multi-agent reinforcement learning," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, London, United Kingdom, May, 2021, pp. 456–464.

[48] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao, "Multi-agent game abstraction via graph attention neural network," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, New York, NY, February, 2020, pp. 7211–7218.

[49] Y. Niu, R. R. Paleja, and M. C. Gombolay, "Multi-agent graph-attention communication and teaming," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, London, United Kingdom, May, 2021, pp. 964–973.

[50] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA (Virtual), 2020, pp. 4106–4115.

[51] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[52] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Conf. on Learning Representations (ICLR)*, Toulon, France, Apr. 2017, pp. 1–9.

[53] G. Xiong, S. Wang, G. Yan, and J. Li, "Reinforcement learning for dynamic dimensioning of cloud caches: A restless bandit approach," *IEEE/ACM Transactions on Networking*, vol. 31, no. 5, pp. 2147–2161, Oct. 2023.

[54] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018, vol. 47.

[55] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, Jun. 2018, pp. 5030–5039.

[56] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992.

[57] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Transactions on circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Oct. 2012.

[58] ITU-T Recommendation H.264 and ISO/IEC 14496-10, *Advanced Video Coding for Generic Audiovisual Services*, International Telecommunication Union Std., 2003. [Online]. Available: https://www.itu.int/rec/T-REC-H.264

[59] J. Han, B. Li, D. Mukherjee, C.-H. Chiang, A. Grange, C. Chen, H. Su, S. Parker, S. Deng, U. Joshi *et al.*, "A technical overview of AV1," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1435–1462, Sep. 2021.

APPENDIX A
THE NETWORK ARCHITECTURE OF PIB

The PIB framework is designed with efficient computational distribution across the camera and edge server to achieve low latency and high accuracy. As illustrated in Figs. 3 and 4, the detailed network architecture is given as follows:

Camera: The camera executes the first two stages of the pipeline: (i) *Feature Extraction* and (ii) *Hyper Encoder*. These stages preprocess the raw video data into a compressed intermediate representation suitable for transmission to the edge server.

Edge Server: Upon receiving the compressed bitstream, the edge server executes (iii) *Hyper Decoder*, (iv) *Projection and Multiview Aggregation*, and (v) *Spatial Aggregation and Classification*. These stages reconstruct the feature maps, fuse multiview information, and generate the pedestrian occupancy map. Below is the detailed breakdown of each stage:

(i) Feature Extraction (ResNet-18 Backbone): The feature extraction employs a modified ResNet-18 backbone to retain spatial resolution critical for subsequent projection and fusion. We assume that B denotes the batch size, H and W are the height and width of the input image.

Layer Name	Input Dimensions	Output Dimensions
Input Image	$[B, 3, H, W]$	$[B, 3, 720, 1280]$
ResNet-18 (Part 1)	$[B, 3, 720, 1280]$	$[B, 64, 180, 320]$
ResNet-18 (Part 2)	$[B, 64, 180, 320]$	$[B, 512, 90, 160]$
Feature Extraction	$[B, 512, 90, 160]$	$[B, 8, 90, 160]$

(ii) Hyper Encoder and (iii) Decoder for Compression: The Hyper Encoder compresses the extracted features at the camera, while the Hyper Decoder reconstructs them at the edge server.

Layer Name	Input Dimensions	Output Dimensions
Hyper Encoder	$[B, 8, 90, 160]$	$[B, 4, 30, 40]$
Hyper Decoder	$[B, 4, 30, 40]$	$[B, 8, 90, 160]$

(iv) Projection and Multiview Aggregation: Feature maps are projected onto a common ground plane and aggregated with coordinate maps for multiview fusion. H_g and W_g are the height and width of the projected ground plane grid.

Layer Name	Input Dimensions	Output Dimensions
Projection	$[B, 8, 90, 160]$	$[B, 8, H_g, W_g]$
Concatenation	$[B, 8, H_g, W_g]$	$[B, N \times 8 + 2, H_g, W_g]$

(v) Spatial Aggregation and Classification: The aggregated features are processed to produce the final pedestrian occupancy map.

Layer Name	Input Dimensions	Output Dimensions
Map Classifier	$[B, N \times 8 + 2, H_g, W_g]$	$[B, 1, H_g, W_g]$