Task-Oriented Integrated Sensing and Semantic Communications for Multi-Device Video Analytics

Yinghui He, *Member, IEEE*, Xin Li, *Member, IEEE*, and Jun Luo, *Fellow, IEEE*

***Abstract*—Video analytics plays a vital role in modern applications such as public safety and smart cities, yet transmitting high-resolution video over wireless networks is severely constrained by bandwidth and latency. Existing semantic communication approaches alleviate communication overhead by discarding irrelevant content, but they often impose prohibitive computational costs on resource-constrained surveillance devices. To overcome this limitation, we propose SenSem, a sensing-assisted semantic communication framework that uniquely leverages channel state information (CSI) to reduce both communication and computation overhead. SenSem first exploits location cues embedded in CSI to estimate the region of interest and crop frames before upload. On the cropped frames, a lightweight semantic evaluator scores blocks, and a joint block selection and transmit power control algorithm maximizes the analytics performance for multi-device uplink; at the edge, a sensing-assisted analytics network injects spatial cues to further boost inference. Extensive evaluations on the WARP platform demonstrate that SenSem consistently outperforms state-of-the-art baselines, achieving superior video analytics accuracy under strict latency constraints. By seamlessly reducing both transmission and device-side computation overhead, SenSem offers a scalable and efficient solution for next-generation wireless video analytics systems.**

***Index Terms*—Integrated sensing and communications, semantic communications, video analytics, deep learning, edge intelligence, task-oriented communications**

I. Introduction

Advancements in artificial intelligence (AI) have greatly enhanced the feasibility of video analytics in practical systems, fueling its rapid adoption across areas, such as intelligent transportation [1], public security [2], and health monitoring [3]. Most video analytics tasks rely on complex deep network architectures, requiring extensive computation resources provided by the edge server located at the base station (BS) for support [4]. However, transmitting high-definition video streams from a large number of cameras to the BS imposes tremendous bandwidth demands and can easily exceed the capacity of existing wireless systems. This issue is particularly critical for latency-sensitive applications such as surveillance, where continuously tracking moving targets and analyzing their behavior in real time are essential for safety and security. The resulting tension between ever-growing video traffic and limited wireless resources raises a fundamental question: how can scalable and real-time video analytics be achieved under stringent bandwidth and latency constraints?

To mitigate this issue, recent research has explored semantic communication as a promising paradigm [5], which transmits compact semantic information instead of raw data. Two mainstream approaches have emerged: recovery-oriented methods [6]–[8], which exploit spatiotemporal correlations to compress video frames for reconstruction at the BS, and task-oriented methods [9], [10], which directly extract task-relevant features for transmission to the BS. Both approaches significantly reduce communication overhead compared with conventional video transmission. Nevertheless, they commonly require computation-intensive deep neural networks to be deployed at the device side in order to extract semantic representations from complex video frames. This assumption is misaligned with real-world surveillance scenarios, where the majority of cameras are low-cost, resource-constrained devices incapable of executing such heavy models. As a result, despite their potential, existing semantic communication pipelines face a major barrier to practical adoption in large-scale IoT-based monitoring systems.

Fortunately, the channel state information (CSI) between the transmitter and receiver also contains sensing information about the surveillance context and targets, and thus, has recently been explored for various sensing tasks, such as activity recognition [11], [12], localization [13], and respiration detection [14], [15], spurring the development of integrated sensing and communications (ISAC) [16]–[18]. Compared to video's rich but high-dimensional content, CSI offers lighter-weight, coarse-grained insights into target position and motion [19], [20]. Such a characteristic helps filter out redundant information in video while demanding far less computational overhead. Moreover, since CSI is inherently available at the BS for communication scheduling and optimization, it can be readily processed using adequate edge computation resources, thereby alleviating the computation burden on resource-constrained surveillance devices.

Motivated by these observations, we propose SenSem, a novel framework that integrates sensing with semantic communication to reduce transmission and computation overhead while enabling high-performance video analytics. We focus on a multi-device single-input-multiple-output (SIMO) video analytics system, where numerous surveillance devices upload data to the BS for real-time inference under strict latency constraints. The key idea of SenSem is to exploit target-related spatial information embedded in CSI to guide both communication and analysis. By leveraging CSI, extraneous regions in video frames can be cropped before transmission, thereby lowering the uplink burden without sacrificing the information needed for analytics. Furthermore, instead of relying on heavy deep models at the device side, lightweight semantic processing is employed to identify the most relevant

Y. He, X. Li, and J. Luo are with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (email: yinghui.he@ntu.edu.sg, l.xin@ntu.edu.sg, junluo@ntu.edu.sg).

portions of the cropped frames for uploading. Finally, the CSI is also utilized at the edge server to enhance inference, demonstrating how sensing and communication can be tightly integrated to support scalable wireless video analytics.

However, realizing this approach entails several challenges. First, the location information inferred from CSI is inherently limited by the number of antennas and the available bandwidth, leading to non-negligible estimation errors that may cause critical regions to be omitted during frame cropping. Second, when a large number of surveillance devices upload simultaneously, severe interference and stringent latency requirements make it difficult to guarantee timely analytics, highlighting the need for an efficient communication strategy that achieves high accuracy under limited delay. Finally, while CSI provides spatial information from the BS's perspective, the video frames are captured from the device's perspective, creating a modality gap that complicates the direct fusion of the two types of information.

To address these challenges, we first propose a location-assisted region of interest (RoI) estimation method for video cropping. Instead of relying solely on the estimated location, we use the peak range in the ToF-AoA spectra [21] to determine the ranges of ToF and AoA and then map them to the RoI using the relationship between the image plane and the world coordinate system. For the second challenge, we design a low-complexity semantic evaluator by expanding the method in [22] and further formulate an optimization problem. By solving it, we develop a joint block selection and transmit power control algorithm for maximizing the analytics performance under the constraints of communication resource and latency. Finally, taking pose estimation as an example, we utilize transformer to extract pose features and propose a sensing-assisted video analytics network that integrates visual and sensing data at the feature level while remaining compatible with the state-of-the-art (SOTA) video-based solution [23]. In summary, we make the following major contributions:

- We propose SenSem, a novel framework that integrates sensing and semantic communications to reduce the computation and communication overhead for multi-device video analytics.
- We introduce a location-assisted RoI estimation method that utilizes the location information to crop video, avoiding unnecessary computation and communication.
- We develop a joint block selection and transmit power control algorithm to maximize the analytics performance within the latency requirement.
- We design a sensing-assisted video analytics network that fuses visual and sensing data at the feature level to enhance edge video analytics.
- We validate SenSem using real-world experiments on the WARP platform and simulation, demonstrating its superior performance compared to benchmark schemes.

The rest of this paper is organized as follows. Section II summarizes the related works. Section III introduces the model of the studied multi-device video analytics system. Section IV presents the detailed design of the proposed SenSem. Section V specifies the evaluation setup and reports evaluation results, and the whole paper is concluded in Section VI.

Notations: Scalars are denoted by lower case, vectors are denoted by boldface lower case, and matrices are denoted by boldface upper case. $\boldsymbol{I}$ represents an identity matrix and $\mathbf{0}$ denotes an all-zero vector. $(\cdot)^*$, $(\cdot)^T$, and $(\cdot)^H$ denote complex conjugate, transpose, and Hermitian transpose, respectively. For a vector $\boldsymbol{a}$, $||\boldsymbol{a}||$ represents its Euclidean norm. $|\cdot|$ represents the absolute value of a complex scalar. $\mathbb{C}^{m\times n}$ ($\mathbb{R}^{m\times n}$) denotes the space of $m \times n$ complex (real) matrix.

II. Related Works

This work is mostly related to semantic communications, as well as similar topics, e.g., task-oriented communications [24]. Existing semantic communication primarily serves the transmission of text [25]–[27], images [28]–[30], and videos [6], [7], [9], [10]. For text transmission, the authors in [25] introduced a Transformer-based system that transmits sentence semantics rather than individual bits, using a newly defined metric, namely sentence similarity. To further reduce the overhead, the authors of [27] explored contextual correlations and designed a DL network with a memory module for the question-answer task. For the image, authors in [28] proposed a convolutional semantic encoder to extract semantic concepts (including the category, spatial arrangement, and visual feature) as the representation unit, and further designed a generative adversarial network-based semantic decoder for image recovery. Moving beyond recovery, the authors [30] considered diverse downstream image tasks and developed a joint source and channel coding to realize simultaneous image recovery and other tasks. Although video can be viewed as a combination of multiple images, adjacent frames in a video are inherently correlated, and traditional video encoding methods leverage this correlation [31]. Along this line, the authors of [32] compressed the differential data between P-frames and I-frames to reduce transmission volume. However, it overlooks the fact that I-frames, due to their larger size, often dominate bandwidth usage. To address this issue, the authors of [7] proposed a reinforcement learning-based bandwidth allocation strategy, which adaptively assigns bandwidth to I-frames and P-frames. Beyond video recovery, the authors of [9] focused on edge video analytics tasks, further reducing transmission overhead by extracting only task-relevant features. Nevertheless, these studies primarily focus on using DL techniques to remove redundant information, which is generally unaffordable for surveillance devices. To address this issue, we leverage the sensing information carried by CSI to remove redundancy with low computation overhead.

This work is also related to the ISAC, and ISAC has two typical applications: communication-assisted sensing [33]–[35] and sensing-assisted communication [36], [37]. The latter primarily leverages the sensing function to enhance channel estimation, thereby reducing pilot overhead and improving communication efficiency. For example, the authors in [37] focused on vehicle-to-infrastructure communication. The sensing function is used to estimate the location of target vehicles and a Kalman filter is further applied for tracking and prediction, thereby significantly reducing the overhead of beam tracking.

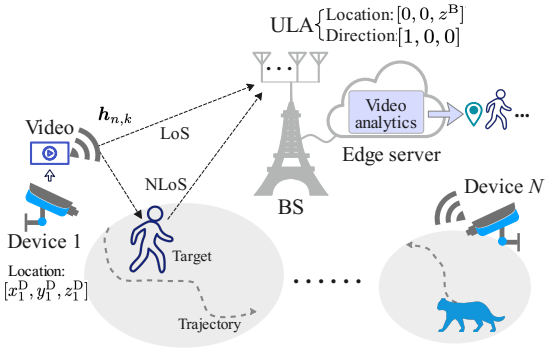


Fig. 1: A multi-device video analytics system.

Building upon the foundation of ISAC, recent research has begun to integrate semantic communication with sensing to further enhance system efficiency. For example, [38] investigated vehicular networks where the BS performs both semantic communication and sensing, and proposed a beamforming algorithm that jointly accounts for communication and sensing performance. Extending this line of work, [39] incorporated security considerations and designed beamforming strategies to prevent eavesdropping by unauthorized users. Different from treating these functions independently, [40] has leveraged semantic communication to transmit sensing data collected by IoT devices, thereby reducing communication overhead. Distinct from these approaches, our work aims to exploit the sensing information inherently contained in CSI, and utilize it to reduce the communication overhead, enabling more efficient semantic transmission.

III. System Model

As depicted in Fig. 1, we consider a multi-device video analytics system consisting of massive surveillance devices $\{1, \cdots, N\}$ and one BS equipped with an edge server, with the location of the BS and device n being $\boldsymbol{p}^{\mathrm{B}} = [0, 0, z^{\mathrm{B}}]^T$ and $\boldsymbol{p}_n^{\mathrm{D}} = [x_n^{\mathrm{D}}, y_n^{\mathrm{D}}, z_n^{\mathrm{D}}]^T$, respectively. The BS is equipped with M receive antennas, and the antennas form a uniform linear array (ULA) with the direction being $[1, 0, 0]^T$. Considering that most surveillance devices are relatively inexpensive, we assume that they are equipped with a single transmit antenna and limited computation capability. Each device aims to monitor a surveillance zone and analyze the video to realize the application of pose estimation.[1] Due to the limited computational capability of the device, local video analytics would lead to high latency and congestion, and thus, the video would be uploaded to the edge server at the BS via wireless transmission for further processing.

To leverage the contextual relevance of video content, we assume that the device performs an upload after collecting a group of pictures (GoP). To realize the multi-device uploading, we divide the devices into J groups, denoted by $\{\mathbb{N}_j, \forall j\}$, with the devices in one group uploading their GoP using the multi-user SIMO (MU-SIMO) technique, simultaneously, and the transmission of different groups following a time-division transmission scheme. Moreover, during the transmission, the OFDM technique is adopted to avoid inter-symbol interference with the bandwidth being B^{w} and the number of subcarriers being K. The transmission contains two phases, i.e., pilot transmission and data transmission. The first one is used to measure CSI between each device and the BS, and the second phase is used for transmitting video data. Considering the requirement of real time, the total latency of the processing at the device and the data transmission should be no more than an upper bound, denoted by $T^{\max}$, for all devices.

In the following, we first introduce the vision model and channel model, and then show how sensing benefits semantic communications by exploring the relationship between information in the CSI and that in the video.

A. Vision Model

In the following, we aim to establish the vision model between the point in the world coordinate system and the pixel in the image for device n. First of all, we need to analyze the transformation from world coordinates to camera coordinates. As shown in Fig. 2, through one translation and two rotations, the camera coordinate system of device n can be derived from the world coordinate system. Specifically, first, the origin of the coordinate system is translated to $\boldsymbol{p}^{\mathrm{D}}$. Then, the coordinate system is rotated about the z-axis by an angle ϑ_z and about the y-axis by an angle ϑ_y, successively. Thus, the transformation from the world coordinate system to the camera coordinate system is given as follows:

$$\boldsymbol{p}_n' = \begin{bmatrix} \cos(\vartheta_y) & 0 & -\sin(\vartheta_y) \\ 0 & 1 & 0 \\ \sin(\vartheta_y) & 0 & \cos(\vartheta_y) \end{bmatrix} \begin{bmatrix} \cos(\vartheta_z) & \sin(\vartheta_z) & 0 \\ -\sin(\vartheta_z) & \cos(\vartheta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} (\boldsymbol{p} - \boldsymbol{p}_n^{\mathrm{D}}),$$

where $\boldsymbol{p}_n'$ and $\boldsymbol{p}$ denote the camera coordinate and the world coordinate for the same point, respectively. After that, we need to analyze the transformation from the camera coordinate to pixels in the image plane. Let d^{I} denote the focal length, and assume that one pixel in the image corresponds to the length of Δy and Δz for the y-axis and z-axis of the image plane. $\boldsymbol{p}_n' = [x_n', y_n', z_n']^T$ is projected in to pixel (u, v) in the image plane [41], as

$$u = d_y^{\mathrm{I}} y_n'/x_n' + u_0, \quad v = d_z^{\mathrm{I}} z_n'/x_n' + v_0, \tag{1}$$

where $d_y^{\mathrm{I}} = d^{\mathrm{I}}/\Delta y$ and $d_z^{\mathrm{I}} = d^{\mathrm{I}}/\Delta z$ are the focal lengths scaled by pixel size, and (u_0, v_0) is the principal point coordinate on the image plane.

B. Channel Model and Data Rate

In this paper, we consider the ray-tracing-based channel model [42] that is widely adopted in the existing communication standards and works. Specifically, as shown in Fig. 1, the channel between the n-th device and the BS contains L paths, with the first one being the line-of-sight (LoS) path and the remaining being non-line-of-sight (NLoS) paths. Let $a_{n,l} \in \mathbb{C}$, $\theta_{n,l} \in \mathbb{R}$, and $\tau_{n,l} \in \mathbb{R}$ denote the path attenuation, angle of arrival (AoA), and time of flight (ToF) for the l-th path of the channel between device n and the BS, respectively. The CSI

[1] In this paper, we take pose estimation as an example since it is one of the widely adopted applications, and the proposed scheme could be easily extended for other applications.

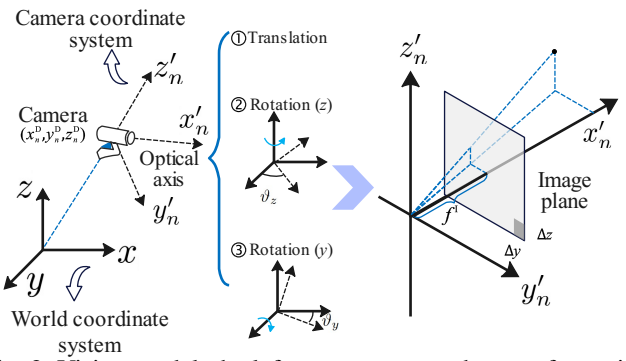


Fig. 2: Vision model: the left part represents the transformation from the world coordinate system to the camera coordinate system, and the right part represents the projection from the camera coordinate system to the image plane.

vector at the k-th subcarrier, denoted by $\boldsymbol{h}_{n,k} \in \mathbb{C}^{M\times 1}$, can be expressed as

$$\boldsymbol{h}_{n,k} = \sum_{l=1}^{L} a_{n,l} e^{i2\pi\tau_{n,l}(f^{\mathrm{c}}+k\Delta f)} \boldsymbol{\alpha}(\theta_{n,l}), \tag{2}$$

where f^{c} is the starting frequency of the broadband, and $\Delta f = B^{\mathrm{w}}/K$ is the subcarrier spacing, $\boldsymbol{\alpha}(\theta) = [1, e^{i2\pi d\sin(\theta)/\lambda}, \cdots, e^{i2\pi(M-1)d\sin(\theta)/\lambda}] \in \mathbb{C}^{M\times 1}$ is the steering vector with λ being the wavelength and d being the antenna spacing. For the convenience of analysis, we assume that the index of the path related to the target in the surveillance area is 2.

Recall that we adopt the time-division scheme for different device groups and the MU-SIMO technique for each group. Let $s^{\mathrm{T}}_{n,k}$ denote the transmit data of the n device in the j-th group at the k-th subcarrier with $\mathbb{E}\{|s^{\mathrm{T}}_{n,k}|^2\} = 1$ and $p^{\mathrm{T}}_{n,k}$ represent the corresponding transmit power. After undergoing the wireless channel $\boldsymbol{h}_{n,k}$, the received signal at the BS can be expressed as

$$\boldsymbol{y}^{\mathrm{T}}_{n,k} = \boldsymbol{h}_{n,k}\sqrt{p^{\mathrm{T}}_{n,k}}s^{\mathrm{T}}_{n,k} + \sum_{n'\in\mathbb{N}_j\setminus\{n\}} \boldsymbol{h}_{n',k}\sqrt{p^{\mathrm{T}}_{n',k}}s^{\mathrm{T}}_{n',k} + \boldsymbol{n}, \tag{3}$$

where $\boldsymbol{n} \in \mathbb{C}^{M\times 1}$ is the complex Gaussian noise with zero mean and covariance matrix $\sigma^2\boldsymbol{I}$. In the above, the first term on the right-hand side of the equation represents the signal received from the n-th device, while the second term represents interference from other devices. To combine the signal from different receive antennas, the BS adopts a receiver vector, denoted by $\boldsymbol{w}_{n,k} \in \mathbb{C}^{M\times 1}$, for the n-th device, and the corresponding signal-to-interference-plus-noise ratio (SINR) can be expressed as

$$\gamma_{n,k} = \frac{p^{\mathrm{T}}_{n,k}|\boldsymbol{w}^H_{n,k}\boldsymbol{h}_{n,k}|^2}{\sigma^2||\boldsymbol{w}_{n,k}||^2 + \sum_{n'\in\mathbb{N}_j\setminus\{n\}} p^{\mathrm{T}}_{n',k}|\boldsymbol{w}^H_{n,k}\boldsymbol{h}_{n',k}|^2}. \tag{4}$$

Thus, the sum of data rate over K subcarriers for the n-th device is

$$R_n = \sum_{k=1}^{K} \Delta f \log_2(1+\gamma_{n,k}). \tag{5}$$

C. How Semantic Communications Benefit from Sensing?

Semantic communications aim to extract important information or features from the image or video and further transmit it instead of the whole image or video, greatly reducing the transmission overhead. To this end, most existing solutions leverage DL-based methods on the entire image. While this kind of approach has been proven to be effective, substantial computational resources are generally required. However, this can be prohibitive for devices with limited computational capabilities, resulting in high latency and congestion.

In fact, the most informative semantic content is often concentrated in a small region of a video frame. Fig. 3 gives an example frame captured by a classroom webcam. It can be observed that only a small portion (approximately 6%) of the image contains meaningful information relevant to pose estimation. Thus, applying computationally intensive DL-based methods to the entire frame incurs significant overhead. Unfortunately, existing approaches struggle to mitigate this inefficiency, as they rely solely on raw data (i.e., the captured frames) and can only identify the region of interest (RoI) after applying DL-based methods.

To address this, we aim to find an auxiliary information source with low cost, and the already available CSI at the BS emerges as a promising candidate. As indicated by the channel model in (2), the measured CSI inherently captures information related to the target within the surveillance area. Leveraging this property, we identify two key benefits of integrating sensing into semantic communications.

• **Estimating the coarse RoI.** Since the measured CSI at the BS contains the location information of the target, it can be leveraged to approximate the target's region within the video frame by applying localization algorithms on CSI and using the vision model in Section III-A.

• **Enhancing pose estimation.** Different parts of the human body interact with wireless signals differently, leading to variations in ToF, AoA, and signal strength at the BS. By analyzing and processing the measured CSI, it is possible to extract meaningful features related to body posture [43]. These features can then serve as complementary information to enhance pose estimation at the edge.

In the next section, we will give a detailed design for exploring these potential benefits.

IV. The Design of SenSem

In this section, we propose SenSem that leverages sensing to assist the semantic communications for reducing both computation and communication overhead and further improving the estimation performance. As illustrated in Fig. 4, it mainly consists of five steps.

CSI acquisition. The first step starts with transmitting pilots to the BS by the devices after collecting a GoP. With received pilots, the BS measures the CSI between the device and itself.

RoI estimation. The measured CSI is used for estimating the location of the target within the surveillance zone, enabling the determination of a coarse RoI for the GoP. The estimated RoI information is then fed back to the devices.

Semantic evaluation. With the received RoI, each device divides the RoI into several blocks and applies a semantic

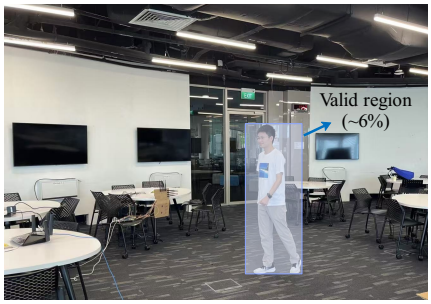


Fig. 3: A captured video frame.

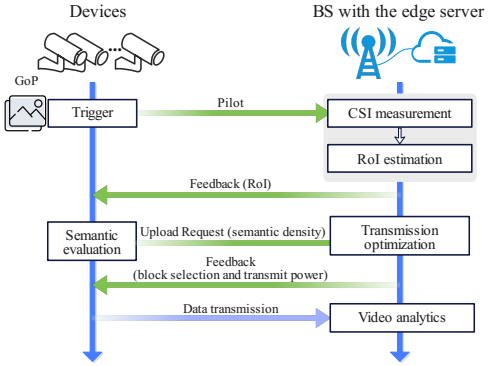


Fig. 4: The overview of SenSem.

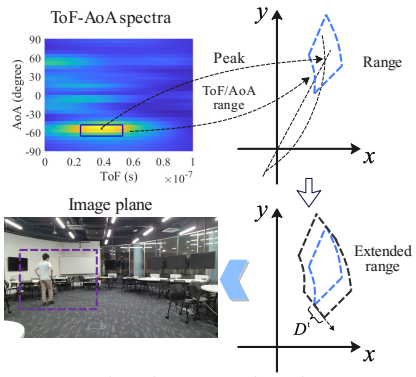


Fig. 5: RoI estimation.

evaluator to measure the semantic density of each block. The density is further uploaded to the BS for requesting subsequent transmission.

Optimization and transmission. After receiving all devices' transmission requests, the BS jointly optimizes block selection and transmit power to maximize the analytics performance under the delay requirement. Upon the optimization results, each device performs the transmission for the selected blocks.

Video analytics. The BS jointly utilizes the received semantic information and the CSI to realize the pose estimation.

Since the CSI acquisition in the first step can be easily realized using existing channel estimation methods designed for multi-user systems [44], we focus on presenting the detailed design for the remaining four steps in Sections IV-A to IV-D.

A. Location-Assisted RoI Estimation

In this section, we elaborate on how to use the measured CSI to calculate the coarse RoI. Before performing localization, we need to remove the LoS path from the measured CSI, as the LoS path generally has the highest signal strength and may interfere with the estimation of the NLoS path related to the target. Recall that the device is located at $(x_n^{\mathrm{D}}, y_n^{\mathrm{D}}, z_n^{\mathrm{D}})$ and the BS is located at $(0, 0, z^{\mathrm{B}})$ with the direction of the ULA being $[1, 0, 0]$. Thus, the AoA and ToF of the LoS path between the BS and the device n are fixed and can be calculated using the locations, as

$$\tau_{n,1} = \frac{\sqrt{(x_n^{\mathrm{D}})^2 + (y_n^{\mathrm{D}})^2 + (z_n^{\mathrm{D}} - z_n^{\mathrm{B}})^2}}{c}, \tag{6}$$

$$\theta_{n,1} = \arcsin\left(\frac{x_n^{\mathrm{D}}}{\sqrt{(x_n^{\mathrm{D}})^2 + (y_n^{\mathrm{D}})^2}}\right), \tag{7}$$

respectively, where c is the speed of light. However, the path attenuation of the LoS path is unknown. To obtain it, we adopt the least-squares method that aims to minimize the difference between the measured CSI and the LoS path, as

$$\min_{\hat{a}_{n,1}} \sum_{k=1}^{K} ||\boldsymbol{h}_{n,k} - \hat{a}_{n,1} e^{i2\pi\tau_{n,1}(f^{\mathrm{c}}+k\Delta f)} \boldsymbol{\alpha}(\theta_{n,1})||^2. \tag{8}$$

With simple mathematical analysis and calculations, the estimated path attenuation $\hat{a}_{n,1}$ can be derived as

$$\hat{a}_{n,1} = \frac{1}{MK} \sum_{k=1}^{K} e^{-i2\pi\tau_{n,1}(f^{\mathrm{c}}+k\Delta f)} \boldsymbol{\alpha}(\theta_{n,1})^H \boldsymbol{h}_{n,k}. \tag{9}$$

Thus, the CSI after removing the LoS path is

$$\boldsymbol{h}_{n,k}^{\mathrm{NLoS}} = \boldsymbol{h}_{n,k} - \hat{a}_{n,1} e^{i2\pi\tau_{n,1}(f^{\mathrm{c}}+k\Delta f)} \boldsymbol{\alpha}(\theta_{n,1}). \tag{10}$$

Upon the CSI in (10), the BS further utilizes the existing localization algorithms to determine the AoA and ToF related to the target, denoted by $\hat{\theta}_{n,2}$ and $\hat{\tau}_{n,2}$, respectively. In this paper, we adopt the algorithm proposed in [45] and can obtain the ToF-AoA spectra, as shown in Fig. 5. However, unlike the fine-grained image, due to the limited bandwidth and number of antennas, the location information obtained from wireless signals often contains non-negligible errors. Using this information to estimate the RoI may result in the loss of some posture details for the target. For example, hand joints might be excluded from the estimated RoI. To address this issue, we need to find the potential range of the target instead of only a location. Specifically, after finding the peak position $(\hat{\theta}_{n,2}, \hat{\tau}_{n,2})$ related to the target, we aim to find the AoA and ToF range of the peak by applying a threshold-based strategy. Let P^{p} and P^{n} denote the magnitude of the peak and the noise in the ToF-AoA spectra, and the threshold can be set as $P^{\mathrm{t}} = \eta(P^{\mathrm{p}} - P^{\mathrm{n}}) + P^{\mathrm{n}}$ with $\eta \in (0, 1)$ being the parameter. In the AoA dimension, we search outward from the peak in both directions until the spectral power falls below P^{t}, thereby determining the AoA range, denoted by $[\hat{\theta}_{n,2}^{\ell}, \hat{\theta}_{n,2}^{\mathrm{u}}]$. Similarly, a similar operation is performed in the ToF dimension to obtain the ToF range, denoted by $[\hat{\tau}_{n,2}^{\ell}, \hat{\tau}_{n,2}^{\mathrm{u}}]$.

Now, we need to transform the obtained ToF and AoA of the target into the coordinate in the world coordinate system. Specifically, given the locations of the BS and the device n, we have the following equations for the relationship among the location of the target, denoted by $\boldsymbol{p}_n^{\mathrm{t}} = [x_n^{\mathrm{t}}, y_n^{\mathrm{t}}, z_n^{\mathrm{t}}]^T$, the ToF, and AoA, as

$$\hat{\tau}_{n,2} c = \sqrt{(x_n^{\mathrm{t}} - x_n^{\mathrm{D}})^2 + (y_n^{\mathrm{t}} - y_n^{\mathrm{D}})^2 + (z_n^{\mathrm{t}} - z^{\mathrm{D}})^2} + \sqrt{(x_n^{\mathrm{t}})^2 + (y_n^{\mathrm{t}})^2 + (z_n^{\mathrm{t}} - z^{\mathrm{B}})^2}, \tag{11}$$

$$\sin(\hat{\theta}_{n,2}) = \frac{x_n^{\mathrm{t}}}{\sqrt{(x_n^{\mathrm{t}})^2 + (y_n^{\mathrm{t}})^2}}. \tag{12}$$

In the above, (11) represents an ellipsoid that the sum of the distances to the BS and device n is a constant, and (12) represents a plane with $x = y\sin(\theta_{n,2})$. Note that the height of the target z_n^{t} can be regarded as the prior knowledge since it generally remains the same for a long

time.[2] Under given z_n^{t}, (11) can be simplified to an ellipse of $a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6$, where coefficients $\{a_i, i = 1, \cdots, 6\}$ can be easily calculated from (11) and the detailed processes are omitted. Thus, $[x_n^{\mathrm{t}}, y_n^{\mathrm{t}}]$ can be regarded as the intersection of the ellipse and the line $(x = y\sin(\hat{\theta}_{n,2}))$.[3] Moreover, the intersection point can be obtained by substituting $x = y\sin(\hat{\theta}_{n,2})$ into the ellipse equation and then solving using the quadratic formula. Considering the ranges of the AoA and ToF ($[\hat{\theta}_{n,2}^{\ell}, \hat{\theta}_{n,2}^{\mathrm{u}}]$ and $[\hat{\tau}_{n,2}^{\ell}, \hat{\tau}_{n,2}^{\mathrm{u}}]$), the potential range of the target in the world coordinate system is defined by two elliptical curves and two straight lines, as shown in Fig. 5.

Thus far, the analysis has treated the target as a point. However, in reality, the target is a 3D object. To avoid information loss, it is necessary to further expand the potential range to include all body information of the object. Since the target's pose information is unknown at this stage, we consider an extreme case: the target extends its arms, and the line segment formed by the arms is perpendicular to the camera's optical axis. Let $2D^{\mathrm{t}}$ denote the length of the segment and its direction is $[\sin(\vartheta_y), \cos(\vartheta_y), 0]$. Thus, we extend the potential range by a distance D^{t} in the directions of $[\sin(\vartheta_y), \cos(\vartheta_y), 0]$ and $[-\sin(\vartheta_y), -\cos(\vartheta_y), 0]$, respectively, to obtain the new range. After that, the potential range with the height being z_n^{t} can be transformed to the region in the image plane using the vision model introduced in Section III-A. The estimated region may be irregular, which goes against the subsequent feedback and semantic extraction. Consequently, as shown in Fig. 5, we use the smallest rectangle that contains the estimated region as the final estimated RoI.

B. Low-Complexity Semantic Evaluator

The existing solution [32] relies on the complex neural network architecture to realize the video semantic information encoding and decoding, being computationally intensive. Thus, it may not be viable for surveillance devices with limited computation capability. To address this issue, inspired by [22], we first partition each frame in the GoP into B_n small blocks with the size being 32×32 and only select the blocks with key information for the applications at the edge using the middle frame in the GoP. Furthermore, to improve encoding efficiency, we directly adopt advanced video coding techniques, such as the H.264 standard [31], for encoding the selected blocks across the entire GoP. This approach fully leverages the dedicated encoding hardware already available in existing devices.

To realize this strategy, the key is to design a semantic evaluator to quantify the amount of semantic information in each block, referred to as semantic density. Since our goal is to select blocks effective for the application at the edge, we can measure the semantic information of a block based on the performance degradation caused by its absence. Directly calculating the degradation involves a significant computational burden, as it requires applying the estimation network at the edge for every frame variant, each time omitting a single block. To address this, following [22], we approximate the performance degradation as the gradient of performance with respect to this block, multiplied by the difference between the block's original value and its replacement value. Thus, for the frame at the n-th device, the semantic density of the b-th block can be measured as

$$\rho_{n,b} = \sum_{i \in \mathbb{P}_{n,b}} \left| \frac{\partial \mathcal{L}(y(x_i), y(C))}{\partial x_i}|_{x_i = H_i} \right| \cdot |H_i - C|, \quad (13)$$

where $\mathbb{P}_{n,b}$ denotes the set of pixels in the block, H_i denotes the i-pixel's original value, and C is a constant used at the edge to replace omitted blocks, i.e., the replacement value. Moreover, $y(\cdot)$ represents the estimation network at the edge and $\mathcal{L}(\cdot)$ is the loss function that describes the difference in output when different pixel values are used.[4] In (13), the term $\frac{\partial \mathcal{L}(y(x_i), y(C))}{\partial x_i}|_{x_i = H_i}$ represents the gradient at the original pixel value and $|H_i - C|$ represents the difference between the original pixel value and replacement value. Consequently, $\rho_{n,b}$ serves as an approximation of the performance degradation incurred when the b-th block is not transmitted to the BS.

Since the calculation of the semantic density in (13) also needs the network at the edge, it is not impractical to directly apply (13). Instead, we train a lightweight neural network as a semantic evaluator to measure the semantic density of each block efficiently. Considering the requirement of the low computational cost, we adopt MobileNetV2 [46] due to its superior performance on mobile devices. Moreover, note that the size of the estimated RoI does not stay constant due to the varying location and pose of the target, and thus, the input and output of the modified MobileNet must be adaptable to these changes. To achieve this adaptability, we expand the RoI to ensure that the input dimensions of the modified MobileNet are multiples of 32 in both width and height. The scaling ratio between the input and output of MobileNet is set to match the block size (i.e., 256 pixels), ensuring that each output corresponds to a single block. To realize it, we modify MobileNet by removing the fully connected layer, which would otherwise impose a fixed input/output size constraint. Additionally, we set the stride of the first convolutional layer and four inverted residual bottleneck layers to 2, enabling efficient downsampling while preserving key features. To train the modified MobileNet, we first perform the location-assisted RoI estimation on the collected frame and further calculate the semantic density using (13) and the pose estimation network. The computed semantic density serves as the ground truth for training MobileNet, with mean square error (MSE) employed as the loss function to optimize the network.

For the computational complexity, the proposed semantic evaluator consists of one initial convolutional layer, followed by seven inverted residual blocks, and a final convolutional

[2]In cases where the target height fluctuates (e.g., raising hands or bending), a conservative upper bound is typically sufficient, while more adaptive methods (e.g., elevation estimation with a 2D antenna array or video-based feedback) could also be applied.

[3]There generally would be two intersections between an ellipse and a line, and we focus on introducing one intersection with the other one dealt with the same manner.

[4]Note that the detailed network structure for pose estimation will be introduced in Section IV-D.

layer. For the l-th (convolutional) layers with kernel size c_l^{k}, input channels C_l^{i}, output channels C_l^{o}, and output spatial resolution (H_l^{o}, W_l^{o}), the computational complexity can be approximated as $\mathcal{O}(H_l^{\mathrm{o}}W_l^{\mathrm{o}}c_1^2C_l^{\mathrm{i}}C_l^{\mathrm{o}})$. For the l-th (inverted residual) layer with expansion ratio t_l^{i}, the computational complexity is given by $\mathcal{O}(H_l^{\mathrm{o}}W_l^{\mathrm{o}}(c^2t_l^{\mathrm{i}}C_l^{\mathrm{i}}+t_l^{\mathrm{i}}(C_l^{\mathrm{i}})^2+t_l^{\mathrm{i}}C_l^{\mathrm{i}}C_l^{\mathrm{o}}))$. Aggregating across all layers, the total computational complexity becomes $\mathcal{O}(\sum_{l=2}^{8} H_l^{\mathrm{o}}W_l^{\mathrm{o}}(c^2t_l^{\mathrm{i}}C_l^{\mathrm{i}}+t_l^{\mathrm{i}}(C_l^{\mathrm{i}})^2+t_l^{\mathrm{i}}C_l^{\mathrm{i}}C_l^{\mathrm{o}})+\sum_{l\in\{1,9\}} H_l^{\mathrm{o}}W_l^{\mathrm{o}}c_l^2C_l^{\mathrm{i}}C_l^{\mathrm{o}})$. In addition, the model size is only about 0.1 M parameters, and for an input of 500×500 pixels, the FLOPs are approximately 650 M. All the above clearly demonstrate the low computational complexity of the proposed semantic evaluator.

C. Analytics Performance Maximization

After obtaining the semantic density of each block, each user uploads the semantic density $\rho_{n,b}$ of each block, and then the BS needs to determine the block selection and transmit power control for maximizing the performance of the video analytics under the limitations of the delay and communication resource. Let $\delta_{n,b}$ denote the selection indicator of the b-th block for the n-th device, where $\delta_{n,b}=1$ indicates that the block is selected, and $\delta_{n,b}=0$ indicates that it would not be transmitted. With the n-th device's data rate R_n given in (5), the transmission latency can be calculated as

$$T_n=\sum_{b=1}^{B_n}\delta_{n,b}V/R_n. \tag{14}$$

where V denotes the average data volume for each block.

In this paper, we aim to maximize the performance of video analytics at the edge. Based on the analysis in Section IV-B, $\rho_{n,b}$ describes the performance degradation when the n-th block is not uploaded. Therefore, the performance of the n-th device can be approximately expressed as

$$A_n=\sum_{b=1}^{B_n}\delta_{n,b}\rho_{n,b}. \tag{15}$$

Then, we can further formulate the following performance maximization problem with joint block selection and power control under the latency requirement:

$$\max_{\left\{\substack{\delta_{n,b},\tau_j,\\ p_{n,k}^{\mathrm{T}},\boldsymbol{w}_{n,k}}\right\}} \sum_{n=1}^{N}\beta_n\left(\sum_{b=1}^{B_n}\delta_{n,b}\rho_{n,b}\right), \tag{16a}$$

$$\text{s.t.}\quad T_n=\frac{\sum_{b=1}^{B_n}\delta_{n,b}V}{R_n}\le\tau_j,\ \forall n\in\mathbb{N}_j,\forall j, \tag{16b}$$

$$\sum_{j=1}^{J}\tau_j\le T^{\mathrm{c}}, \tag{16c}$$

$$\delta_{n,b}\in\{0,1\},0\le p_{n,k}^{\mathrm{T}}\le p^{\max},\forall n,b,k, \tag{16d}$$

where $\beta_n>0$ is the weight of the n-th device, $p^{\max}$ is the upper limit for the transmit power, τ_j is the allocated time for the j-th group, and T^{c} is the transmission latency requirement, determined by the total latency requirement $T^{\max}$ and the computation latency of the semantic evaluator. Moreover, (16b) guarantees that the latency of each device in one group does not exceed the allocated time τ_j, and (16c) ensures the total transmission latency does not exceed its requirement.

By observing problem (16), we can find that the difficulty of solving the problem primarily lies in the integer variables $\delta_{n,b}$ and the complex inter-user interference. To address the first difficulty, we note that, under the same transmission volume, blocks with higher semantic density should be prioritized for transmission. Therefore, we can first sort the blocks for each device in descending order of semantic density, transforming the block selection problem into an optimization of the number of blocks. Let B_n^{s} represent the number of selected blocks, and let the function $g_n(B_n^{\mathrm{s}})$ represent the amounts of the semantic density of the B_n^{s} selected blocks. We can further relax B_n^{s} into a continuous variable since B_n is typically large, and $g_n(B_n^{\mathrm{s}})$ can be fitted as a piecewise linear function. It is easy to find that $g_n(B_n^{\mathrm{s}})$ is concave and monotonically increasing. Meanwhile, with the relaxed B_n^{s}, "=" in constraint (16b) should be satisfied to maximize the objective function and the optimal B_n^{s} is given by

$$B_n^{\mathrm{s},\star}=\tau_jR_n/V,\ n\in\mathbb{N}_j. \tag{17}$$

Then, the optimization problem can be rewritten as

$$\begin{aligned}\max_{\{\tau_j,p_n^{\mathrm{T}},\boldsymbol{w}_{n,k}\}}\quad & \sum_{n=1}^{N}\beta_ng_n(\tau_jR_n/V), \\ \text{s.t.}\quad & \text{(16c) and (16d).}\end{aligned} \tag{18}$$

Regarding problem (18), the transmit power of different devices is still coupled in the data rate expression and $g_n(\cdot)$ in the objective function does not have an explicit functional form, causing the problem difficult to solve directly. To address it, we apply the WMMSE algorithm [47] and give the following lemma.

Lemma 1. *Defining* $(v_{n,k}^{\mathrm{T}})^2=p_{n,k}^{\mathrm{T}}$ *with* $v_{n,k}^{\mathrm{T}}\in\mathbb{R}$, *problem* (18) *has the same optimal solution with the following problem:*

$$\begin{aligned}\min_{\left\{\substack{\kappa_{n,b},\tau_j,\\ v_{n,k}^{\mathrm{T}},\boldsymbol{w}_{n,k}}\right\}}\quad & \sum_{n=1}^{N}\beta_nG_n\left(\tau_j\sum_{k=1}^{K}(\ln\kappa_{n,k}-\kappa_{n,k}e_{n,k}+1)\right), \\ \text{s.t.}\quad & \text{(16c) and } (v_{n,k}^{\mathrm{T}})^2\le p^{\max},\ \forall n,b,\end{aligned} \tag{19}$$

where $\kappa_{n,b}>0$ *is an auxiliary weight variable,* $G_n(x)=-g_n\left(\frac{\Delta f}{V\ln 2}x\right)$, *and* $e_{n,k}$ *is the mean-square estimation error at the* k*-th subcarrier:*

$$\begin{aligned}e_{n,k}=&|\boldsymbol{w}_{k,n}^H\boldsymbol{h}_{n,k}v_{n,k}^{\mathrm{T}}-1|^2+\sigma^2\boldsymbol{w}_{k,n}^H\boldsymbol{w}_{k,n}\\&+\sum_{n'\in\mathbb{N}_j\setminus\{n\}}|\boldsymbol{w}_{n,k}^H\boldsymbol{h}_{n',k}v_{n',k}^{\mathrm{T}}|^2.\end{aligned} \tag{20}$$

Proof: The proof can be found in [47] and the detailed derivation is omitted for brevity. ∎

With Lemma 1, we can apply the block coordinate descent (BCD) method [48] to solve problem (19). It divides the variables into several blocks and successively updates them by solving the corresponding subproblems. Specifically, for problem (18), its variables can be divided into four blocks: $\{\kappa_{n,k},\forall n,k\}$, $\{\boldsymbol{w}_{n,k},\forall n,k\}$, $\{\tau_j,\forall j\}$, and $\{v_{n,k}^{\mathrm{T}},\forall n\}$, and

the corresponding subproblems are solved in the following.

First of all, we optimize $\{\kappa_{n,k}, \forall n, k\}$. Since $\{\kappa_{n,k}\}$ are independent for different n and k, they can be optimized in parallel. For $\kappa_{n,k}$, the subproblem is to maximize $\ln \kappa_{n,k} - \kappa_{n,k} e_{n,k}$ as $G_n(x)$ is a monotonically decreasing function. Consequently, the optimal $\kappa_{n,k}$ can be easily derived as

$$\kappa_{n,k}^{\star} = e_{n,k}^{-1}. \tag{21}$$

Similarly, $\{\boldsymbol{w}_{n,k}, \forall n, k\}$ can be optimized in parallel, and the subproblem for $\boldsymbol{w}_{n,k}$ is to minimize $e_{n,k}$. The optimal $\boldsymbol{w}_{n,k}$ can be easily derive as

$$\boldsymbol{w}_{n,k}^{\star} = \left(\sum_{n' \in \mathbb{N}_j} \boldsymbol{h}_{n',k} \boldsymbol{h}_{n',k}^H |v_{n',k}|^2 + \sigma^2 \boldsymbol{I} \right)^{-1} \boldsymbol{h}_{n,k} v_{n,k}. \tag{22}$$

For optimizing $\{\tau_j, \forall j\}$, the subproblem is convex since $G_n(\cdot)$ is convex and constraint (16c) is linear. Thus, we can apply the Lagrangian method to solve it with the partial Lagrange function being

$$\mathcal{L} = \sum_{n=1}^{N} \beta_n G_n(\tau_j b_n) + \lambda \left(\sum_{j=1}^{J} \tau_j - T^{\max} \right). \tag{23}$$

where $b_n = \sum_{k=1}^{K} (\ln \kappa_{n,k} - \kappa_{n,k} e_{n,k} + 1)$ and λ is the Lagrange multiplier associated with constraint (16c). By applying the Karush-Kuhn-Tucker (KKT) conditions, the optimal solution, denoted by $\tau_j^{\star}$, is given by the following Theorem.

Theorem 1. *The optimal communication resource allocation strategy can be expressed as*

$$\tau_j^{\star} = \left(\Phi_j(-\lambda^{\star}) \right)^+, \tag{24}$$

where $\Phi_j(x)$ *is the inverse function of* $\sum_{n \in \mathbb{N}_j} \beta_n b_n \frac{\partial G_n(x)}{\partial x}|_{x=\tau b_n}$ *with respect to* τ*, i.e.,* $\Phi_j \left(\sum_{n \in \mathbb{N}_j} \beta_n b_n \frac{\partial G_n(x)}{\partial x}|_{x=\tau b_n} \right) = \tau$, $(x)^+ = \max\{x, 0\}$*, and* $\lambda^{\star}$ *is the optimal Lagrange multiplier satisfying constraint* (16c)*.*

Proof: Please refer to Appendix A of the supplemental material. ∎

With Theorem 1, we can apply the classical bisection search algorithm until constraint (16c) is satisfied to determine $\lambda^{\star}$. For the inverse function $\Phi_j(x)$, it is hard to obtain its explicit expression, and we also apply the bisection search algorithm to calculate (24).

For the last block $\{v_{n,k}^{\mathrm{T}}, \forall n\}$, variables in the block are optimized sequentially. The subproblem with respect to $v_{n,k}^{\mathrm{T}}$ is given by

$$\begin{aligned} \min_{v_{n,k}^{\mathrm{T}}} \quad & \sum_{n' \in \mathbb{N}_j \setminus \{n\}} \beta_{n'} G_{n'} \left(-\tau_j \kappa_{n',k} |\boldsymbol{w}_{n',k}^H \boldsymbol{h}_{n,k} v_{n,k}^{\mathrm{T}}|^2 + B_{n',k} \right) \\ & + \beta_n G_n \left(-\tau_j \kappa_{n,k} |\boldsymbol{w}_{n,k}^H \boldsymbol{h}_{n,k} v_{n,k}^{\mathrm{T}} - 1|^2 + A_{n,k} \right), \quad (25a) \\ \text{s.t.} \quad & -\sqrt{p^{\max}} \le v_{n,k}^{\mathrm{T}} \le \sqrt{p^{\max}}, \quad (25b) \end{aligned}$$

where $A_{n,k}$ and $B_{n',k}$ are known values that are not related to $v_{n,k}^{\mathrm{T}}$. The detailed expressions for $A_{n,k}$ and $B_{n',k}$ can be easily derived through (19) but are omitted here due to their complexity. Before solving the subproblem for $v_{n,k}^{\mathrm{T}}$, we have the following lemma.

Algorithm 1: Joint Block Selection and Transmit Power Control Algorithm for Problem (16).

1 Initialize the maximal error tolerance $\epsilon > 0$ and the maximal number of iterations $I^{\max}$;
2 **for** $n = 1, 2, \cdots, N$ **do**
3 Sort the blocks in descending order of semantic density and structure $g_n(B_n^{\mathrm{s}})$;
4 **end**
5 Initialize variables $\{\kappa_{n,k}, \boldsymbol{w}_{n,k}, \tau_j, v_{n,k}^{\mathrm{T}}, \forall n, k, j\}$ and set the iteration number $j = 0$;
6 **repeat**
7 Update $\{\kappa_{n,k}, \forall n, k\}$ according to (21);
8 Update $\{\boldsymbol{w}_{n,k}, \forall n, k\}$ according to (22);
9 Update $\{\tau_j, \forall j\}$ using the bisection search algorithm according to Theorem 1;
10 Update $\{v_{n,k}^{\mathrm{T}}, \forall n, k\}$ using the Golden-section search;
11 $i = i + 1$;
12 **until** *the difference between consecutive values of the objective function is under* ϵ or $i > I^{\max}$.
13 **Output:** $B_n^{\mathrm{s}} = \lfloor \tau_j R_n / V \rfloor, p_{n,k}^{\mathrm{T}} = (v_{n,k}^{\mathrm{T}})^2, \boldsymbol{w}_{n,k}, \forall n, k.$

Lemma 2. *The objective function in problem* (25) *is convex and unimodal.*

Proof: Please refer to Appendix B of the supplemental material. ∎

With Lemma 2, we could also apply the Lagrangian method to find the optimal $v_{n,k}^{\mathrm{T}}$. However, due to the undefined expression of $G_n(x)$, directly solving problem (25) using derivatives is relatively complex. As an alternative, we make use of the unimodality of the objective function, which means that there is only one peak. The Golden-section search algorithm [49] can be employed to find the optimal $v_{n,k}^{\mathrm{T}}$.

So far, we have solved all subproblems for the four blocks, and the overall algorithm to problem (16) is summarized in Algorithm 1, which contains two parts: block sorting and iterative optimization. In the first part, the algorithm needs to sort the frame block for each device, and the computational complexity is $\mathcal{O}(\sum_n B_n \log B_n)$. In the second part, the algorithm needs to successively solve the four subproblems for the four variable blocks in each iteration. Specifically, the computational complexities of steps 7 and 8 are $\mathcal{O}(NKM \max_j |\mathbb{N}_j|)$ and $\mathcal{O}(K(JM + N)M^2)$, respectively. The computational complexity of step 9 is $\mathcal{O}\left(N(\log \frac{1}{\epsilon})^2\right)$ with ϵ being the maximal error tolerance, since step 9 requires a two-layer bisection search. Moreover, the computational complexity of step 10 is $\mathcal{O}\left(NK \log \frac{1}{\epsilon}\right)$. Given $I^{\max}$ being the maximal number of iterations, the computational complexity of Algorithm 1 is

$$\mathcal{O}\left(\sum_n B_n \log B_n + I^{\max} \left(KMN(\max_j |\mathbb{N}_j| + M) \right) \right.$$

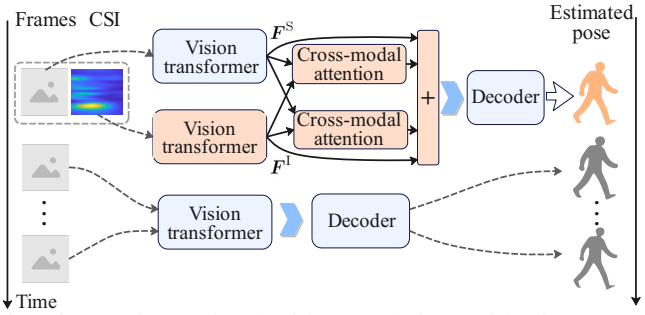


Fig. 6: Sensing-assisted video analytics, with the orange-colored components representing the introduced structures.

$$+I^{\max}\left(KJM^3+N\log\frac{1}{\epsilon}(\log\frac{1}{\epsilon}+K)\right)\Big). \quad (26)$$

Furthermore, the convergence of the proposed algorithm is guaranteed since the BCD method is adopted [48].

D. Sensing-Assisted Video Analytics

Upon the compressed data, the BS first reconstructs the GoP and then performs the pose estimation. As we mentioned in Section III-C, the measured CSI at the BS also contains the pose information related to the target, and thus it can be used in video analytics to improve the performance. According to the procedure shown in Fig. 4, there is only one measured CSI sample available for each GoP. Directly applying the same CSI sample to all frames within the GoP may degrade performance rather than improve it. As an alternative, we assign the CSI sample to the frame with the closest timestamp for enhancing analysis, while the remaining frames are directly processed using a state-of-the-art (SOTA) solution, i.e., ViTPose [23], as depicted in Fig. 6. Subsequently, the results from the CSI-assisted frame are leveraged to refine the pose estimations of the other frames by exploiting temporal correlations, such as through the HoT framework [50]. In this section, we focus on detailing how CSI is used to assist in the pose estimation of the closest frame.

To adapt to the existing solution as much as possible, we consider extending the structure based on ViTPose. We continue using the ToF-AoA spectrum from Section IV-A as input, as it contains spatial information about the target. However, because CSI is collected at the BS while frames are captured at the device, they cannot be directly merged as input. Instead, we adopt feature-level data fusion, as shown in Fig. 6. Specifically, in the original ViTPose, the vision transformer [51] is used to extract features from video frames, which are then fed into a decoder for realizing the pose estimation. Following this approach, we also utilize the vision transformer to extract features from the ToF-AoA spectrum. To enable deep fusion, we adopt a cross-modal attention (CA) mechanism to combine CSI and frame features effectively. Let $\boldsymbol{F}^{\mathrm{S}} \in \mathbb{R}^{L_F \times N_F}$ and $\boldsymbol{F}^{\mathrm{I}} \in \mathbb{R}^{L_F \times N_F}$ denote the extracted features from the CSI and video frame, respectively, after applying the vision transformer. We then project these features into query ($\boldsymbol{Q}$), key ($\boldsymbol{K}$), and value ($\boldsymbol{V}$) representations using trainable linear transformations:

$$\boldsymbol{Q}=\boldsymbol{F}^{\mathrm{I}}\boldsymbol{W}^{\mathrm{Q}},\ \boldsymbol{K}=\boldsymbol{F}^{\mathrm{S}}\boldsymbol{W}^{\mathrm{K}},\ \boldsymbol{K}=\boldsymbol{F}^{\mathrm{S}}\boldsymbol{W}^{\mathrm{V}}, \quad (27)$$

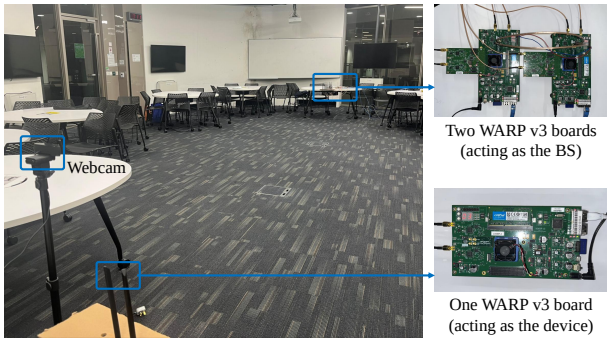


Fig. 7: Experiment setup in a room.

where $\boldsymbol{W}^{\mathrm{Q}} \in \mathbb{R}^{N_F \times D_F}$, $\boldsymbol{W}^{\mathrm{K}} \in \mathbb{R}^{N_F \times D_F}$, $\boldsymbol{W}^{\mathrm{V}} \in \mathbb{R}^{N_F \times D_F}$ are trainable weight matrices. With these representations, the cross-modal attention is formulated as

$$\mathrm{CA}(\boldsymbol{F}^{\mathrm{I}},\boldsymbol{F}^{\mathrm{S}})=\mathrm{Softmax}\left(\frac{\boldsymbol{F}^{\mathrm{I}}\boldsymbol{W}^{\mathrm{Q}}(\boldsymbol{F}^{\mathrm{S}}\boldsymbol{W}^{\mathrm{K}})^T}{\sqrt{D_F}}\right)\boldsymbol{F}^{\mathrm{S}}\boldsymbol{W}^{\mathrm{V}}, \quad (28)$$

where $\mathrm{Softmax}(\cdot)$ is the Softmax function. This operation computes the relationship between the CSI feature $\boldsymbol{F}^{\mathrm{S}}$ and frame feature $\boldsymbol{F}^{\mathrm{I}}$, allowing frame queries to focus on relevant CSI information. Similarly, $\mathrm{CA}(\boldsymbol{F}^{\mathrm{I}},\boldsymbol{F}^{\mathrm{S}})$ allows CSI queries to focus on relevant frame information.

To capture diverse cross-modal interactions, we employ multi-head cross-modal attention: $\mathrm{MHCA}(\boldsymbol{F}^{\mathrm{I}},\boldsymbol{F}^{\mathrm{S}}) = \mathrm{concat}\left(\mathrm{CA}(\boldsymbol{F}^{\mathrm{I}},\boldsymbol{F}^{\mathrm{S}}),\cdots,\mathrm{CA}(\boldsymbol{F}^{\mathrm{I}},\boldsymbol{F}^{\mathrm{S}})\right)\boldsymbol{W}^{\mathrm{M}}$, where $\boldsymbol{W}^{\mathrm{M}}$ is a trainable projection matrix and $\mathrm{concat}(\cdot)$ is the concatenation operation. To preserve modality-specific information, we then apply a residual connection to obtain the fused feature:

$$\boldsymbol{F}^{\mathrm{out}}=\boldsymbol{F}^{\mathrm{S}}+\mathrm{MHCA}(\boldsymbol{F}^{\mathrm{I}},\boldsymbol{F}^{\mathrm{S}})+\boldsymbol{F}^{\mathrm{I}}+\mathrm{MHCA}(\boldsymbol{F}^{\mathrm{S}},\boldsymbol{F}^{\mathrm{I}}).$$

The final fused feature is further input into the subsequent decoder for pose estimation. For training, we follow a strategy similar to that of ViTPose. Moreover, in terms of computational complexity, the proposed network incurs roughly twice the cost due to the additional vision transformer for CSI. This overhead is acceptable since the computation is performed on the edge server, where resources are generally abundant.

V. Evaluation Results

In this section, we first introduce the evaluation setup and then detail the results for verifying SenSem's performance.

A. Evaluation Setup

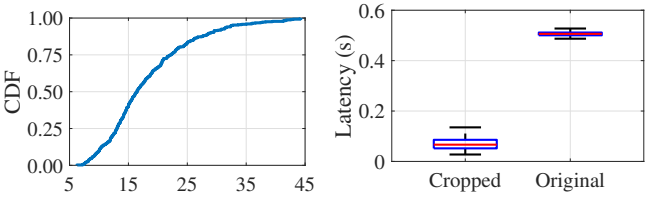
To ensure that the CSI corresponds accurately to the video data, we first conduct real-world experiments to collect a dataset, as illustrated in Fig. 7. A 1080p webcam is used to capture surveillance video, while three WARP v3 boards [52] are employed to collect the CSI. Specifically, two WARP boards are acted as the BS with 8 antennas, and one WARP board is acted as the device with 2 antennas. The three WARP v3 boards operate at the 5.16 GHz with the bandwidth being 20 MHz. The whole bandwidth is further divided into 64 subcarriers, with 52 subcarriers used for sensing. To enhance dataset diversity, the equipment setup is deployed in three

different locations, where three subjects are instructed to move randomly and perform various actions in the monitored area. The dataset consists of approximately 296,000 video frames and 735,000 CSI samples. To obtain the label for the pose estimation, we employ the ViTPose-H [23], one of the SOTA solutions. The dataset is then split into training and test sets at an 8:2 ratio. In our experiments, both the WARP platform and the video camera are connected to the same laptop, ensuring high-precision synchronization between CSI and video frames. In practical 5G systems, standard synchronization mechanisms with sub-microsecond accuracy [53] are sufficient to support SenSem. Moreover, our experiments strictly follow the Institutional Review Board guidelines of our institute.

With the collected dataset, we first train the semantic evaluator in Section IV-B and the video analytics network in Section IV-D over the training set on a Linux server with four RTX A5000 GPUs. Notably, to highlight the performance improvement brought by sensing, we select the ViTPose-B network as the foundation for the video analytics network instead of the more powerful ViTPose-H network. Then, we verify the performance of the proposal with the simulation, and the settings are provided as follows, unless otherwise specified. The system comprises one 8-antenna BS and 32 single-antenna devices. For each device, we randomly select five consecutive frames from the test set as the video to be processed, with one corresponding CSI sample used as the channel between the device and the BS. The latency requirement $T^{\max}$ is set as 150 ms, and the weight is set as $1/N$ for all devices. Moreover, on each device, the trained semantic evaluator is applied to measure the semantic density of each block after cropping each frame using the location-assisted RoI estimation method. For the transmission process, we consider that each group consists of 4 devices, and the transmit power upper limit of each device is set as 24 dBm for the whole bandwidth, with the limit for each subcarrier being 6.84 dBm. The ratio of the path loss to the noise power, i.e., loss-to-noise ratio (LNR), is randomly set within the range of [10,20] dB. At the edge server, we apply the trained video analytics network for pose estimation. Moreover, the performance of video analytics is evaluated using average precision (AP), which measures precision under a given threshold [23]. For example, AP50 corresponds to a threshold of 0.5 and mAP is the mean AP over thresholds of $\{0.5, 0.55, \cdots, 0.95\}$.

B. Micro-Benchmark Studies

First of all, we evaluate the effectiveness of the proposed location-assisted RoI estimation method. Fig. 8(a) illustrates the cumulative distribution function (CDF) for the ratio of the retained region after cropping. It can be seen that the average retained ratio is 18.29 %, corresponding to 37,931 pixels per frame. This result confirms that our proposed method significantly reduces the video volume. Next, we analyze its impact on the computation latency of the semantic evaluator. The semantic evaluator is deployed on a Raspberry Pi 5 with 4 GB RAM, and we measure the computation latency for both the original and cropped frames, as shown in Fig. 8(b). The average computation latency is around 71 ms after cropping



(a) CDF for ratio. (b) Latency.

Fig. 8: Performance evaluation of the proposed RoI estimation method: (a) CDF of the retained region ratio after cropping and (b) computation latency of the semantic evaluator for original and cropped frames.

TABLE I: Time of Existing Methods on Raspberry Pi 5.

Method	DeepWiVe [7]		DCVC [54]		YOLO [55]
Resolution	720p	1080p	720p	1080p	1080p
Delay (s)	76.4	192.5	51.4	120.8	2.6
Method	ADJSCC [56]		Lightweight solution [57]		
Resolution	720p	1080p	480p	720p	1080p
Delay (s)	7.5	17.5	1.9	5.7	13.2

TABLE II: Video Analytics Using the Cropped Frames.

Metrics	mAP	AP50	AP70	AP90
ViTPose-B	93.5%	100.0%	99.3%	82.2%
Sensing-assisted	95.1%	100.0%	99.7%	89.2%

while the average latency is around 507 ms for the original frames.[5] This reduction corresponds to only 14.1 % of the original latency, significantly decreasing computational overhead. The improvement is attributed to the positive correlation between the complexity of convolutional networks and the number of processed pixels. These results demonstrate the efficiency of our proposed method. Moreover, we also evaluate the inference latency of four representative existing semantic encoders as well as YOLOv11 (used for RoI estimation) on the Raspberry Pi, as shown in Tab. I. The results clearly show that all of them exceed the latency constraint of 150 ms. In addition, large-model solutions such as SegGPT [58] cannot be executed on the Raspberry Pi due to memory constraints. Even when deployed on a high-end server, their inference latency remains far beyond the real-time requirements. This indicates that current solutions are challenging to deploy on resource-constrained edge devices, highlighting the need for SenSem.

To verify that no key pose information is lost after cropping, we apply the ViTPose-B network to perform pose estimation on the cropped frames, and the results are summarized in Tab. II. One can clearly see that the performance after cropping remains high, confirming that our method preserves essential pose information. Additionally, we evaluate the sensing-assisted analytics network proposed in Section IV-D and report the results in Tab. II. Compared to the ViTPose-B network, the sensing-assisted network achieves a 2.6 % higher mAP, further validating the effectiveness of incorporating sensing information into video analytics.

[5]The computation latency could be further reduced via model pruning and quantization. However, as our focus is on demonstrating the reduction in overhead, the specific details are omitted in this paper.

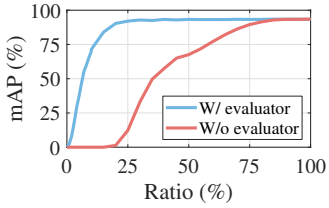


Fig. 9: Analytics performance with/without the evaluator.

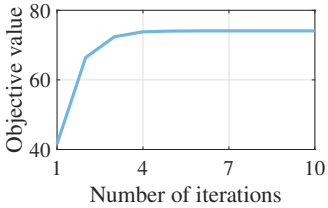


Fig. 10: Convergence behavior of Algorithm 1.

Next, we focus on the semantic evaluator on the cropped frames. To assess its effectiveness, we compute the importance (i.e., semantic density) of each block using the semantic evaluator and select high-importance blocks as input to the ViTPose-B network. As a baseline, we also consider an intuitionistic selection scheme, where blocks are chosen in a fixed order from the top-left to the bottom-right of the frame without considering importance. The performance of both methods under different selection ratios is presented in Fig. 9. We can observe that, with the semantic evaluator, mAP initially increases rapidly with the selection ratio and then stabilizes, following a strictly concave trend. This validates that the semantic density computed by the evaluator accurately reflects the importance of each block, i.e., its semantic information content. Notably, when the selected ratio reaches 30 %, the mAP achieves 92.8 %, which is comparable to the 93.5 % obtained when using the entire frame. In contrast, the baseline scheme exhibits slower performance growth, requiring a selection ratio of approximately 83 % to reach a similar performance level. This highlights that the use of the evaluator can significantly reduce the transmission overhead while maintaining high performance.

Finally, we analyze the convergence behavior of Algorithm 1. Fig. 10 depicts the achieved objective value with the number of iterations. It can be seen that the objective value monotonically increases with the number of iterations, and the proposed algorithm converges within 7 iterations. This demonstrates the convergence of the proposed algorithm. Additionally, the final objective value significantly outperforms that obtained from random initialization, demonstrating the superiority of the proposed approach. It is important to note that the objective value does not directly correspond to final analytics performance; thus, we will provide a performance evaluation against baselines in the next section to showcase the true performance gains.

C. Overall Performance

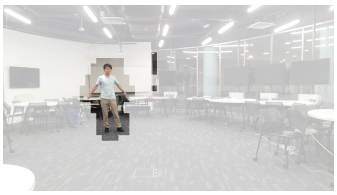
In this section, we evaluate the overall performance of the proposed SenSem. Based on Fig. 8(b), the computation latency is 71 ms, and thus, the upper limit for the communication latency T^{c} is set as 79 ms for Algorithm 1. It is due to that the remained communication and computation overhead introduced by the proposed method is negligible. Pilot transmission is reused without additional cost, while RoI feedback only requires sending a few numeric values, resulting in minimal data exchange. The optimization algorithm features polynomial complexity and converges rapidly, ensuring real-time execution with negligible latency.

First of all, we evaluate SenSem under a single-device scenario with bandwidth being 625 kHz and compare it with the lightweight solution [57]. To meet the latency constraint, we progressively reduced the input resolution until the total computation plus communication delay was within the required limit. The resulting input resolutions are 108×192 for the 150 ms case and 292×500 for the 1 s case. Fig. 12(a) shows the performance of SenSem and [57], with latency budgets of 150 ms and 1 s. The results indicate that our proposed method consistently outperforms [57], whose mAP remains close to zero across all SNR levels. To further analyze this result, Fig. 11 presents visual comparisons at SNR=30 dB, including the original image, the resized input, and the reconstructed image. Based on these comparisons, two direct observations can be made, and an additional contributing factor can be reasonably inferred. First, as illustrated by Fig. 11(a) vs. Fig. 11(c), the aggressive downsampling required to meet the latency constraint removes most fine-grained human-pose details, resulting in substantial semantic loss. Second, as shown in Fig. 11(c) vs. Fig. 11(d), the lightweight network design of [57] further limits reconstruction fidelity, introducing artifacts and blurring that obscure body features. In addition, the surveillance scenario itself reinforces this effect: the background is nearly static while the person moves continuously, making the model more likely to learn the stable background than the highly dynamic human region. Together, these factors lead to the noticeably blurred appearance of the person. In contrast, as shown in Fig. 11(b), SenSem effectively identifies and transmits only the blocks containing human information, thereby preserving key semantic content while drastically reducing communication load. These results confirm the superior efficiency and robustness of the proposed approach compared with SOTA lightweight baselines.

Next, we consider a multi-device case. Since existing semantic communication schemes applied directly to original frames often exceed the total latency limit, such methods are not included in this comparison. We compare SenSem against the following schemes:

• **Conventional scheme.** This scheme uses an H.264 pipeline without sensing or semantic evaluator. The frame is divided

(a) Original frame (1080×1920).

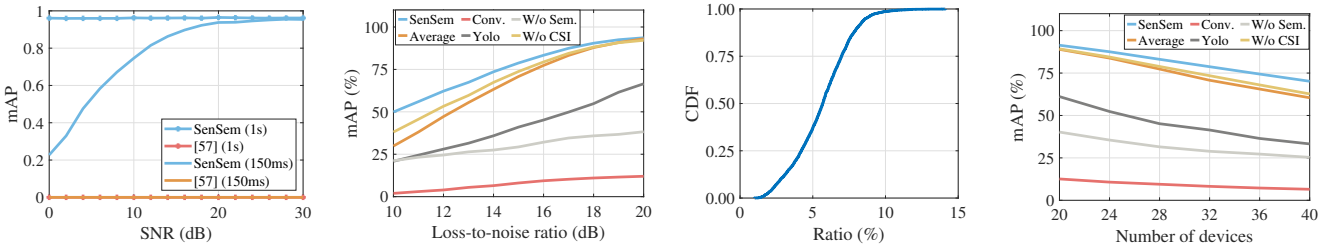


(b) SenSem.

(c) Input frame (108×192).

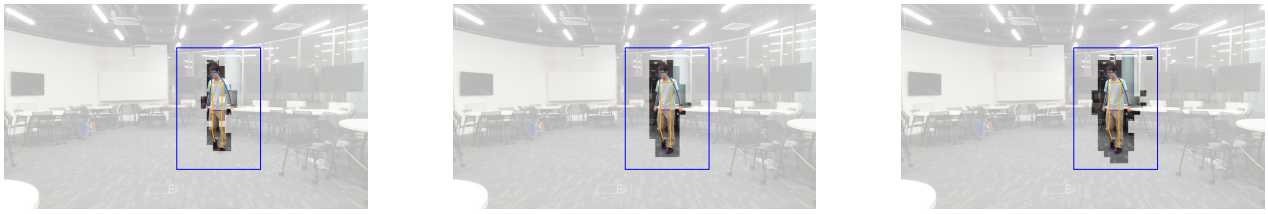
(d) Output frame (108×192).

Fig. 11: Visual comparison between SenSem and [57] at SNR=30 dB: (a) the original image, (b) the blocks uploaded by SenSem, (c)–(d) the input and reconstructed outputs of [57].



(a) Single-device case. (b) LNR. (c) Transmission ratio (20 dB). (d) Number of devices.

Fig. 12: The overall performance of the proposed SenSem: (a) single-device case, and (b)–(d) multi-device case.



(a) LNR=10 dB. (b) LNR=15 dB. (c) LNR=20 dB.

Fig. 13: Visualization of uploaded blocks under different LNR conditions. The blue box indicates the RoI estimated based on localization, the normally colored regions represent the transmitted blocks, and the gray regions denote discarded blocks.

into several blocks, each assumed to have identical semantic density. To meet the communication latency of 150 ms, Algorithm 1 is used to determine the number of selected blocks.

• **YOLO-based scheme.** Each device employs YOLO to detect the RoI. To satisfy the latency requirement, the input resolution of YOLO is set to 192×288, resulting in an inference delay of approximately 88.5 ms. As no semantic evaluator is used, all blocks within the detected RoI are assumed to have identical semantic importance. The number of uploaded blocks is determined by Algorithm 1 to meet the delay constraint.

• **Average scheme.** This scheme differs from SenSem in that it does not employ Algorithm 1. Instead, the transmit power is set as its upper limit, and the number of selected blocks is the same for all devices, which can be determined by applying the binary search until the communication latency is met.

• **Without semantic evaluator ("W/o Sem.").** The transmission latency is set to 150 ms, and all blocks are treated as having equal semantic density during optimization.

• **Without sensing-assisted video analytics ("W/o CSI").** The analytics at the server rely solely on images without CSI.

Firstly, Fig. 12(b) shows the performance of the three schemes with different LNRs. One can clearly observe that the proposed SenSem outperforms all baselines, demonstrating its superiority. First, the performance gap between the proposed scheme and the average scheme is more pronounced at a lower LNR. This is because, at a higher LNR ratio, most blocks with high semantic density can be uploaded, diminishing the relative performance gain. This result verifies the effectiveness of the Algorithm 1. Secondly, it can be observed that the performance of SenSem and the average scheme remains significantly higher than that of the conventional scheme, even at high LNR values. This is because the former two selectively upload informative blocks, whereas the conventional scheme randomly transmits as many blocks as possible, inevitably missing critical information. Moreover, due to the excessive data volume, its performance remains limited even under high SNR, unless the SNR is further increased to unrealistic levels beyond practical settings. Thirdly, SenSem consistently outperforms YOLO-based scheme, since its BS-side CSI-based RoI estimation eliminates device-side inference latency and the semantic evaluator enables prioritized transmission of the most informative blocks. Fourthly, the absence of the semantic evaluator causes a significant performance drop because devices randomly select blocks, making it unlikely that semantically important regions are transmitted. Likewise, removing sensing-assisted video analytics leads to varying degrees of degradation: the drop is small when the video frames already provide high-quality visual information, but becomes substantial when the original visual accuracy is low, where CSI plays a more critical complementary role.

To further analyze the benefits of integrated sensing and semantic communications, we present the ratio of transmitted blocks to total blocks at LNR = 20 dB in Fig. 12(c). From it, we can observe that the average ratio of the transmitted blocks is around 5.5 %, which is consistent with the previous results in Section V-B. Specifically, by leveraging sensing for cropping, the number of blocks to be transmitted is reduced to only 18.29 % of the original, as shown in Fig. 8(a). On top of this, semantic communications further reduce the transmission volume, requiring only about 30 % of the blocks to achieve performance comparable to the full image, as demonstrated in Fig. 9. Consequently, the overall compression ratio is reasonably 5.5 %. These findings confirm that integrated sensing and semantic communications can significantly reduce communication overhead while maintaining high performance.

We further visualize the selected blocks of the same frame under different LNRs in Fig. 13. The results clearly show that the location-assisted RoI estimation effectively discards non-essential regions of the image. Building on this, Algorithm 1 adaptively selects blocks based on LNR, with transmission ratios of 18.9 %, 29.3 %, and 38.7 % for LNR being 10 dB, 15 dB, and 20 dB, respectively. This trend indicates that as LNR increases, more blocks are uploaded, leveraging improved channel conditions. Notably, even in the worst-case scenario

(LNR=10dB), the vast majority of critical blocks are uploaded, ensuring high performance.

Fig. 12(d) illustrates the performance of the six schemes under different numbers of devices. As the number of devices increases, all six schemes exhibit a monotonic performance decline. This is because a higher number of devices intensifies competition for limited communication resources, reducing the available time slots and the number of transmitted blocks for each user. Additionally, it can be observed that the decline rate of the proposed SenSem is lower than that of the average scheme. This is attributed to the superiority of Algorithm 1, which adaptively determines the number of transmitted blocks for each user based on channel conditions and the semantic density of each block.

VI. Conclusion

In this paper, we have proposed SenSem, a novel framework that integrates sensing and semantic communications to reduce transmission and computation overhead in the multi-device video analytics system. Specifically, we have introduced a location-assisted RoI estimation method to reduce the amount of information to be processed. Additionally, we have designed a low-complexity semantic evaluator and, based on this, developed a joint block selection and transmit power control algorithm, ensuring that only high-semantic-content blocks are uploaded to the BS for analytics. At the edge server, we have designed a sensing-assisted video analytics network to enhance analysis performance. To validate SenSem, we have collected a joint CSI and video dataset using three WARP boards and a webcam. Extensive evaluation results have demonstrated that our proposed RoI estimation method and semantic evaluator effectively reduce the number of transmitted blocks to just 5.5 % of the original one, while preserving key information. Meanwhile, the sensing-assisted network improves video analytics performance at the edge. Moreover, compared to baseline methods, the proposed framework achieves higher analytics performance under identical latency constraints, confirming the effectiveness of SenSem.

In this paper, we consider video monitoring/analytics, and SenSem mainly applies to cases where the targets remain visible to the camera; otherwise, the monitoring itself would lose its meaning. Under this, SenSem is suitable for both indoor (e.g., classrooms) and outdoor environments (e.g., smart factories and warehouses). Its performance can be further improved by leveraging larger bandwidths and more antennas to enhance the accuracy of RoI estimation. In some scenarios, however, occlusion may occur between the camera and the target. In practice, such an issue is addressed through multi-camera collaboration. As part of our future work, we plan to explore how SenSem can jointly exploit video and CSI from multiple cameras to further enhance robustness.

References

[1] S. Wan, X. Xu, T. Wang, and Z. Gu, "An Intelligent Video Analysis Method for Abnormal Event Detection in Intelligent Transportation Systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 22, no. 7, pp. 4487–4495, Jul. 2020.

[2] Q. Zhang, H. Sun, X. Wu, and H. Zhong, "Edge Video Analytics for Public Safety: A Review," *Proc. IEEE*, vol. 107, no. 8, pp. 1675–1696, Aug. 2019.

[3] T. Shaik, X. Tao, N. Higgins, L. Li, R. Gururajan, X. Zhou, and U. R. Acharya, "Remote Patient Monitoring using Artificial Intelligence: Current State, Applications, and Challenges," *WIREs Data Mining Knowl. Discovery*, vol. 13, no. 2, p. e1485, Jan. 2023.

[4] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[5] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 5–41, Jan. 2022.

[6] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless Semantic Communications for Video Conferencing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 230–244, Jan. 2023.

[7] T.-Y. Tung and D. Gündüz, "DeepWiVe: Deep-Learning-Aided Wireless Video Transmission," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2570–2583, Sep. 2022.

[8] H. Li, H. Tong, S. Wang, N. Yang, Z. Yang, and C. Yin, "Video Semantic Communication with Major Object Extraction and Contextual Video Encoding," *arXiv:2402.01330*, 2024.

[9] J. Shao, X. Zhang, and J. Zhang, "Task-Oriented Communication for Edge Video Analytics," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 5, pp. 4141–4154, May 2024.

[10] Z. Fang, S. Hu, L. Yang, Y. Deng, X. Chen, and Y. Fang, "PIB: Prioritized Information Bottleneck Framework for Collaborative Edge Video Analytics," *arXiv:2408.17047*, 2024.

[11] Y. He, G. Yu, Y. Cai, and H. Luo, "Integrated Sensing, Computation, and Communication: System Framework and Performance Optimization," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 2, pp. 1114–1128, Feb. 2024.

[12] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and Modeling of WiFi Signal Based Human Activity Recognition," in *Proc. ACM MobiCom*, Sep. 2015, pp. 65–76.

[13] J. Hu, Z. Chen, T. Zheng, R. Schober, and J. Luo, "HoloFed: Environment-Adaptive Positioning via Multi-Band Reconfigurable Holographic Surfaces and Federated Learning," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3736–3751, Dec. 2023.

[14] X. Liu, J. Cao, S. Tang, J. Wen, and P. Guo, "Contactless Respiration Monitoring Via Off-the-Shelf WiFi Devices," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2466–2479, Oct. 2015.

[15] J. Hu, T. Zheng, Z. Chen, H. Wang, and J. Luo, "MUSE-Fi: Contactless MUti-person SEnsing Exploiting Near-field Wi-Fi Channel Variation," in *Proc. ACM MobiCom*, Oct. 2023, pp. 1–15.

[16] F. Liu, Y. Cui, C. Masouros, J. Xu, T. X. Han, Y. C. Eldar, and S. Buzzi, "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, Jun. 2022.

[17] Y. He, J. Liu, M. Li, G. Yu, J. Han, and K. Ren, "SenCom: Integrated Sensing and Communication with Practical WiFi," in *Proc. ACM MobiCom*, Oct. 2023, pp. 1–16.

[18] D. Wen, P. Liu, G. Zhu, Y. Shi, J. Xu, Y. C. Eldar, and S. Cui, "Task-Oriented Sensing, Computation, and Communication Integration for Multi-Device Edge AI," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 3, pp. 2486–2502, Mar. 2024.

[19] Y. Ma, G. Zhou, and S. Wang, "WiFi Sensing with Channel State Information: A Survey," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–36, Jun. 2019.

[20] Y. He, L. Fan, L. Xie, D. Niyato, C. Yuen, and J. Luo, "Invisible Walls: Privacy-Preserving ISAC Empowered by Reconfigurable Intelligent Surfaces," *submitted to IEEE J. Sel. Areas Commun.*, 2025.

[21] X. Li, H. Wang, Z. Chen, Z. Jiang, and J. Luo, "UWB-Fi: Pushing Wi-Fi towards Ultra-wideband for Fine-Granularity Sensing," in *Proc. ACM MobiSys*, Jun. 2024, pp. 42–55.

[22] K. Du, Q. Zhang, A. Arapin, H. Wang, Z. Xia, and J. Jiang, "AccMPEG: Optimizing Video Encoding for Video Analytics," in *Proc. MLSys*, Aug. 2022, pp. 1–17.

[23] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose++: Vision Transformer for Generic Body Pose Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 2, pp. 1212–1230, Feb. 2024.

[24] Y. Shi, Y. Zhou, D. Wen, Y. Wu, C. Jiang, and K. B. Letaief, "Task-Oriented Communications for 6G: Vision, Principles, and Technologies," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 78–85, Jun. 2023.

[25] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep Learning Enabled Semantic Communication Systems," *IEEE Trans. on Signal Process.*, vol. 69, pp. 2663–2675, 2021.

[26] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, "Resource Allocation for Text Semantic Communications," *IEEE Wireless Commun. Lett.*, vol. 11, no. 7, pp. 1394–1398, Jul. 2022.

[27] H. Xie, Z. Qin, and G. Y. Li, "Semantic Communication With Memory," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 8, pp. 2658–2669, Aug. 2023.

[28] D. Huang, F. Gao, X. Tao, Q. Du, and J. Lu, "Toward Semantic Communications: Deep Learning-Based Image Semantic Coding," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 55–71, Jan. 2022.

[29] S. Tang, Q. Yang, L. Fan, X. Lei, A. Nallanathan, and G. K. Karagiannidis, "Contrastive Learning-Based Semantic Communications," *IEEE Trans. on Commun.*, vol. 72, no. 10, pp. 6328–6343, Oct. 2024.

[30] Z. Lyu, G. Zhu, J. Xu, B. Ai, and S. Cui, "Semantic Communications for Image Recovery and Classification via Deep Joint Source and Channel Coding," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 8, pp. 8388–8404, Aug. 2024.

[31] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[32] S. Wang, J. Dai, Z. Liang, K. Niu, Z. Si, C. Dong, X. Qin, and P. Zhang, "Wireless Deep Video Semantic Transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 214–229, Jan. 2022.

[33] Y. He, J. Liu, M. Li, G. Yu, and J. Han, "Forward-Compatible Integrated Sensing and Communication for WiFi," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 9, pp. 2440–2456, Sep. 2024.

[34] F. Dong, F. Liu, S. Lu, Y. Xiong, Q. Zhang, Z. Feng, and F. Gao, "Communication-Assisted Sensing in 6G Networks," *arXiv:2311.07157*, 2023.

[35] Y. He, M. Xu, F. Xiao, and J. Luo, "VersaBeam: Versatile Beamforming for Integrated Sensing and Communication over Commodity Wi-Fi," in *Proc. INFOCOM*, May 2025, pp. 1–10.

[36] K. Meng, Q. Wu, W. Chen, and D. Li, "Sensing-Assisted Communication in Vehicular Networks With Intelligent Surface," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 876–893, Jan. 2024.

[37] F. Liu, W. Yuan, C. Masouros, and J. Yuan, "Radar-Assisted Predictive Beamforming for Vehicular Links: Communication Served by Sensing," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 11, pp. 7704–7719, Nov. 2020.

[38] Y. Yang, Z. Yang, C. Huang, W. Xu, Z. Zhang, D. Niyato, and M. Shikh-Bahaei, "Integrated Sensing, Computing and Semantic Communication for Vehicular Networks," *IEEE Trans. Veh. Commun.*, 2025, early access, doi:10.1109/TVT.2025.3575243.

[39] Y. Yang, M. Shikh-Bahaei, Z. Yang, C. Huang, W. Xu, and Z. Zhang, "Secure Design for Integrated Sensing and Semantic Communication System," in *Proc. IEEE WCNC*, Apr. 2024, pp. 1–7.

[40] S. Mao, C. Yuen, L. Liu, M. Xiao, S. Yu, and N. Zhang, "RIS-enhanced Semantic-aware Sensing, Communication, Computation and Control for Internet of Things," *IEEE Trans. on Wireless Commun.*, 2025, early access, doi:10.1109/TWC.2025.3595550.

[41] G. Stockman and L. G. Shapiro, *Computer Vision*. United States: Prentice Hall PTR, 2001.

[42] G. De la Roche, A. Alayón-Glazunov, and B. Allen, *LTE-Advanced and Next Generation Wireless Networks: Channel Modelling and Propagation*. John Wiley & Sons, 2012.

[43] F. Wang, S. Zhou, S. Panev, J. Han, and D. Huang, "Person-in-WiFi: Fine-Grained Person Perception Using WiFi," in *Proc. IEEE/CVF ICCV*, Oct. 2019, pp. 5452–5461.

[44] S. Pratschner, S. Schwarz, and M. Rupp, "Single-User and Multi-User MIMO Channel Estimation for LTE-Advanced Uplink," in *Proc. IEEE ICC*, May 2017, pp. 1–6.

[45] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter Level Localization Using WiFi," in *Proc. ACM SIGCOMM*, Aug. 2015, pp. 269–282.

[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetv2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE CVPR*, Jun. 2018, pp. 4510–4520.

[47] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An Iteratively Weighted MMSE Approach to Distributed Sum-Utility Maximization for a MIMO Interfering Broadcast Channel," *IEEE Trans. on Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[48] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[49] J. Kiefer, "Sequential Minimax Search for a Maximum," *Proc. Amer. Math. Soc.*, vol. 4, no. 3, pp. 502–506, Jun. 1953.

[50] W. Li, M. Liu, H. Liu, P. Wang, J. Cai, and N. Sebe, "Hourglass Tokenizer for Efficient Transformer-Based 3D Human Pose Estimation," in *Proc. IEEE/CVF CVPR*, Jun. 2024, pp. 604–613.

[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *Proc. ICLR*, May 2021, pp. 1–21.

[52] Rice Univ., "Wireless Open Access Research Platform (WARP)," Online; accessed: 16 January 2025. [Online]. Available: https://warpproject.org/

[53] J. Caleya-Sanchez, P. Muñoz, J. Sánchez-Garrido, E. Florentín, F. Delgado-Ferro, P. Rodriguez-Martin, and P. Ameigeiras, "Empirical Evaluation of a 5G Transparent Clock for Time Synchronization in a TSN-5G Network," *arXiv:2509.06454*, 2025.

[54] J. Li, B. Li, and Y. Lu, "Deep Contextual Video Compression," *Proc. NeurIPS*, vol. 34, pp. 18 114–18 125, Dec. 2021.

[55] R. Khanam and M. Hussain, "Yolov11: An Overview of the Key Architectural Enhancements," *arXiv:2410.17725*, 2024.

[56] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, "Wireless Image Transmission Using Deep Source Channel Coding With Attention Modules," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2315–2328, Apr. 2021.

[57] J. Tu, X. Liu, Y. Wei, F. Zhou, and S. Ma, "Lightweight Semantic Communication for Wireless Image Transmission," *IEEE Wireless Commun. Lett.*, 2025, early access, doi:10.1109/LWC.2025.3614498.

[58] X. Wang, X. Zhang, Y. Cao, W. Wang, C. Shen, and T. Huang, "SegGpt: Towards Segmenting Everything in Context," in *Proc. of the IEEE/CVF ICCV*, Oct. 2023, pp. 1130–1140.